Alípio Jorge Luís Torgo
Pavel Brazdil Rui Camacho
João Gama (Eds.)

LNAI 3721

Knowledge Discovery in Databases: PKDD 2005

9th European Conference on Principles and Practice of Knowledge Discovery in Databases
Porto, Portugal, October 2005, Proceedings

Springer

Lecture Notes in Artificial Intelligence 3721

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Alípio Jorge Luís Torgo
Pavel Brazdil Rui Camacho
João Gama (Eds.)

Knowledge Discovery in Databases: PKDD 2005

9th European Conference on Principles and Practice
of Knowledge Discovery in Databases
Porto, Portugal, October 3-7, 2005
Proceedings

Springer

Series Editors

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

Volume Editors

Alípio Jorge
Luís Torgo
Pavel Brazdil
João Gama
LIACC/FEP, University of Porto
Rua de Ceuta, 118, 6°, 4050-190 Porto, Portugal
E-mail: {amjorge,ltorgo,pbrazdil,jgama}@liacc.up.pt

Rui Camacho
LIACC/FEUP, University of Porto
Rua de Ceuta, 118, 6°, 4050-190 Porto, Portugal
E-mail: rcamacho@fe.up.pt

Library of Congress Control Number: 2005933047

CR Subject Classification (1998): I.2, H.2, J.1, H.3, G.3, I.7, F.4.1

ISSN 0302-9743
ISBN-10 3-540-29244-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29244-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11564126 06/3142 5 4 3 2 1 0

Preface

The European Conference on Machine Learning (ECML) and the European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD) were jointly organized this year for the fifth time in a row, after some years of mutual independence before. After Freiburg (2001), Helsinki (2002), Cavtat (2003) and Pisa (2004), Porto received the 16th edition of ECML and the 9th PKDD in October 3–7.

Having the two conferences together seems to be working well: 585 different paper submissions were received for both events, which maintains the high submission standard of last year. Of these, 335 were submitted to ECML only, 220 to PKDD only and 30 to both. Such a high volume of scientific work required a tremendous effort from Area Chairs, Program Committee members and some additional reviewers. On average, PC members had 10 papers to evaluate, and Area Chairs had 25 papers to decide upon. We managed to have 3 highly qualified independent reviews per paper (with very few exceptions) and one additional overall input from one of the Area Chairs. After the authors' responses and the online discussions for many of the papers, we arrived at the final selection of 40 regular papers for ECML and 35 for PKDD. Besides these, 32 others were accepted as short papers for ECML and 35 for PKDD. This represents a joint acceptance rate of around 13% for regular papers and 25% overall. We thank all involved for all the effort with reviewing and selection of papers.

Besides the core technical program, ECML and PKDD had 6 invited speakers, 10 workshops, 8 tutorials and a Knowledge Discovery Challenge. Our special thanks to the organizers of the individual workshops and tutorials and to the workshop and tutorial chairs Floriana Esposito and Dunja Mladenić and to the challenge organizer Petr Berka. A very special word to Richard van de Stadt for all his competence and professionalism in the management of CyberChairPRO. Our thanks also to everyone from the Organization Committee mentioned further on who helped us with the organization. Our acknowledgement also to Rodolfo Matos and Assunção Costa Lima for providing logistic support.

Our acknowledgements to all the sponsors, Fundação para a Ciência e Tecnologia (FCT), LIACC-NIAAD, Faculdade de Engenharia do Porto, Faculdade de Economia do Porto, KDubiq –Knowledge Discovery in Ubiquitous Environments— Coordinated Action of FP6, Salford Systems, Pascal Network of Excellence, PSE/SPSS, ECCAI and Comissão de Viticultura da Região dos Vinhos Verdes. We also wish to express our gratitude to all other individuals and institutions not explicitly mentioned in this text who somehow contributed to the success of these events.

Finally, our word of appreciation to all the authors who submitted papers to the main conferences and their workshops, without whom none of this would have been possible.

July 2005

Alípio Jorge, Luís Torgo, Pavel Brazdil,
Rui Camacho and João Gama

Organization

ECML/PKDD 2005 Organization

Executive Committee

General Chair

Pavel Brazdil (LIACC/FEP, Portugal)

Program Chairs

Rui Camacho (LIACC/FEUP, Portugal)
João Gama (LIACC/FEP, Portugal)

Alípio Jorge (LIACC/FEP, Portugal)
Luís Torgo (LIACC/FEP, Portugal)

Workshop Chair

Floriana Esposito (University of Bari, Italy)

Tutorial Chair

Dunja Mladenić (Jozef Stefan Institute, Slovenia)

Challenge Chairs

Petr Berka (University of Economics, Czech Republic)
Bruno Crémilleux (Université de Caen, France)

Local Organization Committee

Pavel Brazdil, Alípio Jorge, Rui Camacho, Luís Torgo and João Gama, with the help of people from LIACC-NIAAD, University of Porto, Portugal, Rodolfo Matos, Pedro Quelhas Brito, Fabrice Colas, Carlos Soares, Pedro Campos, Rui Leite, Mário Amado Alves, Pedro Rodrigues; and from IST, Lisbon, Portugal, Cláudia Antunes.

Steering Committee

Nada Lavrač, Jozef Stefan Institute, Slovenia
Dragan Gamberger, Rudjer Boskovic Institute, Croatia
Ljupčo Todorovski, Jozef Stefan Institute, Slovenia
Hendrik Blockeel, Katholieke Universiteit Leuven, Belgium
Tapio Elomaa, Tampere University of Technology, Finland
Heikki Mannila, Helsinki Institute for Information Technology, Finland

Hannu T.T. Toivonen, University of Helsinki, Finland
Jean-François Boulicaut, INSA-Lyon, France
Floriana Esposito, University of Bari, Italy
Fosca Giannotti, ISTI-CNR, Pisa, Italy
Dino Pedreschi, University of Pisa, Italy

Area Chairs

Michael R. Berthold, Germany
Elisa Bertino, Italy
Ivan Bratko, Slovenia
Pavel Brazdil, Portugal
Carla E. Brodley, USA
Rui Camacho, Portugal
Luc Dehaspe, Belgium
Peter Flach, UK
Johannes Fürnkranz, Germany
João Gama, Portugal
Howard J. Hamilton, Canada
Thorsten Joachims, USA
Alípio Jorge, Portugal
Hillol Kargupta, USA
Pedro Larranaga, Spain
Ramon López de Mántaras, Spain
Dunja Mladenić, Slovenia
Hiroshi Motoda, Japan
José Carlos Príncipe, USA
Tobias Scheffer, Germany
Michele Sebag, France
Peter Stone, USA
Luís Torgo, Portugal
Gerhard Widmer, Austria

Program Committee

Jesus Aguilar, Spain
Paulo Azevedo, Portugal
Michael Bain, Australia
José Luís Balcázar, Spain
Elena Baralis, Italy
Bettina Berendt, Germany
Petr Berka, Czech Republic
Michael R. Berthold, Germany
Elisa Bertino, Italy
Hendrik Blockeel, Belgium
José Luís Borges, Portugal
Francesco Bonchi, Italy
Henrik Boström, Sweden
Marco Botta, Italy
Jean-François Boulicaut, France
Pavel Brazdil, Portugal
Carla E. Brodley, USA
Wray Buntine, Finland
Rui Camacho, Portugal
Amilcar Cardoso, Portugal
Barbara Catania, Italy
Jesús Cerquides, Spain
Susan Craw, Scotland
Bruno Cremilleux, France
James Cussens, UK
Luc Dehaspe, Belgium
Vasant Dhar, USA
Sašo Džeroski, Slovenia
Tapio Elomaa, Finland
Floriana Esposito, Italy
Jianping Fan, USA
Ronen Feldman, Israel
Cèsar Ferri, Spain
Peter Flach, UK
Eibe Frank, New Zealand
Alex Freitas, UK
Johannes Fürnkranz, Germany
Thomas Gabriel, Germany
João Gama, Portugal
Dragan Gamberger, Croatia
Minos N. Garofalakis, USA
Bart Goethals, Belgium
Gunter Grieser, Germany
Marko Grobelnik, Slovenia

Howard J. Hamilton, Canada
Colin de la Higuera, France
Melanie Hilario, Switzerland
Haym Hirsh, USA
Frank Höppner, Germany
Andreas Hotho, Germany
Eyke Hullermeier, German
Nitin Indurkhya, Australia
Thorsten Joachims, USA
Aleks Jakulin, Slovenia
Alípio Jorge, Portugal
Murat Kantarcioglu, USA
Hillol Kargupta, USA
Samuel Kaski, Finland
Mehmet Kaya, Turkey
Eamonn Keogh, USA
Roni Khardon, USA
Jorg-Uwe Kietz, Switzerland
Joerg Kindermann, Germany
Ross D. King, UK
Joost N. Kok, The Netherlands
Igor Kononenko, Slovenia
Stefan Kramer, Germany
Marzena Kryszkiewicz, Poland
Miroslav Kubat, USA
Nicholas Kushmerick, Ireland
Pedro Larranaga, Spain
Nada Lavrač, Slovenia
Jure Leskovec, USA
Jinyan Li, Singapore
Charles Ling, Canada
Donato Malerba, Italy
Giuseppe Manco, Italy
Ramon López de Mántaras, Spain
Stan Matwin, Canada
Michael May, Germany
Rosa Meo, Italy
Dunja Mladenić, Slovenia
Eduardo Morales, Mexico
Katharina Morik, Germany
Shinichi Morishita, Japan
Hiroshi Motoda, Japan
Steve Moyle, England
Gholamreza Nakhaeizadeh, Germany
Claire Nédellec, France
Richard Nock, France
Andreas Nürnberger, Germany
Masayuki Numao, Japan
Arlindo Oliveira, Portugal
Georgios Paliouras, Greece
Dino Pedreschi, Italy
Johann Petrak, Austria
Bernhard Pfahringer, New Zealand
Lubomir Popelinsky, Czech
José Carlos Príncipe, USA
Jan Ramon, Belgium
Zbigniew W. Ras, USA
Jan Rauch, Czech Republic
Christophe Rigotti, France
Gilbert Ritschard, Switzerland
John Roddick, Australia
Marques de Sá, Portugal
Lorenza Saitta, Italy
Daniel Sanchez, Spain
Yücel Saygin, Turkey
Tobias Scheffer, Germany
Michele Sebag, France
Marc Sebban, France
Giovanni Semeraro, Italy
Arno Siebes, The Netherlands
Andrzej Skowron, Poland
Carlos Soares, Portugal
Maarten van Someren, The Netherlands
Myra Spiliopoulou, Germany
Nicolas Spyratos, France
Ashwin Srinivasan, India
Olga Stepankova, Czech Republic
Reinhard Stolle, Germany
Peter Stone, USA
Gerd Stumme, Germany
Einoshin Suzuki, Japan
Domenico Talia, Italy
Ah-Hwee Tan, Singapore
Evimaria Terzi, Finland
Ljupčo Todorovski, Slovenia
Luís Torgo, Portugal
Shusaku Tsumoto, Japan
Peter Turney, Canada
Jaideep Vaidya, USA
Maria Amparo Vila, Spain

Dietrich Wettschereck, UK
Gerhard Widmer, Austria
Rüdiger Wirth, Germany
Mohammed J. Zaki, USA
Carlo Zaniolo, USA
Djamel A. Zighed, France

Additional Reviewers

Markus Ackermann
Anastasia Analyti
Fabrizio Angiulli
Orlando Anunciação
Giacometti Arnaud
Stella Asiimwe
Maurizio Atzori
Korinna Bade
Miriam Baglioni
Juergen Beringer
Philippe Bessières
Matjaž Bevk
Abhilasha Bhargav
Steffen Bickel
Stefano Bistarelli
Jan Blaťák
Axel Blumenstock
Janez Brank
Ulf Brefeld
Klaus Brinker
Michael Brückner
Robert D. Burbidge
Toon Calders
Michelangelo Ceci
Tania Cerquitelli
Eugenio Cesario
Vineet Chaoji
Silvia Chiusano
Fang Chu
Chris Clifton
Luís Coelho
Carmela Comito
Gianni Costa
Juan-Carlos Cubero
Agnieszka Dardzinska
Damjan Demšar
Nele Dexters
Christian Diekmann
Isabel Drost
Nicolas Durand
Mohamed Elfeky
Timm Euler
Nicola Fanizzi
Pedro Gabriel Ferreira
Francisco Ferrer
Daan Fierens
Sergio Flesca
Francesco Folino
Blaž Fortuna
Kenichi Fukui
Feng Gao
Yuli Gao
Paolo Garza
Aristides Gionis
Paulo Gomes
Gianluigi Greco
Fabian Güiza
Amaury Habrard
Hakim Hacid
Mohand-Said Hacid
Niina Haiminen
Mark Hall
Christoph Heinz
José Hernández-Orallo
Jochen Hipp
Juan F. Huete
Ali Inan
Ingo Mierswa
Stephanie Jacquemont
François Jacquenet
Aleks Jakulin
Tao Jiang
Wei Jiang
Xing Jiang

Sachindra Joshi
Pierre-Emmanuel Jouve
Michael Steinbach
George Karypis
Steffen Kempe
Arto Klami
Christian Kolbe
Stasinos Konstantopoulos
Matjaz Kukar
Minseok Kwon
Lotfi Lakhal
Carsten Lanquillon
Dominique Laurent
Yan-Nei Law
Roberto Legaspi
Aurélien Lemaire
Claire Leschi
Jure Leskovec
Francesca A. Lisi
Antonio Locane
Corrado Loglisci
Claudio Lucchese
Sara C. Madeira
Alain-Pierre Manine
M.J. Martin-Bautista
Stewart Massie
Cyrille Masson
Carlo Mastroianni
Nicolas Meger
Carlo Meghini
Taneli Mielikäinen
Mummoorthy Murugesan
Mirco Nanni
Mehmet Ercan Nergiz
Siegfried Nijssen
Janne Nikkilä
Blaž Novak
Lenka Novakova
Merja Oja
Riccardo Ortale
Ignazio Palmisano
Raffaele Perego
Sergios Petridis
Viet Phan-Luong
Frederic Piat
Dimitrios Pierrakos
Ulrich Rückert
María José Ramírez
Ganesh Ramakrishnan
Domenico Redavid
Chiara Renso
Lothar Richter
François Rioult
Céline Robardet
Pedro Rodrigues
Domingo S. Rodriguez–Baena
Luka Šajn
Saeed Salem
D. Sanchez
Eerika Savia
Karlton Sequeira
J.M. Serrano
Shengli Sheng
Javed Siddique
Georgios Sigletos
Fabrizio Silvestri
Arnaud Soulet
Hendrik Stange
Jan Struyf
Henri-Maxime Suchier
Andrea Tagarelli
Julien Thomas
Panayiotis Tsaparas
Yannis Tzitzikas
Jarkko Venna
Celine Vens
Nirmalie Wiratunga
Ghim-Eng Yap
Justin Zhan
Lizhuang Zhao
Igor Zwir

ECML/PKDD 2005 Tutorials

Ontology Learning from Text

[illegible]

Learning Automata as a Basis for Multi-agent Reinforcement Learning

[illegible]

Web Mining for Web Personalization

[illegible]

A Practical Time-Series Tutorial with MATLAB

[illegible]

Mining the Volatile Web

[illegible]

Spectral Clustering

[illegible]

Bioinspired Machine Learning Techniques

[illegible]

Probabilistic Inductive Logic Programming

[illegible]

ECML/PKDD 2005 Workshops

Sub-symbolic Paradigms for Learning in Structured Domains

European Web Mining Forum 2005 (EWMF 2005)

Knowledge Discovery in Inductive Databases (KDID 2005)

Mining Spatio-temporal Data

Cooperative Multiagent Learning

Data Mining for Business

Mining Graphs, Trees and Sequences (MGTS 2005)

Knowledge Discovery and Ontologies (KDO 2005)

Knowledge Discovery from Data Streams

Reinforcement Learning in Non-stationary Environments

Discovery Challenge

Table of Contents

Invited Talks

Long Papers

Short Papers

Data Analysis in the Life Sciences — Sparking Ideas —

Michael R. Berthold

ALTANA-Chair for Bioinformatics and Information Mining,
Dept. of Computer and Information Science, Konstanz University, Germany
Michael.Berthold@uni-konstanz.de

Data from various areas of Life Sciences have increasingly caught the attention of data mining and machine learning researchers. Not only is the amount of data available mind-boggling but the diverse and heterogenous nature of the information is far beyond any other data analysis problem so far. In sharp contrast to classical data analysis scenarios, the life science area poses challenges of a rather different nature for mainly two reasons. Firstly, the available data stems from heterogenous information sources of varying degrees of reliability and quality and is, without the interactive, constant interpretation of a domain expert, not useful. Furthermore, predictive models are of only marginal interest to those users – instead they hope for new insights into a complex, biological system that is only partially represented within that data anyway. In this scenario, the data serves mainly to create new insights and generate new ideas that can be tested. Secondly, the notion of feature space and the accompanying measures of similarity cannot be taken for granted. Similarity measures become context dependent and it is often the case that within one analysis task several different ways of describing the objects of interest or measuring similarity between them matter.

Some more recently published work in the data analysis area has started to address some of these issues. For example, data analysis in parallel universes [1], that is, the detection of patterns of interest in various different descriptor spaces at the same time, and mining of frequent, discriminative fragments in large, molecular data bases [2]. In both cases, sheer numerical performance is not the focus; it is rather the discovery of interpretable pieces of evidence that lights up new ideas in the users mind. Future work in data analysis in the life sciences needs to keep this in mind: the goal is to trigger new ideas and stimulate interesting associations.

References

1. Berthold, M.R., Wiswedel, B., Patterson, D.E.: Interactive exploration of fuzzy clusters using neighborgrams. Fuzzy Sets and Systems 149 (2005) 21–37
2. Hofer, H., Borgelt, C., Berthold, M.R.: Large scale mining of molecular fragments with wildcards. Intelligent Data Analysis 8 (2004) 376–385

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, p. 1, 2005.

Machine Learning for Natural Language Processing (and Vice Versa?)

Clai e Ca die

Department of Computer Science, Cornell University, USA
cardie@cs.cornell.edu
http://www.cs.cornell.edu/home/cardie/

Ove he a 10-15 yea , he infl ence of e hod f o achine lea ning ha an fo ed he way ha e ea ch i done in he eld of na al lang age o-ce ing. Thi al will begin by cove ing he hi o y of hi an fo a ion. In a ic la , lea ning e hod have oved cce f l in od cing and-alone ex - oce ing co onen o handle a n be of ling i ic a . Mo eove , he e co onen can be co bined o od ce y e ha exhibi hallow ex - nde anding ca abili ie : hey can, fo exa le, ex ac ey fac f o n e ic ed doc en in li i ed do ain o nd an we o gene al- o e e ion f o o en-do ain doc en collec ion . I will b iefly de c ibe he a e of he a fo he e ac ical ex - oce ing a lica ion , foc ing on he i o an ole ha achine lea ning e hod have layed in hei develo en .

The econd a of he al will ex lo e he ole ha na al lang age o-ce ing igh lay in achine lea ning e ea ch. He e, I will ex lain he ind of ex -ba ed fea e ha a e ela ively ea y o inco o a e in o achine lea ning da a e . In addi ion, I'll o line o e oble f o na al lang age oce - ing ha e i e, o co ld a lea bene f o , new achine lea ning algo i h .

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, p. 2, 2005.

Statistical Relational Learning: An Inductive Logic Programming Perspective

L c De Raed

Institute for Computer Science, Machine Learning Lab,
Albert-Ludwigs-University, Georges-Köhler-Allee, Gebäude 079,
D-79110 Freiburg i. Brg., Germany
deraedt@informatik.uni-freiburg.de

In he a few yea , he e ha been a lo of wo lying a he in e ec ion of obabili y heo y, logic og a ing and achine lea ning [14,18,13,9,6,1,11]. Thi wo i nown nde he na e of a i ical ela ional lea ning [7,5], obabili ic logic lea ning [4], o obabili ic ind c ive logic og a ing. Whe ea o of he exi ing wo have a ed f o a obabili ic lea ning e ec ive and ex ended obabili ic fo ali wi h ela ional a ec , I hall a e a diffe en e ec ive, in which I hall a f o ind c ive logic og a ing and dy how ind c ive logic og a ing fo ali , e ing and echni e can be ex- ended o deal wi h obabili ic i e . Thi adi ion ha al eady con ib ed a ich va ie y of val able fo ali and echni e , incl ding obabili ic Ho n abd c ion by David Poole, PRISM by Sa o, ocha ic logic og a by M g- gle on [13] and C en [2], Baye ian logic og a [10,8] by Ke ing and De Raed , and Logical Hidden Ma ov Model [11].

The ain con ib ion of hi al i he in od c ion of h ee obabili ic ind c ive logic og a ing e ing which a e de ived f o he lea ning f o en ail en , f o in e e a ion and f o oof e ing of he eld of ind c- ive logic og a ing [3]. Each of he e e ing con ib e diffe en no ion of obabili ic logic e e en a ion , exa le and obabili y di ib ion . The e ing, obabili ic lea ning f o en ail en , i inco o a ed in he well- nown PRISM y e [19] and C en ' Fail e Adj ed Maxi i a ion a - oach o a a e e e i a ion in ocha ic logic og a [2]. A novel y e ha wa ecen ly develo ed and ha hi a adig i he nFOIL y e [12]. I co bine ey inci le of he well- nown ind c ive logic og a ing y e FOIL [15] wi h he naïve Baye ' a aoch. In obabili ic lea ning f o en ail- en , exa le a e g o nd fac ha ho ld be obabili ically en ailed by he a ge logic og a . The econd e ing, obabili ic lea ning f o in e e a- ion , i inco o a ed in Baye ian logic og a [10,8], which in eg a e Baye ian ne wo wi h logic og a . Thi e ing i al o ado ed by [6]. Exa le in hi e ing a e He b and in e e a ion ha ho ld be a obabili ic odel fo he a ge heo y. The hi d e ing, lea ning f o oof [17], i novel. I i o iva ed by he lea ning of ocha ic con ex f ee g a a f o ee ban . In hi e ing, exa le a e oof ee ha ho ld be obabili ically ovable f o he n nown ocha ic logic og a . The e ched e ing (and hei in ance e en ed) a e by no ean he only o ible e ing fo obabili ic

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 3–5, 2005.

ind c ive logic , og a . ing, b . ill – I ho e – , ovide . ef l in igh . in o he . a e-of- he-a, of hi exci ing . eld.

Fo, a f ll . , vey of . a i ical , ela ional lea, ning o, , obabili ic ind c ive logic , og, a . ing, he a ho, wo ld li e o , efe, o [4], and fo, . o, e de ail on he , obabili ic ind c ive logic , og, a . ing . e ing o [16], whe, e a longe, and ea, lie, ve, . ion of hi con , ib ion can be fo nd.

Acknowledgements

Thi i join wo, wi h K, i ian Ke, . ing. The a ho, wo ld al o li e o han Niel Landweh, and S nna To, ge fo, in e, e ing collabo, a ion on nFOIL and he lea, ning of SLP , , e ec ively. Thi wo, i a, of he EU IST FET , ojec APRIL II (A lica ion of P, obabili ic Ind c ive Logic P, og, a . ing II).

References

1. C. R. Anderson, P. Domingos, and D. S. Weld. Relational Markov Models and their Application to Adaptive Web Navigation. In D. Hand, D. Keim, O. R. Zaïne, and R. Goebel, editors, *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 143–152, Edmonton, Canada, 2002. ACM Press.
2. J. Cussens. Loglinear models for first-order probabilistic reasoning. In K. B. Laskey and H. Prade, editors, *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 126–133, Stockholm, Sweden, 1999. Morgan Kaufmann.
3. L. De Raedt. Logical settings for concept-learning. *Artificial Intelligence*, 95(1):197–201, 1997.
4. L. De Raedt and K. Kersting. Probabilistic Logic Learning. *ACM-SIGKDD Explorations: Special issue on Multi-Relational Data Mining*, 5(1):31–48, 2003.
5. T. Dietterich, L. Getoor, and K. Murphy, editors. *Working Notes of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields (SRL-04)*, 2004.
6. N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conferences on Artificial Intelligence (IJCAI-99)*, pages 1300–1309, Stockholm, Sweden, 1999. Morgan Kaufmann.
7. L. Getoor and D. Jensen, editors. *Working Notes of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data (SRL-03)*, 2003.
8. K. Kersting and L. De Raedt. Adaptive Bayesian Logic Programs. In C. Rouveirol and M. Sebag, editors, *Proceedings of the Eleventh Conference on Inductive Logic Programming (ILP-01)*, volume 2157 of *LNCS*, Strasbourg, France, 2001. Springer.
9. K. Kersting and L. De Raedt. Bayesian logic programs. Technical Report 151, University of Freiburg, Institute for Computer Science, April 2001.
10. K. Kersting and L. De Raedt. Towards Combining Inductive Logic Programming and Bayesian Networks. In C. Rouveirol and M. Sebag, editors, *Proceedings of the Eleventh Conference on Inductive Logic Programming (ILP-01)*, volume 2157 of *LNCS*, Strasbourg, France, 2001. Springer.

11. K. Kersting, T. Raiko, S. Kramer, and L. De Raedt. Towards discovering structural signatures of protein folds based on logical hidden markov models. In R. B. Altman, A. K. Dunker, L. Hunter, T. A. Jung, and T. E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing*, pages 192 – 203, Kauai, Hawaii, USA, 2003. World Scientific.
12. N. Landwehr, K. Kersting, and L. De Raedt. nfoil: Integrating naive bayes and foil. In *Proceedings of the 20th National Conference on Artificial Intelligence*. AAAI Press, 2005.
13. S. H Muggleton. Stochastic logic programs. In L. De Raedt, editor, *Advances in Inductive Logic Programming*. IOS Press, 1996.
14. D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64:81–129, 1993.
15. J. R. Quinlan and R. M. Cameron-Jones. Induction of logic programs:FOIL and related systems. *New Generation Computing*, pages 287–312, 1995.
16. L. De Raedt and K. Kersting. Probabilistic inductive logic programming. In *Proceedings of the 15th International Conference on Algorithmic Learning Theory*. Springer, 2004.
17. L. De Raedt, K. Kersting, and S. Torge. Towards learning stochastic logic programs from proof-banks. In *Proceedings of the 20th National Conference on Artificial Intelligence*. AAAI Press, 2005.
18. T. Sato. A Statistical Learning Method for Logic Programs with Distribution Semantics. In L. Sterling, editor, *Proceedings of the Twelfth International Conference on Logic Programming (ICLP-1995)*, pages 715 – 729, Tokyo, Japan, 1995. MIT Press.
19. T. Sato and Y. Kameya. Parameter learning of logic programs for symbolic-statistical modeling. *Journal of Artificial Intelligence Research*, 15:391–454, 2001.

Recent Advances in Mining Time Series Data

Ea onn Keogh

Department of Computer Science & Engineering,
University of California, Riverside, USA
eamonn@cs.ucr.edu
http://www.cs.ucr.edu/~eamonn

M ch of he wo ld' ly of da a i in he fo of i e e ie . F he o e, a we hall ee, any y e of da a can be eaningf lly conve ed in o " i e e ie ", incl ding ex , DNA, video, i age e c. The la decade ha een an ex lo ion of in e e in ining i e e ie da a f o he acade ic co ni y. The e ha been igni can wo on algo i h o cla ify, cl e, eg en , index, di cove le , vi ali e, and de ec ano alie /novel ie in i e e ie .

In hi al I will a i e he la e advance in ining i e e ie da a, incl ding:

- New e e en a ion of i e e ie da a.
- New algo i h /de ni ion .
- The ig a ion f o a ic ble o online ble .
- New a ea and a lica ion of i e e ie da a ining.

I will end he al wi h a di c ion of "wha ' lef o do" in i e e ie da a ining.

References

1. E. Keogh. Exact indexing of dynamic time warping. In *Proceedings of the 8th International Conference on Very Large Data Bases*, pages 406–417, 2002.
2. E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 102–111, 2002.
3. E. Keogh, J. Lin, and W. Truppel. Clustering of time series subsequences is meaningless: Implications for past and future research. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 115–122, 2003.
4. E. Keogh, S. Lonardi, and C. Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.
5. C.A. Ratanamahatana and E. Keogh. Everything you know about dynamic time warping is wrong. In *Proceedings of the Third Workshop on Mining Temporal and Sequential Data, in conjunction with the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, 2004.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, p. 6, 2005.

Focus the Mining Beacon: Lessons and Challenges from the World of E-Commerce

Ron Kohavi

Microsoft Corporation, USA
ronnyk@cs.stanford.edu
http://www.kohavi.com

Elec ˋonic Co . eˋce i now en eˋing i . . econd decade, wi h A a on.co and eBay now in exi ence foˋ en yeaˋ . . Wi h . a . ive a o n . of da a, an ac- ionable do ain, and . ea ˋable ROI, . l i le co anie . e da a . ining and nowledge di coveˋ y o ndeˋ . and heiˋ c . o eˋ . and i ˋove in eˋ ac ion . We ˋe en i oˋ an le . on and challenge . ing e-co . eˋce exa le acˋ o . wo di en ion : (i) b . ine . -level o echnical, and (ii) he . ining lifecycle fˋ o da a collec ion, da a waˋ eho . e con ˋ c ion, o di coveˋ y and de loy en . Many of he le . on and challenge aˋ e a licable o do ain o . ide e-co . eˋce.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, p. 7, 2005.

Data Streams and Data Synopses for Massive Data Sets
(Invited Talk)

Yossi Matias

Tel Aviv University,
HyperRoll Inc., Stanford University
matias@tau.ac.il

Abstract. With the proliferation of data intensive applications, it has become necessary to develop new techniques to handle massive data sets. Traditional algorithmic techniques and data structures are not always suitable to handle the amount of data that is required and the fact that the data often streams by and cannot be accessed again. A field of research established over the past decade is that of handling massive data sets using data synopses, and developing algorithmic techniques for data stream models. We will discuss some of the research work that has been done in the field, and provide a decades' perspective to data synopses and data streams.

1 Summary

In recent years, we have witnessed an explosion in data used in various applications. In general, the growth rate in data is known to exceed the increase rate in the size of RAM, and of the available computation power (a.k.a. Moore's Law). As a result, traditional algorithms and data structures are often no longer adequate to handle the massive data sets required by these applications.

One approach to handle massive data sets is to use *external memory algorithms*, designed to make an effective utilization of I/O. In such algorithms the data structures are often implemented in external storage devices, and the objective is in general to minimize the number of I/Os. For a survey of works on external memory algorithms see [6]. Such algorithms assume that the entire input data is available for further processing. There are, however, many applications where the data is only seen once, as it "streams by". This may be the case in, e.g., financial applications, network monitoring, security, telecommunications data management, web applications, manufacturing, and sensor networks. Even in data warehouse applications, where the data may in general be available for additional querying, there are many situations where data analysis needs to be done as the data is loaded into the data warehouse, since the cost of accessing the data in a fully loaded production system may be significantly larger than just the basic cost of I/O. Additionally, even in the largest data warehouses, consisting of hundreds of terabytes, data is only maintained for a limited time, so access to historical data may often be infeasible.

It had thus become necessary to address situations in which massive data sets are required to be handled as they "stream by", and using only limited memory. Motivated by this need, the research field of data streams and data synopses has emerged

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 8 – 9, 2005.

and established over the last few years. We will discuss some of the research work that has been done in the field, and provide a decades' perspective to data streams and data synopses. A longer version of this abstract will be available at [4].

The data stream model is quite simple: it is assumed that the input data set is given as a sequence of data items. Each data item is seen only once, and any computation can be done utilizing the data structures maintained in main memory. These memory resident data structures are substantially smaller than the input data. As such, they cannot fully represent the data as is the case for traditional data structures, but can only provide a synopsis of the input data; hence they are denoted as *synopsis data structures*, or *data synopses* [3].

The use of data synopses implies that data analysis that is dependent on the entire streaming data will often be approximated. Furthermore, ad hoc queries that are dependent on the entire input data could only be served by the data synopses, and as a result only approximate answers to queries will be available. A primary objective in the design of data synopses is to have the smallest data synopses that would guarantee small, and if possible bounded, error on the approximated computation.

As we have shown in [1], some essential statistical data analysis, the so-called *frequency moments,* can be approximated using synopses that are as small as polynomial or even logarithmic in the input size. Over the last few years there has been a proliferation of additional works on data streams and data synopses. See, e.g., the surveys [2] and [5]. These works include theoretical results, as well as applications in databases, network traffic analysis, security, sensor networks, and program profiling; synopses include samples, random projections, histograms, wavelets, and XML synopses, among others. There remain a plethora of interesting open problems, both theoretical as well as applied.

References

1. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. J. of Computer and System Sciences 58 (1999), 137-147. STOC'96 Special Issue
2. Babcock, B., Babu, S., Datar, M., Motwani, R. Widom, J.: Models and issues in data stream systems. In Proc. Symposium on Principles of Database Systems (2002), 1-16
3. Gibbons, P.B., Matias, Y.: Synopses data structures for massive data sets. In: External memory algorithms, DIMACS Series Discrete Math. & TCS, AMS, 50 (1999). Also SODA'99
4. Matias, Y.: Data streams and data synopses for massive data sets. http://www.cs.tau.ac.il/~matias/streams/
5. Muthukrishnan, S.: Data streams: Algorithms and applications. http://www.cs.rutgers.edu/~muthu/stream-1-1.ps
6. Vitter, J.S.: External memory algorithms and data structures. ACM Comput Surv. 33(2): 209-271 (2001)

k-Anonymous Patterns

Maurizio Atzori[1,2], Francesco Bonchi[2], Fosca Giannotti[2], and Dino Pedreschi[1]

[1] Pisa KDD Laboratory, Computer Science Department, University of Pisa, Italy
{atzori, pedre}@di.unipi.it
[2] Pisa KDD Laboratory, ISTI - CNR, Pisa, Italy
{francesco.bonchi, fosca.giannotti}@isti.cnr.it

Abstract. It is generally believed that data mining results do not violate the *anonymity* of the individuals recorded in the source database. In fact, data mining models and patterns, in order to ensure a required statistical significance, represent a large number of individuals and thus conceal individual identities: this is the case of the minimum support threshold in association rule mining. In this paper we show that this belief is ill-founded. By shifting the concept of *k-anonymity* from data to patterns, we formally characterize the notion of a threat to anonymity in the context of pattern discovery, and provide a methodology to efficiently and effectively identify all possible such threats that might arise from the disclosure of a set of extracted patterns.

1 Introduction

Privacy Preserving Data Mining, i.e., the analysis of data mining side-effects on privacy, has recently become a key research issue and is receiving a growing attention from the research community [1,3,9,16]. However, despite such efforts, a common understanding of what is meant by "privacy" is still missing. This fact has led to the proliferation of many completely different approaches to privacy preserving data mining, all sharing the same generic goal: producing a valid mining model without disclosing "private" data. As highlighted in [9], the approaches pursued so far leave a privacy question open: do the data mining results themselves violate privacy? Put in other words, do the disclosure of extracted patterns open up the risk of privacy breaches that may reveal sensitive information? During the last year, few works [7,9,11] have tried to address this problem by some different points of view, but they all require some *a priori* knowledge of what is sensitive and what is not.

In this paper we study when data mining results represent *per se* a threat to privacy, independently of any background knowledge of what is sensitive. In particular, we focus on *individual privacy*, which is concerned with the *anonymity* of individuals.

A prototypical application instance is in the medical domain, where the collected data are typically very sensitive, and the kind of privacy usually required is the anonymity of the patients in a survey. Consider a medical institution where the usual hospital activity is coupled with medical research activity. Since physicians are the data collectors and holders, and they already know everything about their patients, they have unrestricted access to the collected information. Therefore, they can perform real mining on all available information using traditional mining tools – not necessarily the privacy preserving

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 10–21, 2005.

ones. This way they maximize the outcome of the knowledge discovery process, without any concern about privacy of the patients which are recorded in the data. But the anonymity of patients becomes a key issue when the physicians want to share their discoveries (e.g., association rules holding in the data) with their scientific community.

At a first sight, it seems that data mining results do not violate the anonymity of the individuals recorded in the source database. In fact, data mining models and patterns, in order to ensure a required statistical significance, represent a large number of individuals and thus conceal individual identities: this is the case of the minimum support threshold in association rule mining. The next example shows that the above belief is ill-founded.

Example 1. Consider the following association rule:

$$a_1 \wedge a_2 \wedge a_3 \Rightarrow a_4 \quad [sup = 80,\ conf = 98.7\%]$$

where sup and $conf$ are the usual interestingness measures of *support* and *confidence* as defined in [2]. Since the given rule holds for a number of individuals (80), which seems large enough to protect individual privacy, one could conclude that the given rule can be safely disclosed. But, is this all the information contained in such a rule? Indeed, one can easily derive the support of the premise of the rule:

$$sup(\{a_1, a_2, a_3\}) = \frac{sup(\{a_1, a_2, a_3, a_4\})}{conf} \approx \frac{80}{0.987} = 81.05$$

Given that the pattern $a_1 \wedge a_2 \wedge a_3 \wedge a_4$ holds for 80 individuals, and that the pattern $a_1 \wedge a_2 \wedge a_3$ holds for 81 individuals, we can infer that in our database there is just one individual for which the pattern $a_1 \wedge a_2 \wedge a_3 \wedge \neg a_4$ holds.

The knowledge inferred is a clear threat to the anonymity of that individual: on one hand the pattern identifying the individual could itself contain sensitive information; on the other hand it could be used to re-identify the same individual in other databases.

It is worth noting that this problem is very general: the given rule could be, instead of an association, a classification rule, or the path from the root to the leaf in a decision tree, and the same reasoning would still hold. Moreover, it is straightforward to note that, unluckily, the more accurate is a rule, the more unsafe it may be w.r.t. anonymity. As shown later, this anonymity problem can not be simply solved by discarding the most accurate rules: in fact, more complex kinds of threats to anonymity exist which involve more than simply two itemsets.

1.1 Related Works

During the last years a novel problem has emerged in privacy-preserving data mining [7,9,11]: do the data mining results themselves violate privacy? Only little preliminary work is available. The work in [9] studies the case of a classifier trained over a mixture of different kind of data: *public* (known to every one including the adversary), *private/sensitive* (should remain unknown to the adversary), and *unknown* (neither sensitive nor known by the adversary). The authors propose a model for privacy implication of the learned classifier.

In [11] the data owner, rather than sharing the data, prefers to share the mined association rules, but requires that a set of *restricted* association rules are not disclosed. The authors propose a framework to sanitize the output of association rules mining, while blocking some inference channels for the restricted rules.

In [7] a framework for evaluating classification rules in terms of their perceived privacy and ethical sensitivity is described. The proposed framework empowers the data miner with alerts for sensitive rules that can be accepted or dismissed by the user as appropriate. Such alerts are based on an aggregate *sensitivity combination function*, which assigns to each rule a value of sensitivity by aggregating the sensitivity value (an integer between 0 and 9) of each attribute involved in the rule. The process of labelling each attribute with its sensitivity value must be accomplished by the domain expert.

The fundamental difference of these approaches with ours lies in generality: we propose a novel, objective definition of privacy compliance of patterns without any reference to a preconceived knowledge of sensitive data or patterns, on the basis of the rather intuitive and realistic constraint that the anonymity of individuals should be guaranteed.

An important method for protecting individual privacy is k-*anonymity*, introduced in [14], a notion that establishes that the cardinality of the answer to any possible query should be at least k. In this work, it is shown that protection of individual sources does not guarantee protection when sources are cross-examined: a sensitive medical record, for instance, can be uniquely linked to a *named* voter record in a publicly available voter list through some shared attributes. The objective of k-anonymity is to eliminate such opportunities of inferring private information through cross linkage. In particular, this is obtained by a "sanitization" of the source data that is transformed in such a way that, for all possible queries, at least k tuples will be returned. Such a sanitization is obtained by generalization and suppression of attributes and/or tuples [15].

Trivially, by mining a k-anonymized database no patterns threatening the anonymity can be obtained. But such mining would produce models impoverished by the information loss which is intrinsic in the generalization and suppression techniques. Since our objective is to extract valid and interesting patterns, we propose to postpone k-anonymization after the actual mining step. In other words, we do not to enforce k-anonymity onto the source data, but instead we move such a concept to the extracted patterns.

1.2 Paper Contributions

In this paper we study the privacy problem described above in the very general setting of patterns which are boolean formulas over a binary database. Our contribution is twofold:

- we define k-anonymous patterns and provide a general characterization of inference channels holding among patterns that may threat anonymity of source data;
- we develop an effective and efficient algorithm to detect such potential threats, which yields a methodology to check whether the mining results may be disclosed without any risk of violating anonymity.

We emphasize that the capability of detecting the potential threats is extremely useful for the analyst to determine a trade-off among the quality of mining result and the

privacy guarantee, by means of an iterative interaction with the proposed detection algorithm. Our empirical experiments, reported in this paper, bring evidence to this observation. It should also be noted the different setting w.r.t. the other works in privacy preserving data mining: in our context no data perturbation or sanitization is performed; we allow real mining on the real data, while focussing on the *anonymity preservation properties of the extracted patterns*. We have also developed possible strategies to eliminate the threats to anonymity by introducing distortion on the dangerous patterns in a controlled way: for lack of space these results are omitted here but can be found in [5].

2 k-Anonymous Patterns and σ-Frequent Itemsets

We start by defining binary databases and patterns following the notation in [8].

Definition 1. *A binary database $\mathcal{D} = (\mathcal{I}, \mathcal{T})$ consists of a finite set of binary variables $\mathcal{I} = \{i_1, \ldots, i_p\}$, also known as* items*, and a finite multiset $\mathcal{T} = \{t_1, \ldots, t_n\}$ of p-dimensional binary vectors recording the values of the items. Such vectors are also known as* transactions*. A* pattern *for the variables in $\mathcal{I}$ is a logical (propositional) sentence built by AND ($\wedge$), OR ($\vee$) and NOT ($\neg$) logical connectives, on variables in $\mathcal{I}$. The domain of all possible patterns is denoted $\mathcal{P}at(\mathcal{I})$.*

According to Def. 1, $e \wedge (\neg b \vee \neg d)$, where $b, d, e \in \mathcal{I}$, is a pattern. One of the most important properties of a pattern is its frequency in the database, i.e. the number of individuals (transactions) in the database which make the given pattern true[1].

Definition 2. *Given a database $\mathcal{D}$, a transaction $t \in \mathcal{D}$ and a pattern p, we write $p(t)$ if t makes p true. The support of p in $\mathcal{D}$ is given by the number of transactions which makes p true: $\mathcal{S}_{\mathcal{D}}(p) = |\{t \in \mathcal{D} \mid p(t)\}|$.*

The most studied *pattern class* is the itemset, i.e., a conjunction of positive valued variables, or in other words, a set of items. The retrieval of itemsets which satisfy a minimum frequency property is the basic step of many data mining tasks, including (but not limited to) association rules [2,4].

Definition 3. *The set of all itemsets $2^{\mathcal{I}}$, is a pattern class consisting of all possible conjunctions of the form $i_1 \wedge i_2 \wedge \ldots \wedge i_m$. Given a database $\mathcal{D}$ and a minimum support threshold σ, the set of σ-frequent itemsets in $\mathcal{D}$ is denoted*

$$\mathcal{F}(\mathcal{D}, \sigma) = \{\langle X, \mathcal{S}_{\mathcal{D}}(X)\rangle \mid X \in 2^{\mathcal{I}} \wedge \mathcal{S}_{\mathcal{D}}(X) \geq \sigma\}$$

Itemsets are usually denoted in the form of set of the items in the conjunction, e.g. $\{i_1, \ldots, i_m\}$; or sometimes, simply $i_1 \ldots i_m$. Figure 1(b) shows the different notation used for general patterns and for itemsets. The problem addressed in this paper is given by the possibility of inferring from the output of frequent itemset mining, i.e, $\mathcal{F}(\mathcal{D}, \sigma)$, the existence of patterns with very low support (i.e., smaller than an anonymity threshold k, but not null): such patterns represent a threat for the anonymity of the individuals about which they are true.

[1] The notion of truth of a pattern w.r.t. a transaction t is defined in the usual way: t makes p true iff t is a model of the propositional sentence p.

$\mathcal{D}$

	a	b	c	d	e	f	g	h
t_1	1	1	1	1	1	1	1	1
t_2	1	1	1	1	1	0	1	0
t_3	1	1	1	1	1	0	0	0
t_4	1	1	1	1	1	1	1	0
t_5	1	1	1	1	1	0	0	0
t_6	1	1	1	1	1	0	0	0
t_7	1	1	0	1	1	0	0	0
t_8	1	0	0	0	1	1	1	0
t_9	0	0	1	1	1	1	1	0
t_{10}	0	0	1	1	1	0	0	0
t_{11}	0	0	1	1	1	1	1	1
t_{12}	1	1	0	0	0	1	1	0

(a)

Notation: patterns

$sup_{\mathcal{D}}(a \vee f) = 11$
$sup_{\mathcal{D}}(e \wedge (\neg b \vee \neg d)) = 4$
$sup_{\mathcal{D}}(h \wedge \neg b) = 1$

Notation: itemsets

$sup_{\mathcal{D}}(abc) = 6$
$sup_{\mathcal{D}}(abde) = 7$
$sup_{\mathcal{D}}(cd) = 9$

(b)

$\mathcal{F}(\mathcal{D}, 8) = \{\langle \emptyset, 12\rangle, \langle a, 9\rangle, \langle b, 8\rangle, \langle c, 9\rangle, \langle d, 10\rangle, \langle e, 11\rangle, \langle ab, 8\rangle, \langle ae, 8\rangle, \langle cd, 9\rangle, \langle ce, 9\rangle, \langle de, 10\rangle, \langle cde, 9\rangle\}$

(c)

$\mathcal{C}l(\mathcal{D}, 8) = \{\langle \emptyset, 12\rangle, \langle a, 9\rangle, \langle e, 11\rangle, \langle ab, 8\rangle, \langle ae, 8\rangle, \langle de, 10\rangle, \langle cde, 9\rangle\}$

(d)

$\mathcal{MC}h(3, \mathcal{C}l(\mathcal{D}, 6)) = \{\langle \mathcal{C}_{abde}^{abcde}, 1\rangle, \langle \mathcal{C}_{ae}^{abcde}, 1\rangle, \langle \mathcal{C}_{ab}^{abcde}, 1\rangle, \langle \mathcal{C}_{g}^{fg}, 1\rangle, \langle \mathcal{C}_{g}^{eg}, 1\rangle\}$

(e)

Fig. 1. Running example: (a) the binary database $\mathcal{D}$; (b) different notation used for patterns and itemsets; (c) the set of σ-frequent ($\sigma = 8$) itemsets; (d) the set of closed frequent itemsets; (e) the set of maximal inference channels for $k = 3$ and $\sigma = 6$

Definition 4. *Given a database $\mathcal{D}$ and an anonymity threshold k, a pattern p is said to be k-*anonymous *if* $\mathcal{I}_{\mathcal{D}}(p) \geq k$ *or* $\mathcal{I}_{\mathcal{D}}(p) = 0$.

2.1 Problem Definition

Before introducing our anonymity preservation problem, we need to define the inference of supports, which is the basic tool for the attacks to anonymity.

Definition 5. *A set S of pairs $\langle X, n\rangle$, where $X \in 2^{\mathcal{I}}$ and $n \in \mathbb{N}$, and a database $\mathcal{D}$ are said to be σ-compatible if $S \subseteq \mathcal{F}(\mathcal{D}, \sigma)$. Given a pattern p we say that $S \models \mathcal{I}(p) > x$ (respectively $S \models \mathcal{I}(p) < x$) if, for all databases $\mathcal{D}$ σ-compatible with S, we have that $\mathcal{I}_{\mathcal{D}}(p) > x$ (respectively $\mathcal{I}_{\mathcal{D}}(p) < x$).*

Informally, we call *inference channel* any subset of the collection of itemsets (with their respective supports), from which it is possible to infer non k-anonymous patterns. Our mining problem can be seen as frequent pattern extraction with two frequency thresholds: the usual minimum support threshold σ for itemsets (as defined in Definition 3), and an anonymity threshold k for general patterns (as defined in Definition 1).

Note that an itemset with support less than k is itself a non k-anonymous, and thus dangerous, pattern. However, since we can safely assume (as we will do in the rest of this paper) that $\sigma \gg k$, such pattern would be discarded by the usual mining algorithms.

Definition 6. *Given a collection of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$ and an anonymity threshold k, our problem consists in detecting all possible inference channels $\mathcal{C} \subseteq \mathcal{F}(\mathcal{D}, \sigma)$: $\exists p \in \mathcal{P}at(\mathcal{I}) : \mathcal{C} \models 0 < \mathcal{I}_{\mathcal{D}}(p) < k$.*

Obviously, a solution to this problem directly yields a method to formally prove that the disclosure of a given collection of frequent itemsets does not violate the anonymity

constraint: it is sufficient to check that no inference channel exists for the given collection. In this case, the collection can be safely distributed even to malicious adversaries. On the contrary, if this is not the case, we can proceed in two ways:

- mine a new collection of frequent itemsets under different circumstances, e.g., higher minimum support threshold, to look for an admissible collection;
- transform (sanitize) the collection to remove the inference channels.

The second alternative opens up many interesting mining problems, which are omitted here for lack of space, and are discussed in [5].

3 Detecting Inference Channels

In this Section we study how information about non k-anonymous patterns can be possibly inferred from a collection of σ-frequent itemsets. As suggested by Example 1, a simple inference channel is given by any itemset X which has a superset $X \cup \{a\}$ such that $0 < \mathcal{I}_{\mathcal{D}}(X) - \mathcal{I}_{\mathcal{D}}(X \cup \{a\}) < k$. In this case the pair $\langle X, \mathcal{I}_{\mathcal{D}}(X)\rangle, \langle X \cup \{a\}, \mathcal{I}_{\mathcal{D}}(X \cup \{a\})\rangle$ is an inference channel for the non k-anonymous pattern $X \wedge \neg a$, whose support is directly given by $\mathcal{I}_{\mathcal{D}}(X) - \mathcal{I}_{\mathcal{D}}(X \cup \{a\})$. This is a trivial kind of inference channel. Do more complex structures of itemsets exist that can be used as inference channels? In general, the support of a pattern $p = i_1 \wedge \cdots \wedge i_m \wedge \neg a_1 \wedge \cdots \wedge \neg a_n$ can be inferred if we know the support of itemsets $I = \{i_1, \ldots, i_m\}$, $J = I \cup \{a_1, \ldots, a_n\}$, and every itemset L such that $I \subset L \subset J$.

Lemma 1. *Given a pattern $p = i_1 \wedge \cdots \wedge i_m \wedge \neg a_1 \wedge \cdots \wedge \neg a_n$ we have that:*

$$\mathcal{I}_{\mathcal{D}}(p) = \sum_{I \subseteq X \subseteq J} (-1)^{|X \setminus I|} \mathcal{I}_{\mathcal{D}}(X)$$

where $I = \{i_1, \ldots, i_m\}$ and $J = I \cup \{a_1, \ldots, a_n\}$.

Proof. (Sketch) The proof follows directly from the definition of support and the well-known *inclusion-exclusion principle* [10].

Following the notation in [6], we denote the right-hand side of the equation above as $f_I^J(\mathcal{D})$. In the database $\mathcal{D}$ in Figure 1 we have that $\mathcal{I}_{\mathcal{D}}(b \wedge \neg d \wedge \neg e) = f_b^{bde}(\mathcal{D}) = \mathcal{I}_{\mathcal{D}}(b) - \mathcal{I}_{\mathcal{D}}(bd) - \mathcal{I}_{\mathcal{D}}(be) + \mathcal{I}_{\mathcal{D}}(bde) = 8 - 7 - 7 + 7 = 1$.

Definition 7. *Given a database $\mathcal{D}$, and two itemsets $I, J \in 2^{\mathcal{I}}$, $I = \{i_1, \ldots, i_m\}$ and $J = I \cup \{a_1, \ldots, a_n\}$, if $0 < f_I^J(\mathcal{D}) < k$, then the set of itemsets $\{X | I \subseteq X \subseteq J\}$ constitutes an inference channel for the non k-anonymous pattern $p = i_1 \wedge \cdots \wedge i_m \wedge \neg a_1 \wedge \cdots \wedge \neg a_n$. We denote such inference channel $\mathcal{C}_I^J$ and we write $\mathcal{I}_{\mathcal{D}}(\mathcal{C}_I^J) = f_I^J(\mathcal{D})$.*

Example 2. Consider the database $\mathcal{D}$ of Figure 1, and suppose $k = 3$. We have that $\mathcal{C}_{ab}^{abcde}$ is an inference channel of support 1. This means that there is only one transaction $t \in \mathcal{D}$ is such that $a \wedge b \wedge \neg c \wedge \neg d \wedge \neg e$.

The next Theorem states that if there exists a non k-anonymous pattern, then there exists a pair of itemsets $I \subseteq J \in 2^{\mathcal{I}}$ such that $\mathcal{C}_I^J$ is an inference channel.

Theorem 1. $\forall p \in \mathcal{P}at(\mathcal{I}) : 0 < \mathcal{S}upp_{\mathcal{D}}(p) < k \,.\, \exists\, I \subseteq J \in 2^{\mathcal{I}} : \mathcal{C}_I^J$.

Proof. The case of a conjunctive pattern p is a direct consequence of Lemma 1. Let us now consider a generic pattern $p \in \mathcal{P}at(\mathcal{I})$. Without loss of generality p is in *normal disjunctive form*: $p = p_1 \vee \ldots \vee p_q$, where $p_1 \ldots p_q$ are conjunctive patterns. We have that:

$$\mathcal{S}upp_{\mathcal{D}}(p) \geq \max_{1 \leq i \leq q} \mathcal{S}upp_{\mathcal{D}}(p_i).$$

Since $\mathcal{S}upp_{\mathcal{D}}(p) < k$ we have for all patterns p_i that $\mathcal{S}upp_{\mathcal{D}}(p_i) < k$. Moreover, since $\mathcal{S}upp_{\mathcal{D}}(p) > 0$ is there at least a pattern p_i such that $\mathcal{S}upp_{\mathcal{D}}(p_i) > 0$. Therefore, there is at least a conjunctive pattern p_i such that $0 < \mathcal{S}upp_{\mathcal{D}}(p_i) < k$.

From Theorem 1 we conclude that all possible threats to anonymity are due to inference channels of the form $\mathcal{C}_I^J$. However we can divide such inference channels in two subgroups:

1. inference channels involving only frequent itemsets;
2. inference channels involving also infrequent itemsets.

The first problem, addressed in the rest of this paper, is the most essential. In fact, a malicious adversary can easily find inference channels made up only of elements which are present in the disclosed output. However, these inference channels are not the unique possible source of inference: further inference channels involving also infrequent itemsets could be possibly discovered, albeit in a much more complex way.

In fact, in [6] deduction rules to derive tight bounds on the support of itemsets are introduced. Given an itemset J, if for each subset $I \subset J$ the support $\mathcal{S}upp_{\mathcal{D}}(I)$ is known, such rules allow to compute lower and upper bounds on the support of J. Let l be the greatest lower bound we can derive, and u the smallest upper bound we can derive: if we find that $l = u$ then we can infer that $\mathcal{S}upp_{\mathcal{D}}(J) = l = u$ without actual counting. In this case J is said to be a *derivable itemset*. We transpose such deduction techniques in our context and observe that they can be exploited to discover information about infrequent itemsets, and from these to infer non k-anonymous patterns. For lack of space, this higher-order problem is not discussed here, and left to the extended version of this paper. However, here we can say that the techniques to detect this kind of inference channels and to block them are very similar to the techniques for the first kind of channels. This is due to the fact that both kinds of channels rely on the same concept: inferring supports of larger itemsets from smaller ones. Indeed, the key equation of our work (Lemma 1) is also the basis of the deduction rules proposed in [6].

From now on we restrict our attention to the essential form of inference channel, namely those involving frequent itemsets only.

Definition 8. *The set of all $\mathcal{C}_I^J$ holding in $\mathcal{F}(\mathcal{D}, \sigma)$, together with their supports, is denoted* $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma)) = \{\langle \mathcal{C}_I^J, f_I^J(\mathcal{D}) \rangle \mid 0 < f_I^J(\mathcal{D}) < k \wedge \langle J, \mathcal{S}upp_{\mathcal{D}}(J) \rangle \in \mathcal{F}(\mathcal{D}, \sigma)\}$.

Algorithm 1 detects all possible inference channels $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$ that hold in a collection of frequent itemsets $\mathcal{F}(\mathcal{D}, \sigma)$ by checking all possible pairs of itemsets $I, J \in \mathcal{F}(\mathcal{D}, \sigma)$ such that $I \subseteq J$. This could result in a very large number of checks. Suppose that $\mathcal{F}(\mathcal{D}, \sigma)$ is formed only by a maximal itemset Y and all its subsets (an

Algorithm 1 Naïve Inference Channel Detector

Input: $\mathcal{F}(\mathcal{D}, \sigma), k$
Output: $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$
1: $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma)) = \emptyset$
2: **for all** $\langle J, sup(J)\rangle \in \mathcal{F}(\mathcal{D}, \sigma)$ **do**
3: **for all** $I \subseteq J$ **do**
4: **compute** f_I^J;
5: **if** $0 < f_I^J < k$ **then**
6: **insert**$\langle \mathcal{C}_I^J, f_I^J\rangle$ **in** $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$;

itemset is maximal if none of its proper supersets is in $\mathcal{F}(\mathcal{D}, \sigma)$). If $|Y| = n$ we get $|\mathcal{F}(\mathcal{D}, \sigma)| = 2^n$ (we also count the empty set), while the number of possible $\mathcal{C}_I^J$ is $\sum_{1 \leq i \leq n} \binom{n}{i} (2^i - 1)$. In the following Section we study some interesting properties that allow to dramatically reduce the number of checks needed to retrieve $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$.

4 A Condensed Representation of Inference Channels

In this section we introduce a condensed representation of $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, i.e., a subset of $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$ which is more efficient to compute, and sufficient to reconstruct the whole $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$. The benefits of having such condensed representation go far beyond mere efficiency. In fact, removing the redundancy existing in $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, we also implicitly avoid redundant sanitization, when we will block inference channels holding in $\mathcal{F}(\mathcal{D}, \sigma)$ (recall that, as stated before, the issue of how to block inference channels is not covered in this paper).

Consider, for instance, the two inference channels $\langle \mathcal{C}_{ad}^{acd}, 1\rangle$ and $\langle \mathcal{C}_{abd}^{abcd}, 1\rangle$ holding in the database in Fig. 1(a): one is more specific than the other, but they both uniquely identify transaction t_7. It is easy to see that many other families of equivalent, and thus redundant, inference channels can be found. *How can we directly identify one and only one representative inference channel in each family of equivalent ones?* The theory of *closed itemsets* can help us with this problem.

Closed itemsets were first introduced in [12] and since then they have received a great deal of attention especially by an algorithmic point of view [17,13]. They are a concise and lossless representation of all frequent itemsets, i.e., they contain the same information without redundancy. Intuitively, a closed itemset groups together all its subsets that have its same support; or in other words, it groups together itemsets which identify the same group of transactions.

Definition 9. *Given the function $f(T) = \{i \in \mathcal{I} \mid \forall t \in T, i \in t\}$, which returns all the items included in the set of transactions T, and the function $g(X) = \{t \in \mathcal{T} \mid \forall i \in X, i \in t\}$ which returns the set of transactions supporting a given itemset X, the composite function $c = f \circ g$ is the closure operator. An itemset I is closed iff and only if $c(I) = I$. Given a database $\mathcal{D}$ and a minimum support threshold σ, the set of frequent closed itemsets is denoted $\mathcal{C}l(\mathcal{D}, \sigma)$. An itemset $I \in \mathcal{C}l(\mathcal{D}, \sigma)$ is said to be maximal iff $\nexists J \supset I$ s.t. $J \in \mathcal{C}l(\mathcal{D}, \sigma)$.*

Analogously to what happens for the pattern class of itemsets, if we consider the pattern class of conjunctive patterns we can rely on the *anti-monotonicity property of frequency*. For instance, the number of transactions for which the pattern $a \wedge \neg c$ holds is always larger than the number of transactions for which the pattern $a \wedge b \wedge \neg c \wedge \neg d$ holds.

Definition 10. *Given two inference channels $\mathcal{C}_I^J$ and $\mathcal{C}_H^L$ we say that $\mathcal{C}_I^J \preceq \mathcal{C}_H^L$ when $I \subseteq H$ and $(J \setminus I) \subseteq (L \setminus H)$.*

Proposition 1. $\mathcal{C}_I^J \preceq \mathcal{C}_H^L \Rightarrow \forall \mathcal{D} \,.\, f_I^J(\mathcal{D}) \geq f_H^L(\mathcal{D})$.

Therefore, when detecting inference channels, whenever we find a $\mathcal{C}_H^L$ such that $f_H^L(\mathcal{D}) \geq k$, we can avoid checking the support of all inference channels $\mathcal{C}_I^J \preceq \mathcal{C}_H^L$, since they will be k-anonymous.

Definition 11. *An inference channel $\mathcal{C}_I^J$ is said to be* maximal *w.r.t. $\mathcal{D}$ and σ, if $\forall H, L$ such that $I \subseteq H$ and $(J \setminus I) \subseteq (L \setminus H)$, $f_H^L = 0$. The set of maximal inference channels is denoted $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$.*

Proposition 2. $\mathcal{C}_I^J \in \mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma)) \Rightarrow I \in \mathcal{C}l(\mathcal{D}, \sigma) \wedge J$ *is maximal.*

Proof. $i)$ $I \in \mathcal{C}l(\mathcal{D}, \sigma)$: if I is not closed then consider its closure $c(I)$ and consider $J' = J \cup (c(I) \setminus I)$. For the definition of closure, the set of transactions containing I is the same of the set of transactions containing $c(I)$, and the set of transactions containing J' is the same of the set of transactions containing J. It follows that $\mathcal{C}_{c(I)}^{J'} \succeq \mathcal{C}_I^J$ and $f_{c(I)}^J = f_I^J > 0$. Then, if I is not closed, $\mathcal{C}_I^J$ is not maximal.
$ii)$ J is maximal: if J is not maximal then consider its frequent superset $J' = J \cup \{a\}$ and consider $I' = I \cup a$. It is straightforward to see that $f_I^J = f_I^{J'} + f_{I'}^{J'}$ and that $\mathcal{C}_I^{J'} \succeq \mathcal{C}_I^J$ and $\mathcal{C}_{I'}^{J'} \succeq \mathcal{C}_I^J$. Therefore, since $f_I^J > 0$, at least one among $f_I^{J'}$ and $f_{I'}^{J'}$ must be not null. Then, if J is not maximal, $\mathcal{C}_I^J$ is not maximal as well.

The next Theorem shows how the support of any channel in $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$ can be reconstructed from $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$.

Theorem 2. *Given $\mathcal{C}_I^J \in \mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$, let M be any maximal itemset such that $M \supseteq J$. The following equation holds:*

$$f_I^J(\mathcal{D}) = \sum_{c(X)} f_{c(X)}^M(\mathcal{D})$$

where $c(I) \subseteq c(X) \subseteq M$ and $c(X) \cap (J \setminus I) = \emptyset$.

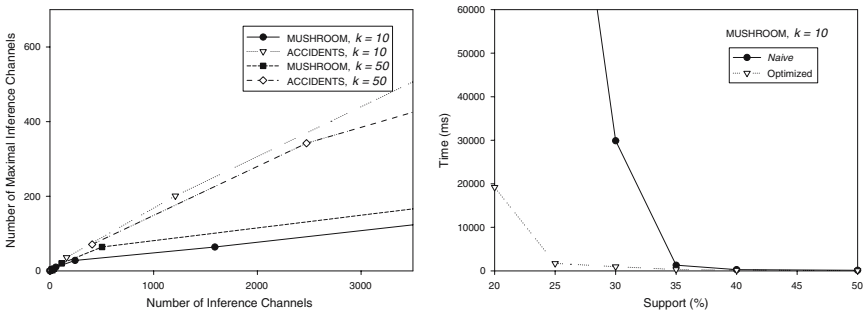
Proof. See [5].

From Theorem 2 we conclude that all the addends needed to compute $f_I^J(\mathcal{D})$ for an inference channel are either in $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ or are null. Therefore, as the set of all closed frequent itemsets $\mathcal{C}l(\mathcal{D}, \sigma)$ contains all the information of $\mathcal{F}(\mathcal{D}, \sigma)$ in a more compact representation, analogously the set $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ represents, without redundancy, all the information in $\mathcal{C}h(k, \mathcal{F}(\mathcal{D}, \sigma))$.

In the database $\mathcal{D}$ of Figure 1(a), given $\sigma = 6$ and $k = 3$, $|\mathcal{C}h(3, \mathcal{F}(\mathcal{D}, 6))| = 58$ while $|\mathcal{MC}h(3, \mathcal{C}l(\mathcal{D}, 6))| = 5$ (Figure 1(e)), a reduction of one order of magnitude which is also confirmed by our experiments on real datasets, as reported in Figure 2(a). Moreover, in order to detect all inference channels holding in $\mathcal{F}(\mathcal{D}, \sigma)$, we can limit ourselves to retrieve only the inference channels in $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$, thus taking in input $\mathcal{C}l(\mathcal{D}, \sigma)$ instead of $\mathcal{F}(\mathcal{D}, \sigma)$ and thus performing a much smaller number of checks. Algorithm 2 exploits the anti-monotonicity of frequency (Prop. 1) and the property of maximal inference channels (Prop. 2) to compute $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ from $\mathcal{C}l(\mathcal{D}, \sigma)$. Thanks to these two properties, Algorithm 2 is much faster, dramatically outperforming the naive inference channel detector (Algorithm 1), and scaling well even for very low support thresholds, as reported in Figure 2(b).

Algorithm 2 Optimized Inference Channel Detector

Input: $\mathcal{C}l(\mathcal{D}, \sigma), k$
Output: $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$
1: $M = \{I \in \mathcal{C}l(\mathcal{D}, \sigma) | I \ is \ maximal\}$;
2: $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma)) = \emptyset$;
3: **for all** $J \in M$ **do**
4: **for all** $I \in \mathcal{C}l(\mathcal{D}, \sigma)$ **such that** $I \subseteq J$ **do**
5: **compute** f_I^J;
6: **if** $0 < f_I^J < k$ **then**
7: **insert** $\langle \mathcal{C}_I^J, f_I^J \rangle$ **in** $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$;



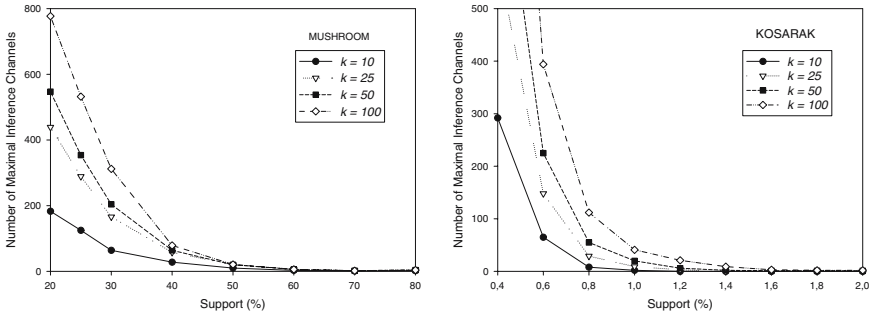
(a) (b)

Fig. 2. Benefits of the condensed representation: size of the representations (a), and run time (b)

5 Anonymity vs. Accuracy: Empirical Observations

Algorithm 2 represents an optimized way to identify all threats to anonymity. Its performance revealed adequate in all our empirical evaluations using various datasets from the FIMI repository[2]; in all such cases the time improvement from the Naïve (Algorithm 1)

[2] `http://fimi.cs.helsinki.fi/data/`



(a) (b)

Fig. 3. Experimental results on cardinality of $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$ on two datasets

to the optimized algorithm is about one order of magnitude. This level of efficiency allows an interactive-iterative use of the algorithm by the analyst, aimed at finding the best trade-off among privacy and accuracy of the collection of patterns. To be more precise, there is a conflict among keeping the support threshold as low as possible, in order to mine all interesting patterns, and avoiding the generation of anonymity threats. The best solution to this problem is precisely to find out the minimum support threshold that generates a collection of patterns with no threats. The plots in Figure 3 illustrate this point: on the x-axis we report the minimum support threshold, on the y-axis we report the total number of threats (the cardinality of $\mathcal{MC}h(k, \mathcal{C}l(\mathcal{D}, \sigma))$), and the various curves indicate such number according to different values of the anonymity threshold k. In Figure 3(a) we report the plot for the MUSHROOM dataset (a dense one), while in Figure 3(b) we report the plot for the KOSARAK dataset which is sparse. In both cases, it is evident the value of the minimum support threshold that represents the best trade-off, for any given value of k. However, in certain cases, the best support threshold can still be too high to mine a sufficient quantity of interesting patterns. In such cases, the only option is to allow lower support thresholds and then to block the inference channels in the mining outcome. This problem, as stated before, is not covered in this paper for lack of space, and will be presented in a forthcoming paper.

6 Conclusions

We introduced in this paper the notion of k-anonymous patterns. Such notion serves as a basis for a formal account of the intuition that a collection of patterns, obtained by data mining techniques and made available to the public, should not offer any possibilities to violate the privacy of the individuals whose data are stored in the source database. To the above aim, we formalized the threats to anonymity by means of inference channel through frequent itemsets, and provided practical algorithms to detect such channels.

Other issues, emerging from our approach, are worth a deeper investigation and are left to future research. These include: (i) a thorough comparison of the various dif-

ferent approaches that may be used to block inference channels; (ii) a comprehensive empirical evaluation of our approach: to this purpose we are conducting a large-scale experiment with real life bio-medical data about patients to assess both applicability and scalability of the approach in a realistic, challenging domain; (iii) an investigation whether the proposed notion of privacy-preserving pattern discovery may be generalized to other forms of patterns and models.

In any case, the importance of the advocated form of privacy-preserving pattern discovery is evident: demonstrably trustworthy data mining techniques may open up tremendous opportunities for new knowledge-based applications of public utility and large societal and economic impact.

References

1. D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM PODS*, 2001.
2. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD*.
3. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD on Management of Data*.
4. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the Twentieth VLDB*, 1994.
5. M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi. k-anonymous patterns. Technical Report 2005-TR-17, ISTI - C.N.R., 2005.
6. T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In *Proceedings of the 6th PKDD*, 2002.
7. P. Fule and J. F. Roddick. Detecting privacy and ethical sensitivity in data mining results. In *Proc. of the 27th conference on Australasian computer science*, 2004.
8. D. Hand, H. Mannila, and P. Smyh. *Principles of Data Mining*. The MIT Press, 2001.
9. M. Kantarcioglu, J. Jin, and C. Clifton. When do data mining results violate privacy? In *Proceedings of the tenth ACM SIGKDD*, 2004.
10. D. Knuth. *Fundamental Algorithms*. Addison-Wesley, Reading, Massachusetts, 1997.
11. S. R. M. Oliveira, O. R. Zaiane, and Y. Saygin. Secure association rule sharing. In *Proc.of the 8th PAKDD*, 2004.
12. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proc. ICDT '99*, 1999.
13. J. Pei, J. Han, and J. Wang. Closet+: Searching for the best strategies for mining frequent closed itemsets. In *SIGKDD '03*, 2003.
14. L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems*, 10(5), 2002.
15. L. Sweeney. k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems*, 10(5), 2002.
16. V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Rec.*, 33(1):50–57, 2004.
17. M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemsets mining. In *2nd SIAM International Conference on Data Mining*, 2002.

Interestingness is Not a Dichotomy: Introducing Softness in Constrained Pattern Mining

Stefano Bistarelli[1,2] and Francesco Bonchi[3]

[1] Dipartimento di Scienze, Università degli Studi "G. D'Annunzio", Pescara, Italy
[2] Istituto di Informatica e Telematica, CNR, Pisa, Italy
[3] Pisa KDD Laboratory, ISTI - C.N.R., Pisa, Italy
bista@sci.unich.it, francesco.bonchi@isti.cnr.it

Abstract. The paradigm of pattern discovery based on constraints was introduced with the aim of providing to the user a tool to drive the discovery process towards potentially *interesting* patterns, with the positive side effect of achieving a more efficient computation. So far the research on this paradigm has mainly focussed on the latter aspect: the development of efficient algorithms for the evaluation of constraint-based mining queries. Due to the lack of research on methodological issues, the constraint-based pattern mining framework still suffers from many problems which limit its practical relevance. As a solution, in this paper we introduce the new paradigm of pattern discovery based on *Soft Constraints*. Albeit simple, the proposed paradigm overcomes all the major methodological drawbacks of the classical constraint-based paradigm, representing an important step further towards practical pattern discovery.

1 Background and Motivations

During the last decade a lot of researchers have focussed their (mainly algorithmic) investigations on the computational problem of *Frequent Pattern Discovery*, i.e. mining patterns which satisfy a user-defined constraint of minimum frequency [1].

The simplest form of a frequent pattern is the frequent itemset.

Definition 1 (Frequent Itemset Mining). *Let $\mathcal{I} = \{x_1, ..., x_n\}$ be a set of distinct items, where an item is an object with some predefined attributes (e.g., price, type, etc.). An itemset X is a non-empty subset of $\mathcal{I}$. A transaction database $\mathcal{D}$ is a bag of itemsets $t \in 2^{\mathcal{I}}$, usually called transactions. The support of an itemset X in database $\mathcal{D}$, denoted $\mathcal{S}_{\mathcal{D}}(X)$, is the number of transactions which are superset of X. Given a user-defined minimum support σ, an itemset X is called frequent in $\mathcal{D}$ if $\mathcal{S}_{\mathcal{D}}(X) \geq \sigma$. This defines the minimum frequency constraint: $\mathcal{C}_{freq[\mathcal{D},\sigma]}(X) \Leftrightarrow \mathcal{S}_{\mathcal{D}}(X) \geq \sigma$.*

Recently the research community has turned its attention to more complex kinds of frequent patterns extracted from more structured data: *sequences*, *trees*, and *graphs*. All these different kinds of pattern have different peculiarities and application fields, but they all share the same computational aspects: a usually very large input, an exponential search space, and a too large solution set. This situation – too many data yielding too many patterns – is harmful for two reasons. First, performance degrades: mining generally becomes inefficient or, often, simply unfeasible. Second, the identification of the

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 22–33, 2005.

fragments of interesting knowledge, blurred within a huge quantity of mostly useless patterns, is difficult. The paradigm of *constraint-based pattern mining* was introduced as a solution to both these problems. In such paradigm, it is the user which specifies to the system what is *interesting* for the current application: constraints are a tool to drive the mining process towards potentially interesting patterns, moreover they can be pushed deep inside the mining algorithm in order to fight the exponential search space curse, and to achieve better performance [15,20,25].

When instantiated to the pattern class of itemsets, the constraint-based pattern mining problem is defined as follows.

Definition 2 (Constrained Frequent Itemset Mining). *A constraint on itemsets is a function* $\mathcal{C} : 2^{\mathcal{I}} \rightarrow \{$ [illegible] $\}$. *We say that an itemset* I *satisfies a constraint if and only if* $\mathcal{C}(I) =$ [illegible]. *We define the* $theory$ *of a constraint as the set of itemsets which satisfy the constraint:* [illegible]$(\mathcal{C}) = \{X \in 2^{\mathcal{I}} \mid \mathcal{C}(X)\}$. *Thus with this notation, the frequent itemsets mining problem requires to compute the set of all frequent itemsets* [illegible]$(\mathcal{C}_{freq[\mathcal{D},\sigma]})$. *In general, given a conjunction of constraints* $\mathcal{C}$ *the constrained frequent itemsets mining problem requires to compute* [illegible]$(\mathcal{C}_{freq}) \cap$ [illegible]$(\mathcal{C})$.

Example 1. The following is an example mining query:

$$\mathcal{Q} : \text{[illegible]}_{\mathcal{D}}(X) \geq 1500 \ \wedge \ avg(X.weight) \leq 5 \ \wedge \ sum(X.price) \geq 20$$

It requires to mine, from database $\mathcal{D}$, all patterns which are frequent (have a support larger than 1500), have average weight less than 5 and a sum of prices greater than 20.

So far constraint-based frequent pattern mining has been seen as a query optimization problem, i.e., developing efficient, sound and complete evaluation strategies for constraint-based mining queries. Or in other terms, designing efficient algorithms to mine all and only the patterns in [illegible]$(\mathcal{C}_{freq}) \cap$ [illegible]$(\mathcal{C})$. To this aim, properties of constraints have been studied comprehensively, and on the basis of such properties (e.g., anti-monotonicity, succinctness [20,18], monotonicity [11,17,6], convertibility [22], loose anti-monotonicity [9]), efficient computational strategies have been defined. Despite such effort, the constraint-based pattern mining framework still suffers from many problems which limit its practical relevance.

First of all, consider the example mining query $\mathcal{Q}$ given above: *where do the three thresholds (i.e., 1500, 5 and 20) come from?* In some cases they can be precisely imposed by the application, but this is rarely the case. In most of the cases, they come from an exploratory mining process, where they are iteratively adjusted until a solution set of reasonable size is produced. This practical way of proceeding is in contrast with the basic philosophy of the constraint-based paradigm: constraints should represent what is a priori interesting, given the application background knowledge, rather than be adjusted accordingly to a preconceived output size. Another major drawback of the constraint-based pattern mining paradigm is its rigidity. Consider, for instance, the following three patterns (we use the notation $\langle v_1, v_2, v_3 \rangle$ to denote the three values corresponding to the three constraints in the conjunction in the example query $\mathcal{Q}$): $p_1 : \langle 1700, 0.8, 19 \rangle$, $p_2 : \langle 1550, 4.8, 54 \rangle$, and $p_3 : \langle 1550, 2.2, 26 \rangle$. The first pattern, p_1, largely satisfies two out of the three given constraints, while slightly violates the third one. According to

the classical constraint-based pattern mining paradigm p_1 would be discarded as non interesting. Is such a pattern really *less interesting* than p_2 and p_3 which satisfy all the three constraints, but which are much less frequent than p_1? Moreover, is it reasonable, in real-world applications, that all constraints are equally important?

All these problems flow out from the same source: the fact that in the classical constraint-based mining framework, a constraint is a function which returns a boolean value $\mathcal{C} : 2^{\mathcal{I}} \rightarrow \{\text{[illegible]}\}$. Indeed, *interestingness is not a dichotomy*.

This consideration suggests us a simple solution to overcome all the main drawbacks of constraint-based paradigm.

Paper Contributions and Organization

In this paper, as a mean to handle interestingness [26,16,24], we introduce the *soft constraint based pattern mining* paradigm, where constraints are no longer rigid boolean functions, but are "soft" functions, i.e., functions with value in a set A, which represents the set of interest levels or costs assigned to each pattern.

- The proposed paradigm is not rigid: a potentially interesting pattern is not discarded for just a slight violation of a constraint.
- Our paradigm creates an order of patterns w.r.t. interestingness (level of constraints satisfaction): this allows to say that a pattern is *more interesting* than another, instead of strictly dividing patterns in interesting and not interesting.
- From the previous point it follows that our paradigm allows to express *top-k* queries based on constraints: the data analyst can ask for the top-10 patterns w.r.t. a given description (a conjunction of soft constraints).
- Alternatively, we can ask to the system to return all and only the patterns which exhibit an interest level larger than a given threshold λ.
- The proposed paradigm allows to assign different weights to different constraints, while in the classical constraint-based pattern discovery paradigm all constraints were equally important.
- Last but not least, our idea is very simple and thus very general: it can be instantiated to different classes of patterns such as itemsets, sequences, trees or graphs.

For the reasons listed above, we believe that the proposed paradigm represents an important step further towards practical pattern discovery.

A nice feature of our proposal is that, by adopting the soft constraint based paradigm, we do not reject all research results obtained in the classical constraint-based paradigm; on the contrary, we fully exploit such algorithmic results. In other terms, our proposal is merely methodological, and it exploits previous research results that were mainly computational.

The paper is organized as follows. In the next Section we briefly review the theory of soft constraints and we define the soft constraint based pattern mining paradigm. In Section 3 we discuss possible alternative instances of the paradigm. In Section 4 we formally define the Soft Constraint Based Pattern Discovery paradigm. We then focus on one of the many possible instances of the proposed paradigm, and we implement it in a concrete Pattern Discovery System. Such a system is built as a wrapper around a classical constraint pattern mining system.

2 Introducing Soft Constraints

Constraint Solving is an emerging software technology for declarative description and effective solving of large problems. Many real life systems, ranging from network management [14] to complex scheduling [2], are analyzed and solved using constraint related technologies. The constraint programming process consists of the generation of requirements (constraints) and solution of these requirements, by specialized constraint solvers. When the requirements of a problem are expressed as a collection of boolean predicates over variables, we obtain what is called the *crisp* (or classical) Constraint Satisfaction Problem (CSP). In this case the problem is solved by finding any assignment of the variables that satisfies all the constraints.

Sometimes, when a deeper analysis of a problem is required, *soft constraints* are used instead. Several formalizations of the concept of soft constraints are currently available. In the following, we refer to the formalization based on *c-semirings* [5]: a semiring-based constraint assigns to each instantiation of its variables an associated value from a partially ordered set. When dealing with crisp constraints, the values are the boolean [illegible] and [illegible] representing the admissible and/or non-admissible values; when dealing with soft constraints the values are interpreted as preferences/costs. The framework must also handle the combination of constraints. To do this one must take into account such additional values, and thus the formalism must provide suitable operations for combination ($\times$) and comparison ($+$) of tuples of values and constraints. This is why this formalization is based on the mathematical concept of semiring.

Definition 3 (c-semirings [5,3]). *A semiring is a tuple $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ such that: A is a set and $\mathbf{0}, \mathbf{1} \in A$; $+$ is commutative, associative and $\mathbf{0}$ is its unit element; $\times$ is associative, distributes over $+$, $\mathbf{1}$ is its unit element and $\mathbf{0}$ is its absorbing element. A c-semiring ("c" stands for "constraint-based") is a semiring $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ such that $+$ is idempotent with $\mathbf{1}$ as its absorbing element and $\times$ is commutative.*

Definition 4 (soft constraints [5,3]). *Given a c-semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ and an ordered set of variables V over a finite domain D, a constraint is a function which, given an assignment $\eta : V \rightarrow D$ of the variables, returns a value of the c-semiring. By using this notation we define $\mathcal{C} = \eta \rightarrow A$ as the set of all possible constraints that can be built starting from S, D and V.*

In the following we will always use the word semiring as standing for c-semiring, and we will explain this very general concept by the point of view of pattern discovery.

Example 2. Consider again the mining query $\mathcal{Q}$. In this context we have that the ordered set of variables V is $\langle$[illegible]$_{\mathcal{D}}(X), avg(X.weight), sum(X.price)\rangle$, while the domain D is: $D($[illegible]$_{\mathcal{D}}(X)) = \mathbb{N}$, $D(avg(X.weight)) = \mathbb{R}^+$, and $D(sum(X.price)) = \mathbb{N}$. If we consider the classical *crisp* framework (i.e., hard constraints) we have the semiring $S_{Bool} = \langle\{true, false\}, \vee, \wedge,$ [illegible] $\rangle$. A constraint C is a function $V \rightarrow D \rightarrow A$; for instance, [illegible]$_{\mathcal{D}}(X) \rightarrow 1700 \rightarrow true$.

The $+$ operator is what we use to compare tuples of values (or patterns, in our context). Let us consider the relation $\leq_S$ (where S stands for the specified semiring) over

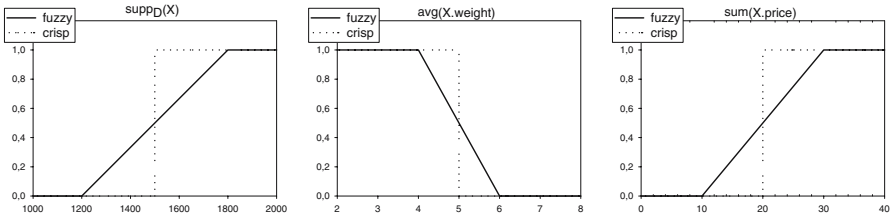
A such that $a \leq_S b$ iff $a + b = b$. It is possible to prove that: $\leq_S$ is a partial order; $+$ and $\times$ are monotone on $\leq_S$; $\mathbf{0}$ is its minimum and $\mathbf{1}$ its maximum, and $\langle A, \leq_S \rangle$ is a complete lattice with least upper bound operator $+$. In the context of pattern discovery $a \leq_S b$ means that the pattern b is *more interesting* than a, where interestingness is defined by a combination of soft constraints. When using (soft) constraints it is necessary to specify, via suitable combination operators, how the level of interest of a combination of constraints is obtained from the interest level of each constraint. The combined weight (or interest) of a combination of constraints is computed by using the operator $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ defined as $(C_1 \otimes C_2)\eta = C_1\eta \times_S C_2\eta$.

Example 3. If we adopt the classical crisp framework, in the mining query $\mathcal{Q}$ of Example 1 we have to combine the three constraints using the $\wedge$ operator (which is the $\times$ in the boolean semiring S_{Bool}). Consider for instance the pattern $p_1 : \langle 1700, 0.8, 19 \rangle$ for the ordered set of variables $V = \langle \mathcal{D}(X), avg(X.weight), sum(X.price) \rangle$. The first and the second constraint are satisfied leading to the semiring level [illegible], while the third one is not satisfied and has associated level [illegible]. Combining the three values with $\wedge$ we obtain [illegible] $\wedge$ [illegible] $\wedge$ [illegible] $=$ [illegible] and we can conclude that the pattern $\langle 1700, 0.8, 19 \rangle$ is not interesting w.r.t. our purposes. Similarly, we can instead compute level [illegible] for pattern $p_3 : \langle 1550, 2.2, 26 \rangle$ corresponding to an interest w.r.t. our goals. Notice that using crisp constraints, the order between values only says that we are interested to patterns with semiring level [illegible] and not interested to patterns with semiring level [illegible] (that is semiring level [illegible] $\leq_{S_{Bool}}$ [illegible]).

3 Instances of the Semiring

Dividing patterns in *interesting* and *non-interesting* is sometimes not meaningful nor useful. Most of the times we can say that each pattern is interesting with a specific level of preference. Soft constraints can deal with preferences by moving from the two values semiring S_{Bool} to other semirings able to give a finer distinction among patters (see [3] for a comprehensive guide to the semiring framework). For our scope the fuzzy and the weighted semiring are the most suitable.

Example 4 (fuzzy semiring). When using fuzzy semiring [12,23], to each pair constraint-pattern is assigned an interest level between 0 and 1, where 1 represents the best value (maximum interest) and 0 the worst one (minimum interest). Therefore the $+$ in this semiring is given by the max operator, and the order $\leq_S$ is given by the usual $\leq$ on real numbers. The value associated to a pattern is obtained by combining the constraints using the minimum operator among the semiring values. Therefore the $\times$ in this semiring is given by the min operator. Recapitulating, the fuzzy semiring is given by $S_F = \langle [0, 1], max, min, 0, 1 \rangle$. The reason for such a max-min framework relies on the attempt to maximize the value of the least preferred tuple. Fuzzy soft constraints are able to model partial constraint satisfaction [13], so to get a solution even when the problem is overconstrained, and also prioritized constraints, that is, constraints with different levels of importance [10]. Figure 1 reports graphical representations of possible fuzzy instances of the constraints in $\mathcal{Q}$. Consider, for instance, the graphical representation of the frequency constraint in Figure 1(C_1). The dotted line describes the behavior



(C_1) (C_2) (C_3)

Fig. 1. Graphical representation of possible fuzzy instances of the constraints in $\mathcal{Q}$

of the *crisp* version (where $1 = true$ and $0 = false$) of the frequency constraint, while the solid line describes a possible fuzzy instance of the same constraint. In this instance domain values smaller than 1200 yield 0 (uninteresting patterns); from 1200 to 1800 the interest level grows linearly reaching the maximum value of 1. Similarly the other two constraints in Figure 1(C_2) and (C_3). In this situation for the pattern $p_1 = \langle 1700, 0.8, 19\rangle$ we obtain that: $C_1(p_1) = 1$, $C_2(p_1) = 1$ and $C_3(p_1) = 0.45$. Since in the fuzzy semiring the combination operator $\times$ is min, we got that the interest level of p_1 is 0.45. Similarly for p_2 and p_3:

- $p_1 : C_1 \otimes C_2 \otimes C_3(1700, 0.8, 19) = min(1, 1, 0.45) = 0.45$
- $p_2 : C_1 \otimes C_2 \otimes C_3(1550, 4.8, 54) = min(1, 0.6, 1) = 0.6$
- $p_3 : C_1 \otimes C_2 \otimes C_3(1550, 2.2, 26) = min(1, 1, 0.8) = 0.8$

Therefore, with this particular instance we got that $p_1 \leq_{S_F} p_2 \leq_{S_F} p_3$, i.e., p_3 is the most interesting pattern among the three.

Example 5 (weighted semiring). While fuzzy semiring associate a level of preference with each tuple in each constraint, in the weighted semiring tuples come with an associated cost. This allows one to model optimization problems where the goal is to minimize the total cost (time, space, number of resources, ...) of the proposed solution. Therefore, in the weighted semiring the cost function is defined by summing up the costs of all constraints. According to the informal description given above, the weighted semiring is $S_W = \langle \mathbb{R}^+, min, sum, +\infty, 0\rangle$. Consider, for instance, the graphical representation of the constraints in the query $\mathcal{Q}$ in Figure 2. In this situation we got that:

- $p_1 : C_1 \otimes C_2 \otimes C_3(1700, 0.8, 19) = sum(50, 20, 205) = 275$
- $p_2 : C_1 \otimes C_2 \otimes C_3(1550, 4.8, 54) = sum(200, 120, 30) = 350$
- $p_3 : C_1 \otimes C_2 \otimes C_3(1550, 2.2, 26) = sum(200, 55, 190) = 445$

Therefore, with this particular instance we got that $p_3 \leq_{S_W} p_2 \leq_{S_W} p_1$ (remember that the order $\leq_{S_W}$ correspond to the $\geq$ on real numbers). In other terms, p_1 is the most interesting pattern w.r.t. this constraints instance.

The weighted and the fuzzy paradigm, can be seen as two different approaches to give a meaning to the notion of optimization. The two models correspond in fact to two definitions of social welfare in utility theory [19]: *"egalitarianism"*, which maximizes the minimal individual utility, and *"utilitarianism"*, which maximizes the sum of the

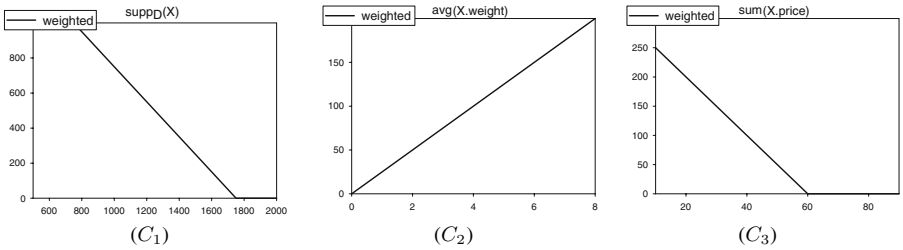


Fig. 2. Graphical representation of possible weighted instances of the constraints in $\mathcal{Q}$

individual utilities. The fuzzy paradigm has an egalitarianistic approach, aimed at maximizing the overall level of interest while balancing the levels of all constraints; while the weighted paradigm has an utilitarianistic approach, aimed at getting the minimum cost globally, even though some constraints may be neglected presenting a big cost. We believe that both approaches present advantages and drawbacks, and may preferred to the other one depending on the application domain. Beyond the fuzzy and the weighted, many other possible instances of the semiring exist, and could be useful in particular applications. Moreover, it is worth noting that the cartesian product of semirings is a semiring [5] and thus it is possible to use the framework also to deal with multicriteria pattern selection.

Finally, note that the soft constraint framework is very general, and could be instantiated not only to unary constraints (as we do in this paper) but also to binary and k-ary constraints (dealing with two or more variables). This could be useful to extend the soft constraint based paradigm to association rules with "2-var" constraints [18].

4 Soft Constraint Based Pattern Mining

In this Section we instantiate soft constraint theory to the pattern discovery framework.

Definition 5 (Soft Constraint Based Pattern Mining). *Let $\mathcal{P}$ denote the domain of possible patterns. A soft constraint on patterns is a function $\mathcal{C} : \mathcal{P} \rightarrow A$ where A is the carrier set of a semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$. Given a combination of soft constraints $\otimes\mathcal{C}$, we define two different problems:*

λ-interesting: *given a minimum interest threshold $\lambda \in A$, it is required to mine the set of all λ-interesting patterns, i.e., $\{p \in \mathcal{P} | \otimes \mathcal{C}(p) \geq \lambda\}$.*

top-k: *given a threshold $k \in \mathbb{N}$, it is required to mine the top-k patterns $p \in \mathcal{P}$ w.r.t. the order $\leq_S$.*

Note that the Soft Constraint Based Pattern Mining paradigm just defined, has many degrees of freedom. In particular, it can be instantiated: (i) on the domain of patterns $\mathcal{P}$ in analysis (e.g., itemsets, sequences, trees or graphs), (ii) on the semiring $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ (e.g., fuzzy, weighted or probabilistic), and (iii) on one of the two possible mining problems, i.e., λ-interesting or top-k mining.

In the rest of this paper we will focus on concretizing a simple instance of this very general paradigm: λ-interesting$_{fuzzy}$ on the pattern class of itemsets.

4.1 Mining λ-Interesting Itemsets on the Fuzzy Semiring

Definition 6. *Let $\mathcal{I} = \{x_1, ..., x_n\}$ be a set of items, where an item is an object with some predefined attributes (e.g., price, type, etc.). A soft constraint on itemsets, based on the fuzzy semiring, is a function $\mathcal{C} : 2^{\mathcal{I}} \rightarrow [0, 1]$. Given a combination of such soft constraints $\otimes\mathcal{C} \equiv \mathcal{C}_1 \otimes \ldots \otimes \mathcal{C}_n$, we define the interest level of an itemset $X \in 2^{\mathcal{I}}$ as $\otimes\mathcal{C}(X) = min(\mathcal{C}_1(X), \ldots, \mathcal{C}_n(X))$. Given a minimum interest threshold $\lambda \in]0, 1]$, the λ-interesting itemsets mining problem, requires to compute $\{X \in 2^{\mathcal{I}} | \otimes\mathcal{C}(X) \geq \lambda\}$.*

In the following we describe how to build a concrete *pattern discovery system* for λ-interesting$_{fuzzy}$ itemsets mining, as a wrapper around a classical constraint pattern mining system. The basic components which we use to build our system are the following:

A *crisp* constraints solver - i.e., a system for mining constrained frequent itemsets, where constraints are classical binary functions, and not soft constraints. Or in other terms, a system for solving the problem in Definition 2. To this purpose we adopt the system which we have developed at Pisa KDD Laboratory within the P^3D project[1]. Such a system is a general Apriori-like algorithm which, by means of *data reduction* and *search space pruning*, is able to push a wide variety of constraints (practically all possible kinds of constraints which have been studied and characterized so far [9]) into the frequent itemsets computation. Based on the algorithmic results developed in the last years by our lab (e.g., [6,7,8,9,21]), our system is very efficient and robust, and to our knowledge, is the unique existing implementation of this kind.

A language of constraints - to express, by means of queries containing conjunctions of constraints, what is interesting for the given application. The wide repertoire of constraints that we admit, comprehends the frequency constraint ($\mathcal{D}(X) \geq \sigma$), and all constraints defined over the following aggregates[2]: $min, max, count, sum, range, avg, var, median, std, md$.

A methodology to define the interest level - that must be assigned to each pair itemset-constraint. In other terms, we need to provide the analyst with a simple methodology to define how to assign for each constraint and each itemset a value in the interval $[0, 1]$, as done, for instance, by the graphical representations of constraints in Figure 1. This methodology should provide the analyst with a knob to adjust the *softness level* of each constraint in the conjunction, and a knob to set the *importance* of each constraint in the conjunction.

Let us focus on the last point. Essentially we must describe how the user can define the fuzzy behavior of a soft constraint. We restrict our system to constraints which behave as those ones in Figure 1: they return a value which grows linearly from 0 to 1 in a certain interval, while they are null before the interval and equal to 1 after the interval. To describe such a simple behavior we just need two parameters: a value associated to the center of the interval (corresponding to the 0.5 fuzzy semiring value), and a parameter to adjust the width of the interval (and consequently the gradient of the function).

[1] `http://www-kdd.isti.cnr.it/p3d/index.html`

[2] $Range$ is $(max - min)$, var is for variance, std is for standard deviation, md is for mean deviation.

Definition 7. *A soft constraint $\mathcal{C}$ on itemsets, based on the fuzzy semiring, is defined by a quintuple $\langle Agg, Att, \theta, t, \alpha \rangle$, where:*

- *$Agg \in \{supp, min, max, count, sum, range, avg, var, median, std, md\}$;*
- *Att is the name of the attribute on which the aggregate agg is computed (or the transaction database, in the case of the frequency constraint);*
- *$\theta \in \{\leq, \geq\}$;*
- *$t \in \mathbb{R}$ corresponds to the center of the interval and it is associated to the semiring value 0.5;*
- *$\alpha \in \mathbb{R}^+$ is the softness parameter, which defines the inclination of the preference function (and thus the width of the interval).*

In particular, if $\theta = \leq$ (as in Figure 1(C_2)) then $\mathcal{C}(X)$ is 1 for $X \leq (t - \alpha t)$, is 0 for $X \geq (t + \alpha t)$, and is linearly decreasing from 1 to 0 within the interval $[t - \alpha t, t + \alpha t]$. The other way around if $\theta = \geq$ (as, for instance, in Figure 1(C_3)). Note that if the softness parameter α is 0, then we obtain the crisp (or hard) version of the constraint.

Example 6. Consider again the query $\mathcal{Q}$ given in Example 1, and its fuzzy instance graphically described by Figure 1. Such query can be expressed in our constraint language as: $\langle$[illegible]$, \mathcal{D}, \geq, 1500, 0.2\rangle, \langle avg, weight, \leq, 5, 0.2\rangle, \langle sum, price, \geq, 20, 0.5\rangle$.

Since the combination operator $\times$ in min, increasing the importance of a constraint w.r.t. the others in the combination means to force the constraint to return lower values for not really satisfactory patterns. By decreasing the softness parameter α, we increase the gradient of the function making the shape of the soft constraint closer to a crisp constraint. This translates in a better value for patterns X which were already behaving well w.r.t. such constraint($\mathcal{C}(X) > 0.5$), and in a lower value for patterns which were behaving not so well ($\mathcal{C}(X) < 0.5$). Decreasing the gradient (increasing α) instead means to lower the importance of the constraint itself: satisfying or not satisfying the constraint does not result in a big fuzzy value difference. Additionally, by operating on t, we can increase the "severity" of the constraint w.r.t. those patterns which were behaving not so well. Therefore, the knob to increase or decrease the importance of a constraint is not explicitly given, because its role, in the fuzzy semiring, can be played by a combined action on the two knobs α and t.

Example 7. Consider again the query $\mathcal{Q}$ given in Example 1, and its fuzzy instance: $\langle$[illegible]$, \mathcal{D}, \geq, 1500, 0.2\rangle, \langle avg, weight, \leq, 5, 0.2\rangle, \langle sum, price, \geq, 20, 0.5\rangle$. As we stated in Example 4, it holds that $p_2 \leq_{S_F} p_3$. In particular, p_2 is better than p_3 w.r.t. constraint C_3, while p_3 is better than p_2 w.r.t. constraint C_2. Suppose now that we increase the importance of C_3, e.g., $\langle sum, price, \geq, 28, 0.25\rangle$. We obtain that $p_3 \leq_{S_F} p_2$:

- $p_2 : C_1 \otimes C_2 \otimes C_3(1550, 4.8, 54) = min(1, 0.6, 1) = 0.6$
- $p_3 : C_1 \otimes C_2 \otimes C_3(1550, 2.2, 26) = min(1, 1, 0.35) = 0.35$

In [5,4] it has been proved that, when dealing with the fuzzy framework, computing all the solution better than a threshold λ can be performed by solving a crisp problem where all the constraint instances with semiring level lower than λ have been assigned level[illegible], and all the instances with semiring level greater or equal to λ have been assigned level[illegible]. Using this theoretical result, and some simple arithmetic we can transform each soft constraint in a corresponding crisp constraint.

Definition 8. *Given a fuzzy soft constraint $\mathcal{C} \equiv \langle Agg, Att, \theta, t, \alpha \rangle$, and a minimum interest threshold λ, we define the crisp translation of $\mathcal{C}$ w.r.t. λ as:*

$$\mathcal{C}^{\lambda}_{crisp} \equiv \begin{cases} Agg(Att) \geq t - \alpha t + 2\lambda\alpha t, & \text{if } \theta = \geq \\ Agg(Att) \leq t + \alpha t - 2\lambda\alpha t, & \text{if } \theta = \leq \end{cases}$$

Example 8. The crisp translation of the soft constraint $\langle sum, price, \geq, 20, 0.5 \rangle$ is $sum(X.price) \geq 26$ for $\lambda = 0.8$, while it is $sum(X.price) \geq 18$ for $\lambda = 0.4$.

Proposition 1. *Given the vocabulary of items $\mathcal{I}$, a combination of soft constraints $\otimes\mathcal{C} \equiv \mathcal{C}1 \otimes \ldots \otimes \mathcal{C}n$, and a minimum interest threshold λ. Let $\mathcal{C}'$ be the conjunction of crisp constraints obtained by conjoining the crisp translation of each constraint in $\otimes\mathcal{C}$ w.r.t. λ: $\mathcal{C}' \equiv \mathcal{C}1^{\lambda}_{crisp} \wedge \ldots \wedge \mathcal{C}n^{\lambda}_{crisp}$. It holds that: $\{X \in 2^{\mathcal{I}} | \otimes\mathcal{C}(X) \geq \lambda\} = \ .(\mathcal{C}')$.*

Proof (sketch). The soundness of the mapping come from the result in [5]. We here have to only give a justification of the formula in Definition 8. This is done by means of Figure 3(b), that shows a graphical representation of the simple arithmetic problem and its solutions.

Therefore, if we adopt the fuzzy semiring, we can fully exploit a classical constraint-based pattern discovery system (and all algorithmic results behind it), by means of a simple translation from soft to crisp constraints. This is exactly what we have done, obtaining a pattern discovery system based on soft constraints built as a wrapper around a classical constraint-based mining system.

4.2 Experimental Analysis

We have conducted some experiments in order to asses the concrete effects obtained by manipulating the α, t and λ parameters. To this purpose we have compared 5 different instances (described in Figure 3(a)) of the query $\mathcal{Q}$:

$$\langle \ldots, \mathcal{D}, \geq, t, \alpha \rangle \langle avg, weight, \leq, t, \alpha \rangle, \langle sum, price, \geq, t, \alpha \rangle$$

where the transactional dataset $\mathcal{D}$, is the well known RETAIL dataset, donated by Tom Brijs and contains the (anonymized) retail market basket data from an anonymous Belgian retail store[3]; and the two attributes $weight$ and $price$ have been randomly generated with a gaussian distribution within the range $[0, 150000]$.

Figure 3(c) reports the number of solutions for the given five queries at different λ thresholds. Obviously as λ increases the number of solutions shrinks accordingly. This behavior is also reflected in queries evaluation times, reported in Figure 3(d): the bigger is the size of the solution set, the longer is the associated computation.

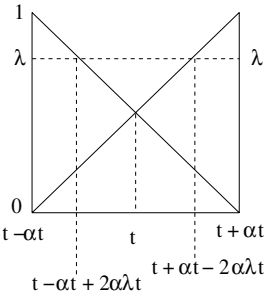
Comparing queries $\mathcal{Q}_1$, $\mathcal{Q}_2$ and $\mathcal{Q}_3$, we can gain more insight about the α parameter. In fact, the three queries differ only by the α associated with one constraint (the frequency constraint). We can observe that, if the λ threshold is not too much selective, increasing the α parameter (i.e., the size of the soft interval), the number of solutions grows. Notice however that, when λ becomes selective enough (i.e., $\lambda > 0.5$), increasing the softness parameter we obtain an opposite behavior. This is due to the fact that, if on one hand a more soft constraint is less severe with patterns not good enough, on the other hand it is less generous with good patterns, which risk to be discarded by an high λ threshold.

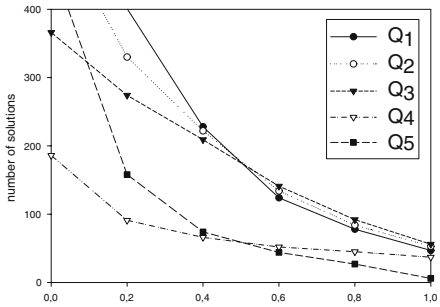
[3] `http://fimi.cs.helsinki.fi/data/`

(a)

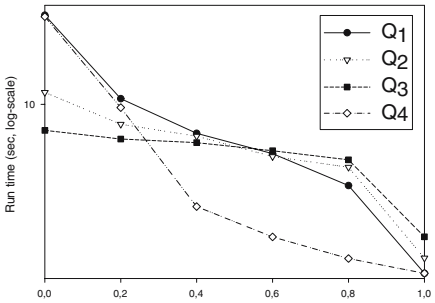
	$\langle supp, \mathcal{D}, \geq, t, \alpha \rangle$		$\langle avg, weight, \leq, t, \alpha \rangle$		$\langle sum, price, \geq, t, \alpha \rangle$	
	t	α	t	α	t	α
$\mathcal{Q}_1$	20	0.8	10000	0.5	20000	0.5
$\mathcal{Q}_2$	20	0.5	10000	0.5	20000	0.5
$\mathcal{Q}_3$	20	0.2	10000	0.5	20000	0.5
$\mathcal{Q}_4$	20	0.8	5000	0.2	20000	0.5
$\mathcal{Q}_5$	20	0.8	5000	0.8	20000	0.5

(b)





(c)



(d)

Fig. 3. (a) description of queries experimented, (b) graphical proof to Proposition 1, (c) and (d) experimental results with λ ranging in $[0, 1]$

References

1. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the Twentieth International Conference on Very Large Databases (VLDB'94)*, 1994.
2. J. Bellone, A. Chamard, and C. Pradelles. Plane - an evolutive planning system for aircraft production. In *Proc. 1st Interantional Conference on Practical Applications of Prolog (PAP92)*, 1992.
3. S. Bistarelli. *Semirings for Soft Constraint Solving and Programming*, volume 2962 of *Lecture Notes in Computer Science*. Springer, 2004.
4. S. Bistarelli, P. Codognet, and F. Rossi. Abstracting soft constraints: Framework, properties, examples. *Artificial Intelligence*, (139):175–211, July 2002.
5. S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based Constraint Solving and Optimization. *Journal of the ACM*, 44(2):201–236, Mar 1997.
6. F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi. ExAMiner: Optimized level-wise frequent pattern mining with monotone constraints. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03)*, 2003.
7. F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi. ExAnte: Anticipated data reduction in constrained pattern mining. In *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03)*, 2003.
8. F. Bonchi and C. Lucchese. On closed constrained frequent pattern mining. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, 2004.

9. F. Bonchi and C. Lucchese. Pushing tougher constraints in frequent pattern mining. In *Proceedings of the Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'05)*, Hanoi, Vietnam, 2005.
10. A. Borning, M. Maher, A. Martindale, and M. Wilson. Constraint hierarchies and logic programming. In *Proc. 6th International Conference on Logic Programming*, 1989.
11. C. Bucila, J. Gehrke, D. Kifer, and W. White. DualMiner: A dual-pruning algorithm for itemsets with constraints. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD'02)*, 2002.
12. D. Dubois, H. Fargier, and H. Prade. The calculus of fuzzy restrictions as a basis for flexible constraint satisfaction. In *Proc. IEEE International Conference on Fuzzy Systems*, pages 1131–1136. IEEE, 1993.
13. E. Freuder and R. Wallace. Partial constraint satisfaction. *AI Journal*, 58, 1992.
14. T. Frühwirth and P. Brisset. Optimal planning of digital cordless telecommunication systems. In *Proc. PACT97*, London, UH, 1997.
15. J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-based, multidimensional data mining. *Computer*, 32(8):46–50, 1999.
16. R. Hilderman and H. Hamilton. *Knowledge Discovery and Measures of Interest*. Kluwer Academic, Boston, 2002.
17. S. Kramer, L. D. Raedt, and C. Helma. Molecular feature mining in hiv data. In *Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining, (SIGKDD'01)*, 2001.
18. L. V. S. Lakshmanan, R. T. Ng, J. Han, and A. Pang. Optimization of constrained frequent set queries with 2-variable constraints. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'99)*, 1999.
19. H. Moulin. *Axioms for Cooperative Decision Making*. Cambridge University Press, 1988.
20. R. T. Ng, L. V. S. Lakshmanan, J. Han, and A. Pang. Exploratory mining and pruning optimizations of constrained associations rules. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD'98)*, 1998.
21. S. Orlando, P. Palmerini, R. Perego, and F. Silvestri. Adaptive and Resource-Aware Mining of Frequent Sets. In *Proc. of the 2002 IEEE Int. Conference on Data Mining (ICDM'02)*, pages 338–345, Maebashi City, Japan, Dec. 2002.
22. J. Pei and J. Han. Can we push more constraints into frequent pattern mining? In *Proceedings of the 6th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD'00)*, 2000.
23. Z. Ruttkay. Fuzzy constraint satisfaction. In *Proc. 3rd IEEE International Conference on Fuzzy Systems*, pages 1263–1268, 1994.
24. S. Sahar. Interestingness via what is not interesting. In *Proc. of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'99)*.
25. R. Srikant, Q. Vu, and R. Agrawal. Mining association rules with item constraints. In *Proceedings of the 3rd ACM International Conference on Knowledge Discovery and Data Mining, (SIGKDD'97)*, 1997.
26. P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proc. of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'2002)*.

Generating Dynamic Higher-Order Markov Models in Web Usage Mining

Jo e Bo ge [1] and Ma Levene[2]

[1] School of Engineering, University of Porto,
R. Dr. Roberto Frias, 4200 - Porto, Portugal
jlborges@fe.up.pt

[2] School of Computer Science and Information Systems, Birkbeck,
University of London, Malet Street, London WC1E 7HX, UK
mlevene@dcs.bbk.ac.uk

Abstract. Markov models have been widely used for modelling users' web navigation behaviour. In previous work we have presented a dynamic clustering-based Markov model that accurately represents second-order transition probabilities given by a collection of navigation sessions. Herein, we propose a generalisation of the method that takes into account higher-order conditional probabilities. The method makes use of the state cloning concept together with a clustering technique to separate the navigation paths that reveal differences in the conditional probabilities. We report on experiments conducted with three real world data sets. The results show that some pages require a long history to understand the users choice of link, while others require only a short history. We also show that the number of additional states induced by the method can be controlled through a probability threshold parameter.

1 Introduction

Modelling e web naviga ion da a i a challenging a ha i con in ing o gain i o ance a he i e of he web and i e -ba e inc ea e. Da a cha -ac e i ing web naviga ion can be collec ed f o e ve o clien -ba ed log le , enabling he econ c ion of e naviga ion e ion [15]. A e ion i ally de ned a a e ence of age viewed by a e wi hin a given i e window. The ba ea ha die e hod o ex ac a e n f o naviga ion da a ha been called , , , , , , , and ch e hod have been a lied in eve al con-ex incl ding e onali a ion, lin edic ion, e-co e ce analy i , ada ive web i e o gani a ion and web age e-fe ching [10].

Seve al a ho have o o ed he e of Ma ov odel o e e en a collec ion of e web naviga ion e ion . Pi ow e al. [12] o o ed a e hod o ind ce he collec ion of longe e ea ing b- e ence , while De h ande e al. [7] o o ed a echni e ha b ild $k^{th}-order$ Ma ov odel and co bine he o incl de he highe o de odel cove ing each a e. On he o he hand, Sa ai [13] e en ed a dy howing ha Ma ov odel have o en ial e in lin edic ion a lica ion , while Zh e al. [16] infe ed a Ma ov odel f o e naviga ion da a o ea e age co-ci a ion and co ling i ila i y.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 34–45, 2005.

An al e na ive e hod of odeling naviga ion e ion a e ee-ba ed odel . Schech e e al. [14] e a ee-ba ed da a c e ha e e en he collec ion of a h infe ed f o log da a o edic he nex age acce ed, while Dong han and J nyi [8] o o ed a hyb id-o de ee-li e Ma ov odel o edic web age acce . In addi ion, Chen and Zhang [6] e a P edic ion by Pa ial Ma ch ee ha e ic he oo o o la node .

In evio wo we o o ed o odel a collec ion of e web naviga ion e ion a a Hy e ex P obabili ic G a a (HPG) [1,2]. A HPG co e ond o a -o de Ma ov odel, which a e e of he N-g a conce [5] o achieve inc ea ed acc acy by inc ea ing he o de of he Ma ov chain; fo he f ll de ail on he HPG conce ee [2]. In [2] an algo i h o ex ac he o f e en ave ed a h f o e da a wa o o ed, and in [3] we have hown ha he algo i h ' co lexi y i , on ave age, linea i e in he n be of a e of he g a a . In [4] we ex ended he HPG odel wi h a dyna ic cl e ing-ba ed e hod ha e a e cloning [9] o acc a ely e e en econd-o de condi ional obabili ie ; he e hod i e en ed in Sec ion 2. In hi wo we gene ali e he e hod given in [4] o highe -o de condi ional obabili ie .

Mo c en web ining y e e echni e ch a cl e ing, a oci-a ion le ining and e en ial a e n ining o ea ch fo a e n in navi-ga ion eco d [10], and do no a e in o acco n he o de in which age we e acce ed. Thi li i a ion ha been ac led by b ilding a e ence of highe -o de Ma ov odel wi h a e hod ha choo e he be odel o e in each ca e [7]. Howeve , we a g e ha a e hod o od ce a ingle odel e e en ing he va iable leng h hi o y of age ha , o fa , been i ing.

The e hod we o o e in Sec ion 3 ai o ll ha ga . By ing he cloning o e a ion we d lica e a e co e onding o age ha e i e a longe hi o y o nde and he choice of lin ha e ade. In hi way he o -lin f o a given a e e flec he n-o de condi ional obabili ie of he in- a h o he a e. In addi ion, he o o ed odel ain ain he f nda en al o e ie of he HPG odel [1], while oviding a i able la fo fo ili ing an algo i h fo ining he naviga ion a e n ha a e in o acco n he o de of age view .

In Sec ion 2 we eview he e en ial of he dyna ic cl e ing e hod, in Sec ion 3 we ex end he e hod o odel highe -o de obabili ie , and in Sec ion 4 we e en he ex e i en al e l . Finally, in Sec ion 5 we give o concl ding e a .

2 Background

In evio wo [1,2] we o o ed o odel e naviga ion da a a Hy e ex P obabili ic G a a (HPG), which co e ond o a -o de Ma ov odel. We now eview he HPG odel wi h he aid of an exa le.

Con ide a web i e wi h even web age , $\{A_1, A_2, \ldots, A_7\}$, and he collec-ion of naviga ion e ion given on he lef - ide of Fig e 1 (NOS e e en he n be of occ ence of each e ion). A naviga ion e ion give i e o a e-

ence of age viewed by a e, wi hin a given i e window. To each web age he, e co, e ond a a e in he odel. In addi ion, he a, a e, S, e, e en he , a e of eve, y naviga ion e ion, and he a nal a e, F, e, e en he la a e of eve, y naviga ion e ion. The, e i a, an i ion co, e onding o each ai, of age vi i ed in e ence, a, an i ion f, o S o he , a e of a e ion, and a, an i ion f, o he la a e of a e ion o F. The odel i inc, e en ally b il by, oce ing he co le e collec ion of naviga ion e ion .

Session	NOS
A_1, A_2, A_3	3
A_1, A_2, A_4	2
A_5, A_2, A_3	1
A_5, A_2, A_4	1
A_6, A_2, A_3	2
A_6, A_2, A_4	1
A_1, A_2, A_4, A_7	1
A_5, A_2, A_4, A_7	3
A_6, A_2, A_4, A_7	2

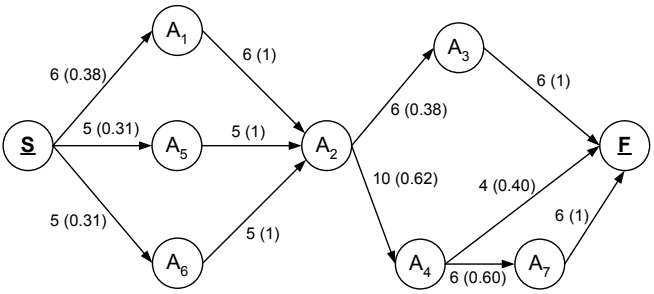


Fig. 1. A collection of user navigation sessions and the corresponding first-order model

A, an i ion, obabili y i e i a ed by he, a io of he n be, of i e he, an i ion wa, ave, ed and he n be, of i e he ancho, a e wa vi i ed. The, igh - ide of Fig, e 1 how a, e, e en a ion of he , -o, de, odel co, e- onding o he in e ion . Nex o a lin , he , n be, give he n be, of i e he lin wa, ave, ed and he n be, in a, en he e give i e i a ed, obabili y.

In [4] we, o o ed a e hod o inc, ea e he HPG, eci ion in o, de, o acc -, a ely, e, e en econd-o, de, obabili ie . The e hod a e e of a cloning o e, a ion, whe, e a a e i d lica ed if , -o, de, obabili ie dive, ge f, o he co, e onding econd-o, de, obabili ie . In addi ion, he e hod e a cl - e, ing algo, i h o iden ify he be way o di, ib e a a e' in-lin be ween a a e and i clone . We now, e en he e en ial, o e, ie of he odel, o o ed in [4], which we ex end o highe, -o, de, obabili ie in Sec ion 3.

Given a odel wi h a e $\{S, A_1, ..., A_n, F\}$, we le w_i, e, e en he n be, of i e he age co, e onding o A_i wa vi i ed, $w_{i,j}$ be he n be, of i e he lin f, o A_i o A_j wa, ave, ed, and $w_{i,j,k}$ be he n be, of i e he e ence A_i, A_j, A_k wa, ave, ed. In addi ion, we le $p_{i,j} = w_{i,j}/w_i$ be he , -o, de, an i ion, obabili y f, o A_i o A_j, and $p_{i,k\,j} = w_{i,k,j}/w_{i,k}$, be he econd-o, de, an i ion, obabili y. Al o, he acc, acy h, e hold, γ, e he highe ad iible diffe, ence be ween a , -o, de, and a econd-o, de, obabili y; a odel i aid o be , . if he, e i no lin ha viola e he con, ain e by γ.

In he exa le given in Fig, e 1, he e, ' a naviga ion behavio, i lie ha $p_{1,23} = p_{1,24} = 0.5$. The, efo, e, fo, $\gamma = 0.1$, he a e A_2 i no acc, a e, ince $|p_{1,2\,3} - p_{2,3}| > 0.1$, and need o be cloned. To clone a e A_2, we le each in-lin

de ne a vec o of . econd-o de obabili ie ; each of he vec o ' co onen . co e ond o an o -lin f o a e A_2. In he exa le, a e A_2 ha h ee in-lin and wo o -lin , ind cing h ee vec o of . econd-o de obabili ie : fo $i = \{3, 4\}$ we have $P_{1,2i} = \{0.5, 0.5\}$, $P_{5,2i} = \{0.2, 0.8\}$ and $P_{6,2i} = \{0.4, 0.6\}$.

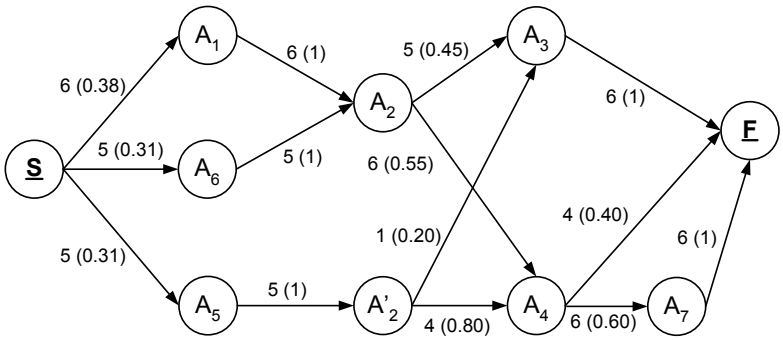


Fig. 2. The second-order HPG model obtained when applying the dynamic clustering method with $\gamma = 0.1$ to the first-order model given in Figure 1

The e hod a lie a k- ean cl e ing algo i h o he collec ion of . econd-o de vec o , in o de o iden ify g o of . i ila vec o . wi h e- ec o γ. Fig e 2 . how he e l of a lying he . e hod o . a e A_2. Since $|p_{1,2\,i} - p_{6,2\,i}| < 0.1$, fo $i = \{3, 4\}$, lin . f o A_1 and A_6 a e a . igned o one clone, he lin f o A_5 i a . igned o he o he clone. The an i ion co n . fo he o -lin . a e da ed a follow : $w_{2,3} = w_{1,2,3} + w_{6,2,3}$, $w_{2,4} = w_{1,2,4} + w_{6,2,4}$, $w_{2',3} = w_{5,2,3}$, $w_{2',4} = w_{5,2,4}$. No e ha , a e A_4 i acc a e, . ince all i . in-lin . have an e ivalen . o ce . a e, and, o eove , eve y . a e having j . one o - lin i acc a e by de ni ion. The efo e, he . odel given in Fig e 2 acc a ely e e en . eve y . econd-o de an i ion obabili y.

3 A Dynamic Clustering Method to Model Higher-Order Probabilities

We now ex end he . e hod e en ed in [4] o inco o a e highe -o de obabili ie . In a . econd-o de HPG . odel, he an i ion obabili ie f o a given . a e a e con ide ed o be acc a e, if all in-lin . o i ind ce iden ical . econd-o de obabili ie . Si ila ly, in a hi d-o de . odel eve y wo-lin a h o a . a e . ind ce iden ical hi d-o de obabili ie . In gene al, o acc a ely . odel n-o de obabili ie each $(n-1)$-leng h a h o a . a e . ind ce iden ical n-o de condi ional obabili ie . E i a e of he n-o de condi ional obabili ie a e ob ained f o he n-g a co n . .

In he following, we le he leng h of a a h be ea ed by he n be of lin . i i co o ed of, and we call he leng h of he a h f o a . a e

o he a ge a e he , .. of hi a e; $d = 0$ co e ond o he a ge a e and $d = n - 1$ co e ond o he fa he a e f o he a ge when a e ing he odel fo o de n. We le $w_{1,\ldots,n}$ e e en he n-g a co n , and $p_{i\ldots j,kt} = w_{i,\ldots,j,k,t}/w_{i,\ldots,j,k}$ e e en he n-o de condi ional obabili y of going o a e A_t given ha he $(n-1)$-leng h a h $A_i, \ldots, A_j, A_k$ wa followed. Al o, we le $\overrightarrow{l}$ e e en a a h and $p_{\overrightarrow{l},kt}$ he condi ional obabili y of an i ion (A_k, A_t) given he a h $\overrightarrow{l}$. Al o, we le $\overrightarrow{l}_{[d]}$ be he a e a de h d on $\overrightarrow{l}$ and $v_{\overrightarrow{l}}$ be he vec o of n-o de condi ional obabili ie given a h $\overrightarrow{l}$. If a a e y need c_y clone , we le y_i, wi h $i = \{1, \ldots, c_y\}$, e e en y and i $c_y - 1$ addi ional clone . Finally, we le $\overrightarrow{l}_c$ be he cl e o which a h $\overrightarrow{l}$ wa a igned.

Fo a a e x, he n-o de condi ional obabili ie a e a e ed in h ee e :

(i) A ly a b ea h- ea ch oced e o ind ce he $(n-1)$-leng h in- a h o a e x, e i a e he co e onding n-o de condi ional obabili ie and, fo each a h, $\overrightarrow{l}$, o e he condi ional obabili ie in a vec o $v_{\overrightarrow{l}}$ (he vec o ' di en ion i given by he n be of o -lin f o x). If he diffe ence be ween a condi ional obabili y and he co e onding an i ion obabili y i g ea e han γ, label he a e a needing o be cloned.

(ii) If x need cloning, a ly he k- ean algo i h o he obabili y vec o , $v_{\overrightarrow{l}}$. The n be of cl e k i inc e en ed n il in he nal ol ion, and in eve y cl e , he di ance be ween each vec o and i cen oid i alle han γ.

(iii) Iden ify a e ha need o be cloned o e a a e he a h o x. S a e incl ded in a h o x a e a e ed in de cending de h o de f o $d = n-1$ o $d = 0$. Fo de h d, we le a e x of a a h o x, who e la a e i y, be na ed a y a h- e x o x. Th , o e a a e a h o x, a e y a de h, d, need a any clone a he n be of di inc a h e xe wi h he a e leng h ha a e a igned o diffe en cl e . The weigh of he in and o -lin of y and i clone a e de e ined by he n-g a co n . Af e cloning y he in- a h o x need o be da ed.

We now e en an exa le of he e hod and a e do-code de c i ion. In a ic la , we eval a e he hi d-o de obabili ie fo he odel in Fig e 2. The condi ional obabili ie ind ced by he a h o A_4 a e: fo $i = \{7, F\}$ we have $p_{12,4i} = \{0.33, 0.67\}$, $p_{62,4i} = \{0.67, 0.33\}$ and $p_{52,4i} = \{0.75, 0.25\}$. Th , ince he e obabili ie a e no clo e o he co e onding econd-o de obabili ie , A_4 i no hi d-o de acc a e fo $\gamma = 0.1$. Table 1 give he in- a h o A_4, he hi d-o de condi ional obabili ie and he e l ing cl e ing a ign en . A e l , a e A_2 fo $d = 1$ need one clone, and fo $d = 0$ a e A_4 al o need one clone. Fig e 3 give he e l ing hi d-o de odel.

In Fig e 3, he a h S, A_1, A_2, A_4 ha obabili y e i a e of $0.38 \cdot 1.00 \cdot 0.50 = 0.19$. I can be een ha in Fig e 1, f o a o al of 16 e ion , 3 begin wi h he 3-g a A_1, A_2, A_4 e l ing in a obabili y e i a e of 0.19. Al o, acco ding o he hi d-o de odel, a h S, A_5, A_2, A_4, A_7 ha obabili y $0.31 \cdot 1.00 \cdot 0.80 \cdot 0.71 = 0.18$. I can be een ha in he in da a 3 e ion

Table 1. The paths to A_4, the third-order conditional probabilities and the resulting clustering assignment

$d=2$	$d=1$	$d=0$	3rd order vectors		cluster
A_1	A_2	A_4	0.33	0.67	1
A_6	A_2	A_4	0.67	0.33	2
A_5	A_2'	A_4	0.75	0.25	2

begin wi h A_5, A_2, A_4, A_7, e l ing in a obabili y e i a e of 0.19. In bo h ca e he diffe ence be ween he wo e i a e i below 0.1, which i he val e eci ed fo he acc acy obabili y h e hold, γ.

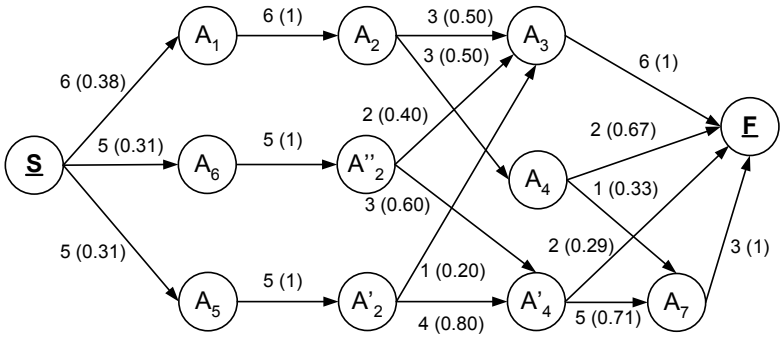


Fig. 3. The third-order model obtained when applying the dynamic clustering method with $\gamma = 0.1$ to the model given in Figure 2

Al e na ively, he ini ial obabili y of a a e can be e i a ed a $w_i / \sum_j w_j$ and eve y a e ha a lin f o S. In Fig e 3 he e i a o al of 54 age view , and, fo exa le, $p_{S,2} = 6/54 = 0.11$ and $p_{S,4'} = 7/54 = 0.13$. Fo he a h A_2, A_4, A_7 he obabili y e i a e i given by he of he obabili ie of a h A_2, A_4, A_7, a h A_2', A_4', A_7 and a h A_2'', A_4', A_7, which i 0.12. In he in e ion , hown in Fig e 1, we have a o al of 54 3-g a co n (incl ding 3-g a a ing wi h S and ending wi h F) and he co n of A_2, A_4, A_7 i 6, he efo e, i e i a e i $6/54 = 0.11$. Fo a h A_1, A_2, A_3 he odel give 0.05, while he e ion analy i give 0.05. Bo h ca e a e acc a e wi h e ec o $\gamma = 0.1$.

The e do-code de c i ion of he algo i h , which i le en he e hod, i now given. We le n be he o de wi h which o eval a e he odel, $HPG_{(n-1)}$ be he evio o de odel, and $(n+1)$-g a be he n-g a co n of i e $n+1$.

```
Algorithm (HPG_(n-1), n, γ, (n+1)-grams)
  begin:
    for each state x
      induce in-paths of length n − 1 to x
      for each in-path l⃗
        for each out-link i from x
          estimate p_(l⃗,xi) and store in v_l⃗
          if (|p_(l⃗,xi) − p_(x,i)| > γ) the state needs to be cloned
        end for
      end for
      if state needs to be cloned
        apply k-means to collection of vectors v_l⃗
        for depth d = (n − 1) to d = 0
          for each state y at depth d
            c_y = num. distinct path prefixes assigned to different clusters
            create c_y − 1 clones of state y
            for each in-path l⃗ to x
              if ( l⃗_[d] = y and l⃗_c > 1) redirect link to corresponding clone
            end for
            for state y_i with i = {1, . . . , c_y}
              for each in-link t to y_i
                for each out-link r from y_i
                  w_(t,y_i) = w_(t,y_i) + w_(t,y_i,r) , w_(y_i,r) = w_(y_i,r) + w_(t,y_i,r)
                end for
              end for
              remove out-links from y_i such that w_(y_i,r) = 0
            end for
            update ngram counts to take into account clones
          end for
          update in-paths with clone references
        end for
      end if
    end for
  end.
```

4 Experimental Evaluation

Fo, he ex e, i en al eval a ion we analy ed h, ee , eal wo, ld da a . e . . By . ing da a f, o h, ee diffe, en . o , ce we ai o a . e . he cha, ac e, i ic of o , . odel in a wide eno gh va, ie y of . cena, io . O , , evio . ex e, ience ha . hown ha i i di c l o c, ea e , ando da a . e . ha . i , o, he cha, ac e, i ic of , eal wo, ld da a, and he, efo, e loo ing a . eve, al da a . e . i nece . a, y.

The . , . da a . e (CS) i f, o a nive, . i y . i e, wa . ade available by he a ho, . of [15] and , e , e en . wo wee . of . age da a in 2002. The . i e wa

coo ie ba ed, age caching wa , ohibi ed and da a wa ade available wi h he e ion iden i ed. We li he da a e in o h ee b e in o de o enhance analy i in a wide va ie y of cena io . The econd da a e (MM) wa ob ained f o he a ho of [11] and co e ond o one on h of age f o he M ic Machine i e (`machines.hyperreal.org`) in 1999 . The da a wa o gani ed in e ion and caching wa di abled d ing collec ion. We li he da a e in o fo b e , each co e onding o a wee of age. The hi d da a e (LTM) e e en fo y day of age f o he London T an o M e web i e in 2003 (`www.ltmuseum.co.uk`). The da a wa ob ained in a aw fo a . We l e ed .gif and .j g e e , and e e wi h an e o a code. Se ion we e de ned a con ec ive e e f o a given IP add e wi hin a 30 in e i e window and a axi e ion leng h of 100 e e wa e . We li hi da a e in o fo b e , each co e onding o en day of age da a.

Table 2 give he a y cha ac e i ic fo each da a e ; iden i e he da a e , , give he n be of di inc age vi i ed, %1 and %≤ 2 indica e, e ec ively, he e cen age of age wi h j one vi i and wi h wo o le vi i . Al o, give he ave age n be of o -lin e a e, he anda d devia ion, he ave age n be of in-lin e age and he anda d devia ion. Finally, give he n be of e ion , he ave age e ion leng h, he anda d devia ion, and he o al n be of e e . The va iabil y on he n be of a e ind ced by he odel fo a given web i e can be ex lained by he n be of age wi h le han one vi i . Al o, when he n be of age wi h few vi i inc ea e he ave age n be of o -lin and in-lin dec ea e . The ave age e ion leng h i able b he anda d devia ion how ha he MM da a ha a highe va iabili y on he e ion leng h.

Table 2. Summary characteristics of the real data sets

ds	pg	$\%1v$	$\%\leq 2v$	aOL	sOL	aIL	sIL	ses	$aSes$	$sSes$	Req
LTM_1	2998	0.62	0.68	4.5	9.6	4.4	11.6	9743	7.6	13.5	74441
LTM_2	1648	0.19	0.27	8.4	13.8	8.3	16.6	11070	7.4	13.2	82256
LTM_3	1610	0.27	0.37	7.8	12.8	7.7	15.0	9116	7.7	13.1	70558
LTM_4	1586	0.24	0.34	7.8	13.3	7.7	15.9	9965	7.8	13.4	78179
MM_1	8715	0.30	0.45	4.7	12.4	4.6	14.1	14734	6.4	37.8	94989
MM_2	5356	0.32	0.44	6.0	18.9	5.9	20.7	14770	6.1	14.7	90682
MM_3	5101	0.26	0.38	6.0	15.6	5.9	17.7	10924	6.7	35.2	73378
MM_4	6740	0.35	0.49	5.1	18.5	4.9	19.8	14080	6.3	23.8	88053
CS_1	3128	0.52	0.67	3.4	10.1	3.1	10.4	7000	4.8	6.5	33854
CS_2	3946	0.59	0.74	2.8	9.3	2.6	9.9	7000	5.0	8.4	34897
CS_3	5028	0.62	0.76	2.8	9.4	2.6	11.6	6950	5.5	12.8	38236

The lef -hand ide of Fig e 4, how , fo he h ee e e en a ive da a e , he va ia ion of he odel i e wi h i o de fo $\gamma = 0$. (The da a e f o each o ce eveal al o iden ical behavio). Fo he MM_1 da a e a la ge e cen age of a e cloning i e fo ed fo econd and hi d-o de obabili ie

which indica e ha he e i no igni can diffe ence be ween hi d-o de ob-
abili ie and he co e onding highe -o de obabili ie , and ha he MM i e
only e i e a ho hi o y when deciding which lin o follow. The CS da a e
how a lowe inc ea e in he odel' i e, and he odel can be een o each
clo e o f ll acc acy wi h e ec o fo h-o de obabili ie . Finally, he LTM
da a e how an inc ea e in he n be of a e fo o he even h-o de
obabili ie , eaning ha he choice of which lin o follow i clea ly infl enced
by a ela ively long e ence of evio ly vi i ed web age .

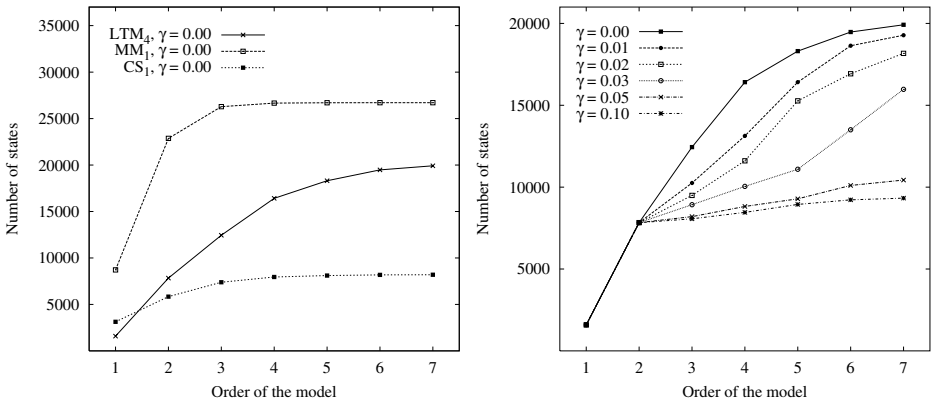


Fig. 4. The increase in model size with the model's order for $\gamma = 0$ and the increase in size for several values of the probability threshold, γ, with the LTM_4 data set

The igh -hand ide of Fig e 4, how he effec of γ, on he odel i e fo he LTM_4 da a e and he lef -hand ide of Fig e 5, how he effec fo he CS_1 da a e . In bo h ca e i can be een ha by ning he val e of γ i i o ible o con ol he inc ea e on a odel ' n be of a e . Fo bo h da a e he diffe ence in he n be of a e i no eviden fo econd-o de odel . Fo hi d and highe -o de odel i i o ible o ed ce he n be of a e ind ced by he e hod by allowing o e ole ance on e e en ing he condi ional obabili ie (by e ing $\gamma > 0$). Se ing γ o a val e g ea e han 0.1 e l in al o no cloning fo highe o de .

Fig e 6 how o e a i ic on he n be of clone e a e fo he LTM_4 and CS_1 da a e , wi h $\gamma = 0.02$. The he ave age n be of clone e a e (avg) i highe fo he LTM_4 da a e han fo he CS_1 da a e , a ex ec ed by in ec ing he lef - ide of Fig e 4. The anda d devia ion (dev) indica e a b an ial va iabili y in he n be of clone e a e, a fac ha i o ed by he axi n be of clone (ax) and he indica ed e cen ile . Fo he LTM_4 da a e 50% of he a e we e neve cloned and 75% have a o ix clone fo he even h o de . In he CS_1 da a e 75% of he a e we e neve cloned and 90% of he a e have a o even clone fo he even h

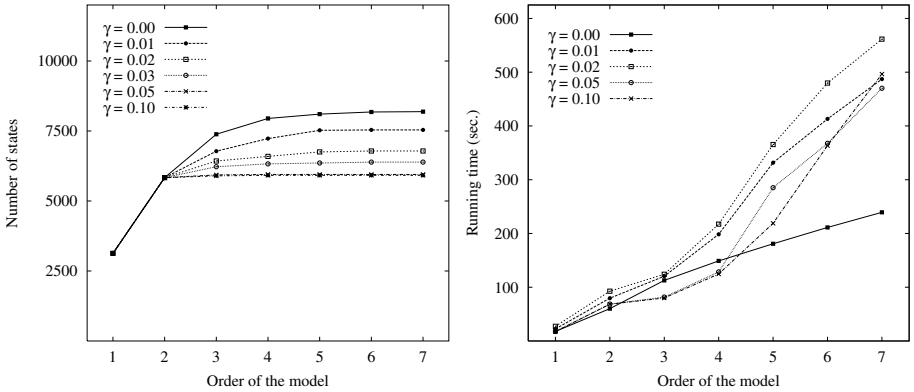


Fig. 5. The increase in model size with the model's order for several values of the probability threshold, γ, for the CS_1 data set and variation of the running time with the model's order for several values of the probability threshold for the LTM_4 data set

o de . The e e l hel o o iva e o in e e in he dyna ic odel, ince while fo o e a e he lin choice of he co e onding age de end on he naviga ion hi o y, fo o he a e he lin choice i co le ely inde enden of he naviga ion hi o y.

	LTM_4 $\gamma = 0.02$					
	order					
	2	3	4	5	6	7
avg	3.94	4.99	6.32	8.63	9.67	10.46
stdev	10.86	15.29	24.85	40.88	47.20	50.69
max	205	307	683	989	1193	1265
75%	4.00	5.00	5.00	6.00	6.00	6.00
85%	10.00	11.25	13.00	14.00	16.25	18.00

	CS_1 $\gamma = 0.02$					
	order					
	2	3	4	5	6	7
avg	0.87	1.06	1.11	1.16	1.17	1.17
stdev	5.40	7.06	7.3	7.92	7.96	7.96
max	138	175	180	208	208	208
75%	0.00	0.00	0.00	0.00	0.00	0.00
95%	4.00	5.00	5.00	5.00	5.00	5.00

Fig. 6. Statistics on the number of clones per state with the model's order for the LTM_4 and CS_1 data set with $\gamma = 0.02$

The igh -hand ide of Fig e 5, and he lef -hand ide of Fig e 7, how o analy i of he nning i e of he algo i h fo wo e e en a ive da a e . We no e ha , while og a ing he e hod, we did no a e a ic la ca e ega ding he i le en a ion e ciency. The e hod i clo e o linea i e fo $\gamma = 0$, ince in ch ca e no cl e ing i needed. Fo $\gamma > 0$ he k- ean e hod i a lied and we le k inc ea e n il a ol ion which ee he h e hold c i e ia i ob ained. Fo he e o ed ex e i en , we e k o va y acco ding o he ex e ion $k = ceiling(1.5k)$ in o de o ob ain a low inc e en on i val e in he age and a la ge inc ea e of he k val e in he b e en age . Finally, he igh -hand ide of Fig e 7, how , fo he LTM_1 da a e , he inc ea e in n be, of a e wi h he odel' o de fo h ee e hod ed

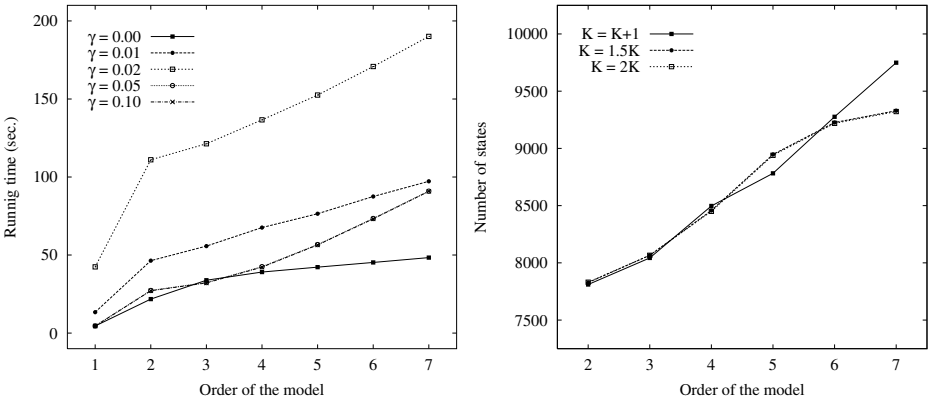


Fig. 7. The running time with the model's order for several values of γ for the CS_1 data set and the increase in model size with the model's order for three methods to set the number of clusters (k) for the LTM_1 data set

o inc ea e k, in he k- ean e hod. The e l how ha fo lowe o de he n be of a e i no ve y en i ive o he e hod, and fo o de highe han ve he fa e e hod ove e i a e he n be of cl e b wi h he bene of a fa e nning i e (which i no hown in he lo).

5 Concluding Remarks

We have o o ed a gene ali a ion of he HPG odel by ing a a e cloning o e a ion ha i able o acc a ely odel highe -o de condi ional obabili ie . The e l ing dyna ic high-o de Ma ov odel i ch ha he obabili ie of he o -lin f o a given a e eflec he n-o de condi ional obabili ie of he a h o he a e. Th , he odel i able o ca e a va iable leng h hi o y of age , whe e diffe en hi o y leng h a e needed o acc a ely odel e naviga ion. In addi ion, he e hod a e e of a obabili y h e hold oge he wi h a cl e ing echni e ha enable o con ol he n be of addi ional a e ind ced by he e hod a he co of o e acc acy. Finally, he odel ain ain he f nda en al o e ie of he HPG odel, [1], oviding a i able la fo fo an algo i h ha can ine naviga ion a e n , a ing in o acco n he o de of age view .

We e o ed on ex e i en wi h h ee di inc eal wo ld da a e . F o he e l we can concl de ha , fo o e web i e e naviga e wi h only a ho hi o y of he age evio ly vi i ed (fo exa le, he MM i e) b in o he i e e hold a longe hi o y in hei e o y (fo exa le, he LTM i e). The e l al o gge ha , in a given i e, diffe en age e i e diffe en a o n of hi o y in o de o nde and he o ible o ion e have when deciding on which lin o clic on. Thi o o in e e in he o o ed

dyna ic e hod ha odel each a e wi h he e i ed hi o y de h. The e l indica e ha he cl e ing e hod i in e e ing fo la ge i e , whe e he n be of a e ind ced fo high o de e , fo $\gamma = 0$, beco e n anageable.

In he ho e we lan o cond c a dy o analy e he e an ic of he le ind ced by diffe en o de obabili y odel . We al o lan o e fo a a i ical co a i on of b e en o de obabili ie ai ed a de e ining if he e i cien a i ical evidence ha he addi ional odel co lexi y in oving o a highe o de j i e he co e onding inc ea e in he algo i h ' co lexi y. I wo ld al o be in e e ing o be able o e i a e he n be of cl e nece a y o achieve he e i ed acc acy in o de o eed he e hod. Finally, a co a a ive dy wi h ee-ba ed odel i al o lanned.

References

1. J. Borges. *A data mining model to capture user web navigation patterns.* PhD thesis, University College London, London University, 2000.
2. J. Borges and M. Levene. Data mining of user navigation patterns. In B. Masand and M. Spliliopoulou, editors, *Web Usage Analysis and User Profiling*, Lecture Notes in Artificial Intelligence (LNAI 1836), pages 92–111. Springer Verlag, 2000.
3. J. Borges and M. Levene. An average linear time algorithm for web usage mining. *Int. Jou. of Information Technology and Decision Making*, 3(2):307–319, June 2004.
4. J. Borges and M. Levene. A dynamic clustering-based markov model for web usage mining. cs.IR/0406032, 2004.
5. Eugene Charniak. *Statistical Language Learning.* The MIT Press, 1996.
6. X. Chen and X. Zhang. A popularity-based prediction model for web prefetching. *Computer*, 36(3):63–70, March 2003.
7. M. Deshpande and G. Karypis. Selective markov models for predicting web page accesses. *ACM Transactions on Internet Technology*, 4(2):163–184, 2004.
8. X. Dongshan and S. Junyi. A new markov model for web access prediction. *Computing in Science and Engineering*, 4(6):34–39, November/December 2002.
9. M. Levene and G. Loizou. Computing the entropy of user navigation in the web. *Int. Journal of Information Technology and Decision Making*, 2:459–476, 2003.
10. B. Mobasher. Web usage mining and personalization. In Munindar P. Singh, editor, *Practical Handbook of Internet Computing.* Chapman Hall & CRC Press, 2004.
11. Mike Perkowitz and Oren Etzioni. Towards adaptive web sites: conceptual framework and case study. *Artificial Intelligence*, 118(2000):245–275, 2000.
12. J. Pitkow and P. Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *Proc. of the 2nd Usenix Symposium on Internet Technologies and Systems*, pages 139–150, Colorado, USA, October 1999.
13. Ramesh R. Sarukkai. Link prediction and path analysis using markov chains. *Computer Networks*, 33(1-6):377–386, June 2000.
14. S. Schechter, M. Krishnan, and M. D. Smith. Using path profiles to predict http requests. *Computer Networks and ISDN Systems*, 30:457–467, 1998.
15. M. Spiliopoulou, B. Mobasher, B. Berendt, and M. Nakagawa. A framework for the evaluation of session reconstruction heuristics in web usage analysis. *IN-FORMS Journal on Computing*, (15):171–190, 2003.
16. J. Zhu, J. Hong, and J. G. Hughes. Using markov models for web site link prediction. In *Proc. of the 13th ACM Conf. on Hypertext and Hypermedia*, pages 169–170, June 2002.

TREE^2 - Decision Trees for Tree Structured Data

Björn Bringmann and Albrecht Zimmermann

Institute of Computer Science, Machine Learning Lab,
Albert-Ludwigs-University Freiburg, Georges-Köhler-Allee 79,
79110 Freiburg, Germany
{bbringma, azimmerm}@informatik.uni-freiburg.de

Abstract. We present TREE^2, a new approach to *structural classification*. This integrated approach induces decision trees that test for pattern occurrence in the inner nodes. It combines state-of-the-art tree mining with sophisticated pruning techniques to find the most discriminative pattern in each node. In contrast to existing methods, TREE^2 uses no heuristics and only a single, statistically well founded parameter has to be chosen by the user. The experiments show that TREE^2 classifiers achieve good accuracies while the induced models are smaller than those of existing approaches, facilitating better comprehensibility.

1 Introduction

Classification is one of the most important data mining tasks. Whereas traditional approaches have focused on flat representations, using feature vectors or attribute-value representations, there has recently been a lot of interest in more expressive representations, such as sequences, trees and graphs [1,2,3,4,5]. Motivations for this interest include drug design, since molecules can be represented as graphs or sequences. Classification of such data paves the way towards drug design on the screen instead of extensive experiments in the lab. Regarding documents, XML, essentially a tree-structured representation, is becoming ever more popular. Classification in this context allows for more efficient dealing with huge amounts of electronic documents.

Existing approaches to classifying structured data (such as trees and graphs) can be categorized into various categories. They differ largely in the way they derive structural features for discriminating between examples belonging to the different classes.

A first category can be described as a pure [illegible] The propositionalization approach typically generates a very large number of features and uses an attribute-value learner to build a classifier. The resulting classifiers are often hard to understand due to the large number of features used which are possibly also combined in a non-trivial way (e.g. in a SVM).

A second class of systems can be described as the [illegible] e.g. Zaki [4]. Even though the resulting rules often yield high predictive accuracy, the number of generated rules typically explodes, making the resulting classifier difficult to understand.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 46–58, 2005.

Both the association rule and propositionalization approaches consider feature generation and classification in two independent steps. [illegible] form a third category of systems that integrates feature construction with classification. This category includes inductive logic programming systems, such as FOIL [6] and PROGOL [7], as well as the DT-GBI approach of Motoda [illegible] [5]. For those approaches to be computationally feasible they have to perform heuristic search, possibly generating non-optimal features.

All techniques mentioned above share the need to specify a number of user-defined parameters, which is often non-trivial.

In this work we present a different approach called TREE^2. It is motivated by recent results on finding correlated patterns, allowing to find the k best features according to a convex optimization criterion such as χ^2 or [illegible] [8]. Rather than generating a large number of features or searching for good features in a heuristic manner, TREE^2 searches for the best features to be incorporated in a decision tree by employing a branch-and-bound search, pruning w.r.t. the best pattern seen so far. As in DT-GBI, a decision tree is induced but at each node, the [illegible] best feature is computed. There are several advantages: TREE^2 is an [illegible], has stronger guarantees than GBI, only one parameter has to be set (the significance level), and the resulting classifiers are far smaller and easier to understand than those of the propositionalization and association rule approaches.

The paper is organized as follows: in Section 2 we describe earlier work on the topic and relate it to our approach; in Section 3, we discuss technical aspects of our method and outline our algorithm; in Section 4, the experimental evaluation is explained and its results discussed. We conclude in Section 5 and point to future work directions.

2 Related Work

[illegible] has been done with different techniques. Firstly, there are several propositionalization approaches, e.g. [2] and [3]. While details may differ, the basic mechanism in these approaches is to first mine all patterns that are unexpected according to some measure (typically frequency). Once those patterns have been found, instances are transformed into bitstrings, denoting occurrence of each pattern. Classifiers are trained using this bitstring representation. While these approaches can show excellent performance and have access to the whole spectrum of machine learning techniques there are possible problems. Obviously the decision which patterns to consider special, e.g. by fixing a minimum frequency, will have an effect on the quality of the model. The resulting feature set will probably be very large, forcing pruning of some kind. Finally, interpretation of the resulting model is not easy, especially if the classifier is non-symbolic, e.g. a SVM.

A second group of approaches is similar to the [illegible] approach [9]. Again, outstanding patterns are mined but each of them has to associate with the class value. Zaki [illegible]'s XRULES classifier is of this variety. Each

pattern is then considered as a rule predicting its class. Usually, the resulting rule set has to be post-processed and/or a conflict resolution technique employed. As in the propositionalization techniques, the choice of constraints under which to mine is not straight-forward and choosing the resolution technique can strongly influence performance, as has been shown e.g. in [10,11]. Additionally, the resulting classifier often consists of thousands of rules, making interpretation by the user again difficult.

Finally, there exist integrated techniques that do not mine patterns, but construct features during building the classifier. Since structural data can be represented in predicate logic, techniques such as FOIL [6] and PROGOL [7] are capable of doing that. While ILP approaches are elegant and powerful, working on large datasets can be too computationally expensive. An approach such as DT-GBI [5], on the other hand, constructs the features it uses for the tests of the induced decision tree by doing graph-mining. What is common to these approaches is that feature induction is usually done in a heuristic way, often by greedy maximization of a correlation measure during beam search. Responsibility of deciding the parameters governing this search is placed upon the user. For instance, in FOIL decisions have to be made on the beam size and the maximum number of literals that are allowed in the rule body. Similarly, DT-GBI requires the user to specify beam size, the maximum number of specializations in each node, and possibly a minimum frequency that should not be violated. As Motoda shows in his work [5], finding the right value for the beam size and the maximum number of specializations requires essentially a meta-search in the space of possible classifiers.

In contrast, the only parameter to be specified for TREE^2 is the cut-off value for growing the decision tree. By basing this value on the p-values for the χ^2-distribution, the user has a well-founded guide-line for choosing this value.

While all the above techniques focus on directly using structural information for classification purposes, a different approach is exemplified by [12]. Instead of explicitly representing the structures used, kernels are employed that quantify similarities between entities. While the resulting classifiers are very accurate, the use of e.g. a graph kernel together with an SVM make analyzing the model difficult.

3 Methodology

In this section we explain the pattern matching notion used by the TREE^2 approach, discuss upper bound calculation, the main component of the principled search for the most discriminating pattern, and formulate the algorithm itself.

3.1 Matching Embedded Trees

Several representations for structured data such as graphs, trees and sequences exist. In this paper we will focus on tree structured data, like XML, only. Thus, we need a notion for matching tree structured data.

A rooted k-tree t is a set of k nodes V_t where each $v \in V_t$, except one called root, has a parent denoted $\pi(v) \in V_t$. We use $\lambda(v)$ to denote the label of a node and an operator $\prec$ to denote the order from left to right among the children of a node. The transitive closure of π will be denoted π^*. Let $\mathcal{L}$ be a formal language composed of all labeled, ordered, rooted trees and $\mathcal{D} \subset \mathcal{L}$ a database. To count trees $t \in \mathcal{D}$ containing a pattern p we define a function $d_t : \mathcal{L} \rightarrow \{0, 1\}$ to be 1 iff p matches the tree t and 0 otherwise.

Several notions of tree matching exist. As in Zaki [illegible]'s work [4] we used a notion called [illegible] which is defined as follows:

Definition 1. [illegible] t [illegible] t' [illegible] $\varphi : V_t \rightarrow V_{t'}$ [illegible] $\forall u, v \in V_t : \lambda(u) = \lambda(\varphi(u)) \wedge u \prec v \Leftrightarrow \varphi(u) \prec \varphi(v) \wedge \pi^*(u) = v \Leftrightarrow \pi^*(\varphi(u)) = \varphi(v)$

An example of an embedded tree is given in Figure 1.

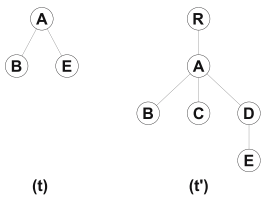


Fig. 1. The tree t is embedded in t'

	c_1	c_2	
T	y_T	$x_T - y_T$	x_T
$\neg T$	$m - y_T$	$n - m - (x_T - y_T)$	$n - x_T$
	m	$n - m$	n

Fig. 2. A Contingency Table

We use [illegible] to compare our approach with Zaki [illegible]'s technique. This notion is more flexible than simple subtrees and the mining process is still efficient. In general, other matching notions (see [1]) and even different representations could be used with our technique . This includes not only other notions of matching trees, but also graphs, sequences etc., since the general principles of our approach apply to all domains.

3.2 Correlation Measures

Popular approaches to finding relevant patterns in the data are based on the support-confidence framework, mining frequent patterns, in the hope of capturing statistically significant phenomena, with high predictive power. This framework has some problems though, namely the difficulty of choosing a "good" support and the fact that confidence tends to reward patterns occurring together with the majority class. To alleviate these problems, we use correlation measures for selecting discriminative patterns. A correlation measure compares the expected frequency of the joint occurrence of a pattern and a certain class value to the observed frequency. If the resulting value is larger than a certain threshold then the deviation from the independence assumption is considered statistically significant enough to assume a relationship between pattern and class label.

Example 1. [illegible] c_1 [illegible] c_2 [illegible] T [illegible] T [illegible] c_1 [illegible] χ^2 [illegible] 4.5 Information Gain [illegible] T [illegible] 0.079

We organize the observed frequencies of a tree pattern T in a contingency table, cf. Figure 2, with x_T denoting the total number of occurences in the dataset and y_T the occurences in the subset corresponding to the first class. Since the two variables are sufficient for calculating the value of a correlation measure on this table, we will view these measures as real-valued functions $\sigma : \mathbb{N}^2 \mapsto \mathbb{R}$ for the remainder of this paper.

While calculating the correlation value of a given pattern is relatively simple, directed search towards better solutions is somewhat more difficult since correlation measures have no desirable properties such as [illegible]. But if they are convex it is possible to calculate an upper bound on the score that can be achieved by specializations of the current pattern T and thus to decide whether this branch in the search tree should be followed.

3.3 Convexity and Upper Bounds

It can be proved that χ^2 and [illegible] are convex. For the proofs of the convexity of χ^2 and [illegible] we refer the reader to [8].

Convex functions take their extreme values at the points forming the convex hull of their domain D. Consider the graph of $f(x)$ in Figure 3(A). Assume the function's domain is restricted to the interval $[k, l]$ which also makes those points the convex hull of D. Obviously, $f(k)$ and $f(l)$ are locally maximal, with $f(l)$ being the global maximum. Given the current value of the function at $f(c)$ and assuming that it is unknown whether c increases or decreases, evaluating f at k and l allows to check whether it is possible for any value of c to put the value of f over the threshold.

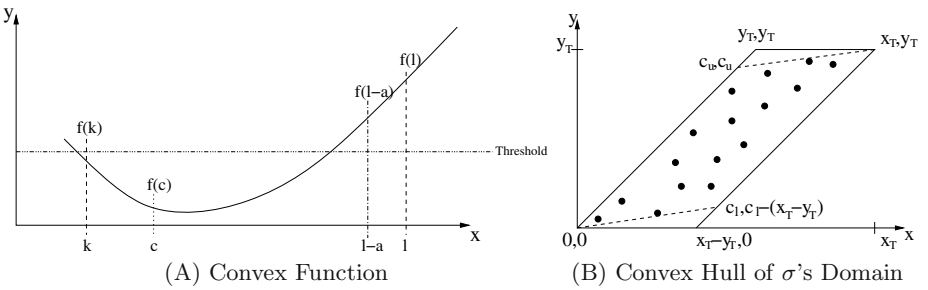


Fig. 3. Convex Function and Convex Hull of the set of possible $\langle x'_T, y'_T \rangle$

For the two-dimensional case, the extreme values are reached at the vertices of the enclosing polygon (in our case the four vertices of the parallelogram in Figure 3(B)). This parallelogram encloses all possible tuples $\langle x'_T, y'_T \rangle$ that correspond to occurence counts of specializations of the current pattern T. The tuple $\langle 0, 0 \rangle$ corresponds to a pattern that does not occur in the dataset and therefore does not have to be considered in calculating the upper bound. $\langle x_T, y_T \rangle$ represents a valid pattern, but in the context of upper bound calculation denotes a specialization of the current pattern T that is equally good in discriminative power. Since general structures have a higher expected probability of being effective on unseen data, we prefer those and thus disregard this tuple as well. Thus the upper bound on $\sigma(T')$ is $ub_\sigma(T) = \max\{\sigma(y_T, y_T), \sigma(x_T - y_T, 0)\}$. For an in-depth discussion of upper bound calculation we refer the reader to [8,11].

Example 2. [illegible] σ [illegible] χ^2, $ub_{\chi^2}(T) = \max\{9.52, 2.08\}$, [illegible] $x = 10$, $y = 8$ [illegible] 9.52 [illegible] $\chi^2(x_T, y_T) = 4.5$ [illegible] T [illegible] T [illegible]

While this upper bound calculation is correct for [illegible], an additional problem w.r.t. χ^2 lies in the fact that the information provided by the score of χ^2 is not always reliable. Statistical theory says that for a contingency table with one degree of freedom, such as the one we are considering here, the expected number of occurrences has to be greater than or equal to 5 for the χ^2 score to be reliable. This means that a χ^2-value on $\langle y_T, y_T \rangle$ or $\langle x_T - y_T, 0 \rangle$ is not necessarily reliable. Thus, upper bound calculation has to be modified to achieve reliability. Based on the size of the class and of $\mathcal{D}$, upper and lower bounds c_u, c_l on x'_T for which all four cells have an expected count of 5 can be calculated and the values of the tuples adjusted accordingly. Two of the new vertices are shown as $\langle c_u, c_u \rangle$ and $\langle c_l, c_l - (x_T - y_T) \rangle$.

3.4 The TREE^2 Algorithm

The TREE^2 algorithm (shown as Algorithm 1) constructs a binary decision tree in the manner of ID3 [13]. In the root node and each inner node, the occurrence of a tree pattern is tested against the instance to be classified. A resulting tree could look like the example given in Figure 4. In each node, the subtree having the best discriminative effect on the corresponding subset is found by a systematic branch-and-bound search. The mining process is shown in the subroutine ENUMERATEBESTSUBTREE. The space of possible patterns is traversed using canonical enumeration and the value of σ calculated for each candidate pattern. If this value lies above the best score seen so far, the current pattern is the most discriminating on this subset so far and the threshold is raised to its σ-value. An upper bound on the value specializations of the current pattern can achieve is calculated and pruning of the search space using this upper bound and the threshold is performed. In this way, we separate the success of the technique from user decisions about the search strategy. The only decision a user has to

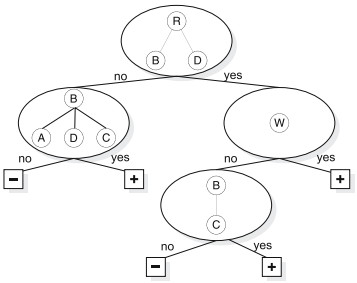


Fig. 4. A decision tree as produced by the TREE^2 algorithm

make is the one w.r.t. a stopping criterion for further growth of the tree. To this effect, a minimum value for the score of the correlation measure has to be specified, which can be based on statistical theory, thus giving the user a better guidance for making this decision.

Algorithm 1 The TREE^2 algorithm

TREE^2($\mathcal{D}, \sigma, \tau$, DT)
1: p = ENUMERATEBESTSUBTREE($\top, 0, \sigma, \tau, \varnothing$)
2: **if** $p \neq \varnothing$ **then**
3: Add node including p to the DT
4: TREE^2($\{T \in \mathcal{D}|p$ embedded in $T\}$, σ, τ, DT)
5: TREE^2($\{T \in \mathcal{D}|p$ not embedded in $T\}$, σ, τ, DT)
6: **return** DT

ENUMERATEBESTSUBTREE(t, τ, σ, τ, p)
1: **for all** canonical expansions t' of t **do**
2: **if** $\sigma(t') > \tau \wedge \sigma(t') \geq \tau$ **then**
3: $p = t', \ \tau = \sigma(t')$
4: **if** $ub(t') \geq \tau$ **then**
5: p = ENUMERATEBESTSUBTREE($t', \tau, \sigma, \tau, p$)
6: **return** p

TREE^2 has several desirable properties. Firstly, the resulting classifier is integrated in the sense that it uses patterns directly, thus circumventing the need for the user to restrict the amount of features and making the resulting classifier more understandable. Secondly, by using correlation measures for quantifying the quality of patterns, we give the user a sounder theoretical foundation on which to base the decision about which learned tests to consider significant and include in the model. Thirdly, we avoid using heuristics that force the user to decide on the values of parameters that could have a severe impact on the resulting model's accuracy. Using principled search guarantees that TREE^2 finds the best discriminating pattern for each node in the decision tree w.r.t. the correlation measure used. Finally, as the experiments show, the resulting decision tree is far smaller than the rule sets produced by XRULES classifier [4], while achieving comparable accuracy, and is therefore more easily interpretable by human users.

4 Experimental Evaluation

For the experimental evaluation, we compared our approach to XRULES and a decision tree base-line approach on the XML data used in Zaki [illegible]'s publication [4]. Furthermore, we compared TREE^2 to a base-line approach using frequency mining for a SVM classifier and two PROGOL results on the regression-friendly subset of the Mutagenesis dataset.

XML Data. The XML data used in our experiments are log files from web-site visitors' sessions. They are separated into three weeks (CSLOG1, CSLOG2, and CSLOG3) and each session is classified as its producing visitor coming either from an `.edu` domain or from any other domain. Characteristics of the datasets are shown in Table 1. For the comparison we built decision trees with the

Table 1. Characteristics of Datasets (taken from [4])

DB	#Sessions	edu	other	%edu	%other
CSLOG1	8074	1962	6112	24.3	75.7
CSLOG2	7409	1687	5722	22.8	77.2
CSLOG12	13934	2969	10965	21.3	78.7
CSLOG3	7628	1798	5830	23.6	76.4

χ^2 distribution's significance value for 90%, 95% and 99% respectively. In each setting we used one set of data for training and another one for testing. Following Zaki's notation, CSLOGx-y means that we trained on set x and tested on set y. For the base-line approach we mined the 100 patterns having the highest discriminative effect on the data, transformed the data into bitstring instances according to the found patterns, and built decision trees using all 100 patterns in one run ([illegible]) and the 50 best patterns in another run ([illegible]) with the WEKA [14] implementation of the C4.5 [15] algorithm. We compare the accuracies of the resulting classifiers against each other as well as the complexity of the model which we measure by the number of rules used by XRULES, and by the number of leaves in the decision trees, which corresponds to the number of rules that can be derived from the trees, respectively.

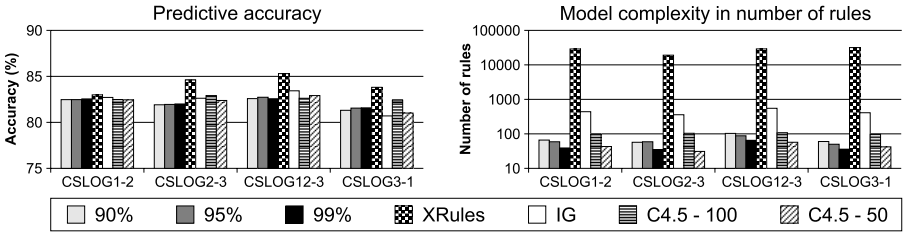


Fig. 5. Accuracies and size in rules of the different approaches

Results are summarized in Figure 5. As can be seen, the accuracies of the induced classifiers do not vary much. The only approach that significantly outperforms (by 2-3%) the other techniques on all but the CSLOG1-2 setting, is XRULES. At the same time, the size of XRULES' models is also significantly greater. While the TREE^2 trees induced with [illegible] have several hundred nodes and all trees induced with χ^2 (both TREE^2 and base-line) between 35 and 103 nodes, the smallest XRULES model consists of more than 19000 rules. Patterns tested against in the inner decision tree nodes consist of 3-7 nodes only. Since this is similar to the size of patterns used in XRULES' rules, complexity is really reduced and not just pushed inside the classifier. In comparing the other approaches, several things are noticeable. Raising the threshold from the 90% to the 95% significance level for χ^2-induced TREE^2 trees does not decrease accuracy (even improving it slightly in 3 cases). Raising it further to the 99% level has no clear effect. The tree size decreases, though, on average by 7.5 nodes from the 90% to the 95% setting. Raising the significance level further to 99% decreases the tree size by 18 nodes on average.

For the base-line approach we mined patterns correlating strongly with the classes and trained a classifier on them. This approach achieves competitive results w.r.t the accuracy. The clear drawback is that deciding on the number of features to use is not straightforward. Using only 50 instead of 100 features produces all kinds of behavior. In some cases the accuracy does not change. In other cases the classifier using 50 features outperforms the one using 100 or vice versa. Also, the base-line approach using 100 patterns tends to use most of these, even if TREE^2 trees of similar quality are much smaller.

Finally, using [illegible] as quality criterion shows mainly one thing - that it is difficult to make an informed decision on cut-off values. The accuracies and sizes shown refer to decision trees induced with a cut-off value of 0.001. For one thing, the resulting trees grow far bigger than the χ^2-trees. Additionally, the accuracies in comparison with the χ^2 approach vary, giving rise to one worse tree, one of equal quality and two better ones. None of the differences in accuracy is significant though. Inducing decision trees with a cut-off value of 0.01 lowers accuracy by 1.5 to 3 percentage points, with the induced trees still being larger than the χ^2 trees.

Mutagenicity Data. For this setting, we chose the regression-friendly subset of the well known Mutagenicity dataset used in [16]. We compare with the results of the ILP system PROGOL reported in [16,17] and the results of the base-line approach reported in [3]. Since the Mutagenicity dataset consists of molecules represented as graphs, a transformation from the SMILES representation into so-called fragment-trees is used that is explained following this paragraph.

[illegible] The SMILES language [18] is used by computational chemists as a compact encoding of molecular structure. It is supported by many tools as OpenBabel or Daylight ([19,20]). The language contains symbols for atoms, bonds, branches, and can express cycles. Using a decomposition-algorithm by Karwath and De Raedt [21], a SMILES-String can, after some

reformatting, be decomposed into a so-called [illegible]. Since there is no [illegible] SMILES-string for a molecule, the fragment tree is not unique either. The decomposition-algorithm recursively splits the string into [illegible] $\{_xT\}_x$ and [illegible] $A(B)C$. In the resulting fragment-tree the leaves contain pure cycles or linear fragments without further branches. The inner nodes of such a tree contain fragments still containing branches while the root node is the whole molecule. The edge labels denote the type of decomposition (i.e. the part of the branch or the number of the cycle). Thus, the leaves of a fragment-tree contain a lot of information decomposed into very small fragments. As in [3] we drop the edge labels and labeled all but the leaf nodes with a new, unique label. Hence, the tree-structure represents the abstract structure of the molecule with the chemical information in the leaves.

Figure 6 shows a molecule on the left-hand side which could be encoded by the SMILES-string $N-c1ccc(cc1)-O-c2ccc(cc2)-[Cl]$. This string represents the same as $N\{_0cccc(cc\}_0)O\{_1cccc(cc\}_1)[Cl]$. The corresponding fragment-tree is shown on the right-hand side of Figure 6.

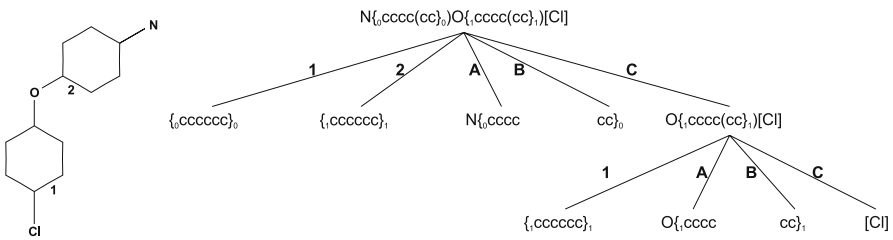


Fig. 6. A molecule with the encoding $N-c1ccc(cc1)-O-c2ccc(cc2)-[Cl]$ and the corresponding fragment-tree

[illegible] Predictive accuracy for each approach was estimated using ten-fold cross-validation. Reported are average accuracies and standard deviation (if known). For TREE^2, trees were induced at the 95% significance level for χ^2 and with a cut-off value of 0.01 for [illegible]. The results reported in [16] were achieved using PROGOL and working only on structural information, in [17], numerical values suggested by experts were used as well. This work reports only an average accuracy. The resulting accuracies and the size of the corresponding theories are shown in Table 2.

As can be seen, for both measures TREE^2 gives similar results to the purely structural PROGOL approach, with the differences being not significant. At the same time, the χ^2 induced model is far smaller than the other two. Again, the patterns tested against in the inner nodes are not overly complex (5-11 nodes). When PROGOL uses the expert-identified attributes as well, its accuracy increases. Since we do not have access to the standard deviation of these experiments, we cannot make a significance statement. Finally, the base-line approach,

Table 2. Accuracies and complexity of the models on the mutagenicity dataset

Approach	Predictive Accuracy	Average Size of the Model
TREE^2 χ^2	80.26±7.14	2.3 Nodes
TREE^2 *IG*	81.76±9	11.8 Nodes
PROGOL '94 [16]	80±3	9 Clauses
PROGOL '95 [17]	84	4 Clauses
FREQUENT SMILES [3]	86.70	214 Patterns

which mined all patterns frequent in one class and not exceeding a given frequency in the other class, and built a model using these features in an SVM, significantly outperforms the TREE^2 classifiers. On the other hand, by using a SVM, the results will hardly be interpretable for humans anymore and the amount of patterns used is larger than in the TREE^2 models by two orders of magnitude.

5 Conclusion and Future Work

We presented TREE^2, an integrated approach to structural classification. The algorithm builds a decision tree for tree structured data that tests for pattern occurrence in the inner nodes. Using an optimal branch-and-bound search, made possible by effective pruning, TREE^2 finds the most discriminative pattern for each subset of the data considered. This allows the user to abstract the success of the classifier from decisions about the search process, unlike in existing approaches that include heuristics. Basing the stopping criterion for growing the decision tree on statistically well founded measures rather than arbitrary thresholds whose meaning is somewhat ambiguous gives the user better guidance for selecting this parameter. It also alleviates the main problem of the support-confidence framework, namely the generation of very large rule sets that are incomprehensible to the user and possibly include uninformative rules w.r.t. classification.

As the experiments show, TREE^2 classifiers are effective while being less complex than existing approaches. While using χ^2 for assessing the quality of discriminative patterns, raising or lowering the significance threshold affects the induced trees in an expected manner. In contrast, using [illegible] is more difficult, since selecting the cut-off value has no statistical foundations. While base-line approaches, that separate feature generation and classifier construction, achieve very good results, it is not entirely clear how to justify the selected the number of features mined. Furthermore, there exists a gap in interpretability since the classifier used might combine the mined features in a way that is not easily accessible to the user.

So far, we have restricted ourselves to a single representation, [illegible], a certain type of classifier, [illegible], and two measures. Future work will include evaluating other correlation measures and applying our approach to different

representations. Finally, the success of using effective conflict resolution strategies in the XRULES classifier suggests the expansion our approach to ensemble classifiers.

Acknowledgments. We would like to thank Mohammed J. Zaki for providing the datasets and the XRULES algorithm. Furthermore, we would like to thank Andreas Karwath, Kristian Kersting, Robert Egginton and Luc De Raedt for interesting discussions and comments to our work.

References

1. Kilpeläinen, P.: Tree Matching Problems with Applications to Structured Text Databases. PhD thesis, University of Helsinki (1992)
2. Kramer, S., Raedt, L.D., Helma, C.: Molecular feature mining in HIV data. In Provost, F., Srikant, R., eds.: Proc. KDD-01, New York, ACM Press (2001) 136–143
3. Bringmann, B., Karwath, A.: Frequent SMILES. In: Lernen, Wissensentdeckung und Adaptivität, Workshop GI Fachgruppe Maschinelles Lernen, LWA. (2004)
4. Zaki, M.J., Aggarwal, C.C.: XRules: an effective structural classifier for XML data. In Getoor, L., Senator, T.E., Domingos, P., Faloutsos, C., eds.: KDD, Washington, DC, USA, ACM (2003) 316–325
5. Geamsakul, W., Matsuda, T., Yoshida, T., Motoda, H., Washio, T.: Performance evaluation of decision tree graph-based induction. In Grieser, G., Tanaka, Y., Yamamoto, A., eds.: Discovery Science, Sapporo, Japan, Springer (2003) 128–140
6. Quinlan, J.R.: Learning logical definitions from relations. Machine Learning **5** (1990) 239–266
7. Muggleton, S.: Inverse entailment and PROGOL. New Generation Computing **13** (1995) 245–286
8. Morishita, S., Sese, J.: Traversing itemset lattices with statistical metric pruning. In: Proceedings of the Nineteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Dallas, Texas, USA, ACM (2000) 226–236
9. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In Agrawal, R., Stolorz, P.E., Piatetsky-Shapiro, G., eds.: KDD, New York City, New York, USA, AAAI Press (1998) 80–86
10. Mutter, S., Hall, M., Frank, F.: Using classification to evaluate the output of confidence-based association rule mining. In Webb, G.I., Yu, X., eds.: Australian Conference on Artificial Intelligence, Cairns, Australia, Springer (2004) 538–549
11. Zimmermann, A., De Raedt, L.: Corclass: Correlated association rule mining for classification. [22] 60–72
12. Gärtner, T., Lloyd, J.W., Flach, P.A.: Kernels and distances for structured data. Machine Learning **57** (2004)
13. Quinlan, J.R.: Induction of decision trees. Machine Learning **1** (1986) 81–106
14. Frank, E., Hall, M., Trigg, L.E., Holmes, G., Witten, I.H.: Data mining in bioinformatics using weka. Bioinformatics **20** (2004) 2479–2481
15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
16. Srinivasan, A., Muggleton, S., King, R., Sternberg, M.: Mutagenesis: ILP experiments in a non-determinate biological domain. In Wrobel, S., ed.: Proceedings of the 4th International Workshop on Inductive Logic Programming. Volume 237., Gesellschaft für Mathematik und Datenverarbeitung MBH (1994) 217–232

17. King, R.D., Sternberg, M.J.E., Srinivasan, A.: Relating chemical activity to structure: An examination of ILP successes. New Generation Comput. **13** (1995) 411–433
18. Weininger, D.: SMILES, a chemical language and information system 1. Introduction and encoding rules. J. Chem. Inf. Comput. Sci. **28** (1988) 31–36
19. The OpenBabel Software Community: Open Babel. http://openbabel.sourceforge.net/ (2003)
20. Daylight Chemical Information Systems, Inc. http://www.daylight.com/ (2004)
21. Karwath, A., De Raedt, L.: Predictive graph mining. [22] 1–15
22. Suzuki, E., Arikawa, S., eds.: Discovery Science, 7th International Conference, DS 2004, Padova, Italy, October 2-5, 2004, Proceedings. In Suzuki, E., Arikawa, S., eds.: DS 2004, Padova, Italy, Springer (2004)

Agglomerative Hierarchical Clustering with Constraints: Theoretical and Empirical Results

Ian Davidson and S.S. Ravi

Department of Computer Science, University at Albany - State University of New York, Albany, NY 12222
{davidson, ravi}@cs.albany.edu

Abstract. We explore the use of instance and cluster-level constraints with agglomerative hierarchical clustering. Though previous work has illustrated the benefits of using constraints for non-hierarchical clustering, their application to hierarchical clustering is not straight-forward for two primary reasons. First, some constraint combinations make the feasibility problem (Does there exist a single feasible solution?) **NP**-complete. Second, some constraint combinations when used with traditional agglomerative algorithms can cause the dendrogram to stop prematurely in a dead-end solution even though there exist other feasible solutions with a significantly smaller number of clusters. When constraints lead to efficiently solvable feasibility problems and standard agglomerative algorithms do not give rise to dead-end solutions, we empirically illustrate the benefits of using constraints to improve cluster purity and average distortion. Furthermore, we introduce the new γ constraint and use it in conjunction with the triangle inequality to considerably improve the efficiency of agglomerative clustering.

1 Introduction and Motivation

Hierarchical clustering algorithms are run once and create a dendrogram which is a tree structure containing a k-block set partition for each value of k between 1 and n, where n is the number of data points to cluster allowing the user to choose a particular clustering granularity. Though less popular than non-hierarchical clustering there are many domains [16] where clusters naturally form a hierarchy; that is, clusters are part of other clusters. Furthermore, the popular agglomerative algorithms are easy to implement as they just begin with each point in its own cluster and progressively join the closest clusters to reduce the number of clusters by 1 until $k = 1$. The basic agglomerative hierarchical clustering algorithm we will improve upon in this paper is shown in Figure 1. However, these added benefits come at the cost of time and space efficiency since a typical implementation with symmetrical distances requires $\Theta(mn^2)$ computations, where m is the number of attributes used to represent each instance.

In this paper we shall explore the use of instance and cluster level constraints with hierarchical clustering algorithms. We believe the use of such constraints with *hierarchical* clustering is the first though there exists work that uses spatial constraints to find specific types of clusters and avoid others [14,15]. The similarly named *constrained hierarchical clustering* [16] is actually a method of combining partitional and hierarchical clustering algorithms; the method does not incorporate apriori constraints. Recent work

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 59–70, 2005.

Agglomerative($S = \{x_1, \ldots, x_n\}$) **returns** $Dendrogram_k$ for $k = 1$ to $|S|$.

1. $C_i = \{x_i\}, \forall i$.
2. **for** $k = |S|$ **down to** 1
   $Dendrogram_k = \{C_1, \ldots, C_k\}$
   $d(i,j) = D(C_i, C_j), \forall i, j; \quad l, m = argmin_{a,b}\, d(a,b)$.
   $C_l = Join(C_l, C_m); \quad Remove(C_m)$.
   endloop

Fig. 1. Standard Agglomerative Clustering

[1,2,12] in the non-hierarchical clustering literature has explored the use of instance-level constraints. The **must-link** and **cannot-link** constraints require that two instances must both be part of or not part of the same cluster respectively. They are particularly useful in situations where a large amount of unlabeled data to cluster is available along with some labeled data from which the constraints can be obtained [12]. These constraints were shown to improve cluster purity when measured against an extrinsic class label not given to the clustering algorithm [12]. The δ **constraint** requires the distance between any pair of points in two different clusters to be at least δ. For any cluster C_i with two or more points, the ϵ**-constraint** requires that for each point $x \in C_i$, there must be another point $y \in C_i$ such that the distance between x and y is at most ϵ. Our recent work [4] explored the computational complexity (difficulty) of the *feasibility* problem: **Given** a value of k, does there exist at least one clustering solution that satisfies all the constraints and has k clusters? Though it is easy to see that there is no feasible solution for the three cannot-link constraints *CL(a,b),CL(b,c),CL(a,c)* for $k < 3$, the general feasibility problem for cannot-link constraints is **NP**-complete by a reduction from the graph coloring problem. The complexity results of that work, shown in Table 1 (2nd column), are important for data mining because when problems are shown to be intractable in the worst-case, we should avoid them or should not expect to find an exact solution efficiently.

We begin this paper by exploring the feasibility of agglomerative **hierarchical** clustering under the above four mentioned instance and cluster-level constraints. This problem is *significantly* different from the feasibility problems considered in our previous work since the value of k for hierarchical clustering is not given. We then empirically show that constraints with a modified agglomerative hierarchical algorithm can improve the quality and performance of the resultant dendrogram. To further improve performance we introduce the γ constraint which when used with the triangle inequality can yield large computation saving that we have bounded in the best and average case. Finally, we cover the interesting result of an irreducible clustering. If we are given a feasible clustering with k_{max} clusters then for certain combination of constraints joining the two closest clusters may yield a feasible but "dead-end" solution with k clusters from which no other feasible solution with less than k clusters can be obtained, even though they are known to exist. Therefore, the created dendrograms may be incomplete.

Throughout this paper $D(x, y)$ denotes the Euclidean distance between two points and $D(X, Y)$ the Euclidean distance between the centroids of two groups of instances. We note that the feasibility and irreducibility results (Sections 2 and 5) are not neces-

Table 1. Results for Feasibility Problems for a Given k (partitional clustering) and Unspecified k (hierarchical clustering)

Constraint	Given k	Unspecified k	Unspecified k - Deadends?
Must-Link	**P** [9,4]	**P**	No
Cannot-Link	**NP**-complete [9,4]	**P**	Yes
δ-constraint	**P** [4]	**P**	No
ϵ-constraint	**P** [4]	**P**	No
Must-Link and δ	**P** [4]	**P**	No
Must-Link and ϵ	**NP**-complete [4]	**P**	No
δ and ϵ	**P** [4]	**P**	No
Must-Link, Cannot-Link, δ and ϵ	**NP**-complete [4]	**NP**-complete	Yes

sarily for Euclidean distances and are hence applicable for single and complete linkage clustering while the γ-constraint to improve performance (Section 4) is applicable to any metric space.

2 Feasibility for Hierarchical Clustering

In this section, we examine the feasibility problem for several different types of constraints, that is, the problem of determining whether the given set of points can be partitioned into clusters so that all the specified constraints are satisfied.

Definition 1. ***Feasibility problem for Hierarchical Clustering*** *(*FHC*)*

Instance: *A set S of nodes, the (symmetric) distance $d(x, y) \geq 0$ for each pair of nodes x and y in S and a collection C of constraints.*

Question: *Can S be partitioned into subsets (clusters) so that all the constraints in C are satisfied?*

When the answer to the feasibility question is "yes", the corresponding algorithm also produces a partition of S satisfying the constraints. We note that the FHC problem considered here is *significantly* different from the constrained non-hierarchical clustering problem considered in [4] and the proofs are different as well even though the end results are similar. For example in our earlier work we showed intractability results for some constraint types using a straightforward reduction from the graph coloring problem. The intractability proof used in this work involves more elaborate reductions. For the feasibility problems considered in [4], the number of clusters is in effect, another constraint. In the formulation of FHC, there are *no* constraints on the number of clusters, other than the trivial ones (i.e., the number of clusters must be at least 1 and at most $|S|$).

We shall in this section begin with the same constraints as those considered in [4]. They are: (a) Must-Link (ML) constraints, (b) Cannot-Link (CL) constraints, (c) δ constraint and (d) ϵ constraint. In later sections we shall introduce another cluster-level

constraint to improve the efficiency of the hierarchical clustering algorithms. As observed in [4], a δ constraint can be efficiently transformed into an equivalent collection of ML-constraints. Therefore, we restrict our attention to ML, CL and ϵ constraints. We show that for any *pair* of these constraint types, the corresponding feasibility problem can be solved efficiently. The simple algorithms for these feasibility problems can be used to seed an agglomerative or divisive hierarchical clustering algorithm as is the case in our experimental results. However, when all three types of constraints are specified, we show that the feasibility problem is **NP**-complete and hence finding a clustering, let alone a good clustering, is computationally intractable.

2.1 Efficient Algorithms for Certain Constraint Combinations

When the constraint set C contains only ML and CL constraints, the FHC problem can be solved in polynomial time using the following simple algorithm.

1. Form the clusters implied by the ML constraints. (This can be done by computing the transitive closure of the ML constraints as explained in [4].) Let $C_1, C_2, \ldots, C_p$ denote the resulting clusters.
2. If there is a cluster C_i ($1 \leq i \leq p$) with nodes x and y such that x and y are also involved in a CL constraint, then there is no solution to the feasibility problem; otherwise, there is a solution.

When the above algorithm indicates that there is a feasible solution to the given FHC instance, one such solution can be obtained as follows. Use the clusters produced in Step 1 along with a singleton cluster for each node that is not involved in an ML constraint. Clearly, this algorithm runs in polynomial time. We now consider the combination of CL and ϵ constraints. Note that there is always a trivial solution consisting of $|S|$ singleton clusters to the FHC problem when the constraint set involves only CL and ϵ constraints. Obviously, this trivial solution satisfies both CL and ϵ constraints, as the latter constraint only applies to clusters containing two or more instances.

The FHC problem under the combination of ML and ϵ constraints can be solved efficiently as follows. For any node x, an **ϵ-neighbor** of x is another node y such that $D(x, y) \leq \epsilon$. Using this definition, an algorithm for solving the feasibility problem is:

1. Construct the set $S' = \{x \in S \ : \ x$ does not have an ϵ-neighbor$\}$.
2. If some node in S' is involved in an ML constraint, then there is no solution to the FHC problem; otherwise, there is a solution.

When the above algorithm indicates that there is a feasible solution, one such solution is to create a singleton cluster for each node in S' and form one additional cluster containing all the nodes in $S - S'$. It is easy to see that the resulting partition of S satisfies the ML and ϵ constraints and that the feasibility testing algorithm runs in polynomial time. The following theorem summarizes the above discussion and indicates that we can extend the basic agglomerative algorithm with these combinations of constraint types to perform efficient hierarchical clustering. However, it does not mean that we can always use traditional agglomerative clustering algorithms as the closest-cluster-join operation can yield dead-end clustering solutions as discussed in Section 5.

Theorem 1. *The* FHC *problem can be solved efficiently for each of the following combinations of constraint types: (a) ML and CL (b) CL and ϵ and (c) ML and ϵ.* □

2.2 Feasibility Under ML, CL and ϵ Constraints

In this section, we show that the FHC problem is **NP**-complete when all the three constraint types are involved. This indicates that creating a dendrogram under these constraints is an intractable problem and the best we can hope for is an approximation algorithm that may **not** satisfy all constraints. The **NP**-completeness proof uses a reduction from the One-in-Three 3SAT with positive literals problem (OPL) which is known to be **NP**-complete [11]. For each instance of the OPL problem we can construct a constrained clustering problem involving ML, CL and ϵ constraints. Since complexity results are worse case, the existence of just these problems is sufficient for theorem 2.

One-in-Three 3SAT with Positive Literals (OPL)

Instance: A set $C = \{x_1, x_2, \ldots, x_n\}$ of n Boolean variables, a collection $Y = \{Y_1, Y_2, \ldots, Y_m\}$ of m clauses, where each clause $Y_j = (x_{j_1}, x_{j_2}, x_{j_3}\}$ has exactly three non-negated literals.

Question: Is there an assignment of truth values to the variables in C so that exactly one literal in each clause becomes true?

Theorem 2. *The* FHC *problem is **NP**-complete when the constraint set contains ML, CL and ϵ constraints.*

The proof of the above theorem is somewhat lengthy and is omitted because of space reasons. (The proof appears in an expanded technical report version of this paper [5] that is available on-line.)

3 Using Constraints for Hierarchical Clustering: Algorithm and Empirical Results

To use constraints with hierarchical clustering we change the algorithm in Figure 1 to factor in the above discussion. As an example, a constrained hierarchical clustering algorithm with must-link and cannot-link constraints is shown in Figure 2. In this section we illustrate that constraints can improve the quality of the dendrogram. We purposefully chose a small number of constraints and believe that even more constraints will improve upon these results. We will begin by investigating must-link and cannot-link constraints using six real world UCI datasets. For each data set we clustered all instances but removed the labels from 90% of the data (S_u) and used the remaining 10% (S_l) to generate constraints. We randomly selected two instances at a time from S_l and generated must-link constraints between instances with the same class label and cannot-link constraints between instances of differing class labels. We repeated this process twenty times, each time generating 250 constraints of each type. The performance measures reported are averaged over these twenty trials. All instances with missing values were

Table 2. Average Distortion per Instance and Average Percentage Cluster Purity over Entire Dendrogram

Data Set	Distortion		Purity	
	Unconstrained	Constrained	Unconstrained	Constrained
Iris	3.2	2.7	58%	66%
Breast	8.0	7.3	53%	59%
Digit (3 vs 8)	17.1	15.2	35%	45%
Pima	9.8	8.1	61%	68%
Census	26.3	22.3	56%	61%
Sick	17.0	15.6	50%	59%

ConstrainedAgglomerative(S,ML,CL) **returns** $Dendrogram_i$, *i* = k_{min} ... k_{max}

Notes: In Step 5 below, the term "mergeable clusters" is used to denote a pair of clusters whose merger does not violate any of the given CL constraints. The value of t at the end of the loop in Step 5 gives the value of $k_{\min}$.

1. Construct the transitive closure of the ML constraints (see [4] for an algorithm) resulting in r connected components $M_1, M_2, \ldots, M_r$.
2. If two points $\{x, y\}$ are both a CL and ML constraint then output "No Solution" and stop.
3. Let $S_1 = S - (\bigcup_{i=1}^{r} M_i)$. Let $k_{\max} = r + |S_1|$.
4. Construct an initial feasible clustering with $k_{\max}$ clusters consisting of the r clusters M_1, $\ldots$, M_r and a singleton cluster for each point in S_1. Set $t = k_{\max}$.
5. **while** (there exists a pair of mergeable clusters) **do**
   (a) Select a pair of clusters C_l and C_m according to the specified distance criterion.
   (b) Merge C_l into C_m and remove C_l. (The result is *Dendrogram*$_{t-1}$.)
   (c) $t = t - 1$.
   endwhile

Fig. 2. Agglomerative Clustering with ML and CL Constraints

removed as hierarchical clustering algorithms do not easily handle such instances. Furthermore, all non-continuous columns were removed as there is no standard distance measure for discrete columns.

Table 2 illustrates the quality improvement that the must-link and cannot-link constraints provide. Note that we compare the dendrograms for k values between k_{min} and k_{max}. For each corresponding level in the unconstrained and constrained dendrogram we measure the average distortion ($1/n * \sum_{i=1}^{n} D(x_i - C_{f(x_i)})$, where $f(x_i)$ returns the index of the closest cluster to x_i) and present the average over all levels. It is important to note that we are not claiming that agglomerative clustering has distortion as an objective function, rather that it is a good measure of cluster quality. We see that the distortion improvement is typically of the order of 15%. We also see that the average percentage purity of the clustering solution as measured by the class label purity improves. The cluster purity is measured against the extrinsic class labels. We believe these improvement are due to the following. When many pairs of clusters have simi-

lar short distances, the must-link constraints guide the algorithm to a better join. This type of improvement occurs at the bottom of the dendrogram. Conversely, towards the top of the dendrogram the cannot-link constraints rule out ill-advised joins. However, this preliminary explanation requires further investigation which we intend to address in the future. In particular, a study of the most informative constraints for hierarchical clustering remains an open question, though promising preliminary work for the area of non-hierarchical clustering exists [2].

We next use the cluster-level δ constraint with an arbitrary value to illustrate the great computational savings that such constraints offer. Our earlier work [4] explored ϵ and δ constraints to provide background knowledge towards the "type" of clusters we wish to find. In that paper we explored their use with the Aibo robot to find objects in images that were more than 1 foot apart as the Aibo can only navigate between such objects. For these UCI data sets no such background knowledge exists and how to set these constraint values for non-spatial data remains an active research area. Hence we test these constraints with arbitrary values. We set δ equal to 10 times the average distance between a pair of points. Such a constraint will generate hundreds even thousands of must-link constraints that can greatly influence the clustering results and algorithm efficiency as shown in Table 3. We see that the minimum improvement was 50% (for Census) and nearly 80% for Pima. This improvement is due to the constraints effectively creating a pruned dendrogram by making $k_{max} \ll n$.

Table 3. The Rounded Mean Number of Pair-wise Distance Calculations for an Unconstrained and Constrained Clustering using the δ constraint

Data Set	Unconstrained	Constrained
Iris	22,201	3,275
Breast	487,204	59,726
Digit (3 vs 8)	3,996,001	990,118
Pima	588,289	61,381
Census	2,347,305,601	563,034,601
Sick	793,881	159,801

4 Using the γ Constraint to Improve Performance

In this section we introduce a new constraint, the γ constraint and illustrate how the triangle inequality can be used to further improve the run-time performance of agglomerative hierarchical clustering. Though this improvement does not affect the worst-case analysis, we can perform a best case analysis and an expected performance improvement using the Markov inequality. Future work will investigate if tighter bounds can be found. There exists other work involving the triangle inequality but not constraints for non-hierarchical clustering [6] as well as for hierarchical clustering [10].

Definition 2. *(The γ Constraint For Hierarchical Clustering) Two clusters whose geometric centroids are separated by a distance greater than γ cannot be joined.*

IntelligentDistance (γ, $C = \{C_1, \ldots, C_k\}$)
returns $d(i,j)\ \forall i,j$.

1. **for** $i = 2$ **to** $n-1$ $\quad d_{1,i} = D(C_1, C_i)$ endloop
2. **for** $i = 2$ to $n-1$
   for $j = i+1$ to $n-1$ $\quad \hat{d_{i,j}} = |d_{1,i} - d_{1,j}|$
   if $\hat{d_{i,j}} > \gamma$ then $d_{i,j} = \gamma + 1$ *; do not join* **else** $d_{i,j} = D(x_i, x_j)$
   endloop
   endloop
3. return $d_{i,j}, \forall i,j$.

Fig. 3. Function for Calculating Distances Using the γ Constraint and the Triangle Inequality

The γ constraint allows us to specify how geometrically well separated the clusters should be. Recall that the triangle inequality for three points a, b, c refers to the expression $|D(a,b) - D(b,c)| \leq D(a,c) \leq D(a,b) + D(c,b)$ where D is the Euclidean distance function or any other metric function. We can improve the efficiency of the hierarchical clustering algorithm by making use of the lower bound in the triangle inequality and the γ constraint. Let a, b, c now be cluster centroids and we wish to determine the closest two centroids to join. If we have already computed $D(a,b)$ and $D(b,c)$ and the value $|D(a,b) - D(b,c)|$ *exceeds* γ, then we need not compute the distance between a and c as the lower bound on $D(a,c)$ already exceeds γ and hence a and c cannot be joined. Formally the function to calculate distances using geometric reasoning at a particular dendrogram level is shown in Figure 3. Central to the approach is that the distance between a central point (c) (in this case the first) and every other point is calculated. Therefore, when bounding the distance between two instances (a, b) we effectively calculate a triangle with two edges with know lengths incident on c and thereby lower bound the distance between a and b. How to select the best central point and the use of multiple central points remains future important research.

If the triangle inequality bound exceeds γ, then we save making m floating point power calculations if the data points are in m dimensional space. As mentioned earlier we have no reason to believe that there will be at least one situation where the triangle inequality saves computation in *all problem instances*; hence in the worst case, there is no performance improvement. But in practice it is expected to occur and hence we can explore the best and expected case results.

4.1 Best Case Analysis for Using the γ Constraint

Consider the n points to cluster $\{x_1, ..., x_n\}$. The first iteration of the agglomerative hierarchical clustering algorithm using symmetrical distances is to compute the distance between each point and every other point. This involves the computation ($D(x_1, x_2)$, $D(x_1, x_3)$,..., $D(x_1, x_n)$),..., ($D(x_i, x_{i+1})$, $D(x_i, x_{i+2})$,...,$D(x_i, x_n)$),..., ($D(x_{n-1}, x_n)$), which corresponds to an arithmetic series $n-1+n-2+\ldots+1$ of computations. Thus for agglomerative hierarchical clustering using *symmetrical* distances the number of distance computations is $n(n-1)/2$ for the base level. At the next

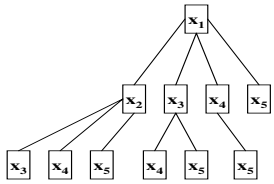


Fig. 4. A Simple Illustration for a Five Instance Problem of How the Triangular Inequality Can Save Distance Computations

level we need only recalculate the distance between the newly created cluster and the remaining $n-2$ points and so on. Therefore, the total number of distance calcluation is $n(n-1)/2+(n-1)(n-2)/2 = (n-1)^2$. We can view the base level calculation pictorially as a tree construction as shown in Figure 4. If we perform the distance calculation at the first level of the tree then we can obtain bounds using the triangle inequality for **all** branches in the second level. This is as bounding the distance between two points requires the distance between these points and a common point, which in our case is x_1. Thus in the best case there are only $n-1$ distance computations instead of $(n-1)^2$.

4.2 Average Case Analysis for Using the γ Constraint

However, it is highly unlikely that the best case situation will ever occur. We now focus on the average case analysis using the Markov inequality to determine the *expected* performance improvement which we later empirically verify. Let ρ be the average distance between any two instances in the data set to cluster. The triangle inequality provides a lower bound; if this bound exceeds γ, computational savings will result. We can bound how often this occurs if we can express γ in terms of ρ, hence let $\gamma = c\rho$.

Recall that the general form of the Markov inequality is: $P(X = x \geq a) \leq \frac{E(X)}{a}$, where x is a single value of the continuous random variable X, a is a constant and $E(X)$ is the expected value of X. In our situation since X is distance between two points chosen at random, $E(X) = \rho$ and $\gamma = a = c\rho$ as we wish to determine when the distance will exceed γ. Therefore, at the lowest level of the tree ($k = n$) then the *number of times* the triangle inequality will save us computation time is $n\frac{E(X)}{a} = n\frac{\rho}{c\rho} = n/c$, indicating a saving of a factor of $1/c$ at this lowest level. As the Markov inequality is a rather weak bound then in practice the saving may be substantially different as we shall see in our empirical section. The computation saving that are obtained at the bottom of the dendrogram are reflected at higher levels of the dendrogram. When growing the entire dendrogram we will save at least $n/c+(n-1)/c \ldots +1/c$ distance calculations. This is an arithmetic sequence with the additive constant being $1/c$ and hence the total expected computations saved is at least $n/2(2/c+(n-1)/c) = (n^2+n)/2c$. As the total computations for regular hierarchical clustering is $(n-1)^2$, the computational saving is expected to be by a approximately a factor of $1/2c$.

Consider the 150 instance IRIS data set (n=150) where the average distance (with attribute value ranges all being normalized to between 0 and 1) between two instances is 0.6; that is, $\rho = 0.6$. If we state that we do not wish to join clusters whose centroids are

Table 4. The Efficiency of Using the Geometric Reasoning Approach from Section 4 (Rounded Mean Number of Pair-wise Distance Calculations)

Data Set	Unconstrained	Using γ Constraint
Iris	22,201	19,830
Breast	487,204	431,321
Digit (3 vs 8)	3,996,001	3,432,021
Pima	588,289	501,323
Census	2,347,305,601	1,992,232,981
Sick	793,881	703,764

separated by a distance greater than 3.0, then $\gamma = 3.0 = 5\rho$. By not using the γ constraint and the triangle inequality the total number of computations is 22201, and the number of computations that are saved is at least $(150^2 + 150)/10 = 2265$; hence the saving is about 10%. We now show that the γ constraint can be used to improve efficiency of the basic agglomerative clustering algorithm. Table 4 illustrates the improvement that using a γ constraint equal to five times the average pairwise instance distance. We see that the average improvement is consistent with the average case bound derived above.

5 Constraints and Irreducible Clusterings

In the presence of constraints, the set partitions at each level of the dendrogram must be feasible. We have formally shown that if k_{max} is the maximum value of k for which a feasible clustering exists, then there is a way of joining clusters to reach another clustering with k_{min} clusters [5]. In this section we ask the following question: will traditional agglomerative clustering find a feasible clustering for each value of k between k_{max} and k_{min}? We formally show that in the worse case, for certain types of constraints (and combinations of constraints), if mergers are performed in an arbitrary fashion (including the traditional hierarchical clustering algorithm, see Figure 1), then the dendrogram may prematurely dead-end. A premature dead-end implies that the dendrogram reaches a stage where no pair of clusters can be merged without violating one or more constraints, even though other sequences of mergers may reach significantly higher levels of the dendrogram. We use the following definition to capture the informal notion of a "premature end" in the construction of a dendrogram. How to perform agglomerative clustering in these dead-end situations remains an important open question.

Definition 3. *A feasible clustering $C = \{C_1, C_2, \ldots, C_k\}$ of a set S is* ***irreducible*** *if no pair of clusters in C can be merged to obtain a feasible clustering with $k - 1$ clusters.*

The remainder of this section examines the question of which combinations of constraints can lead to premature stoppage of the dendrogram. We first consider each of the ML, CL and ϵ-constraints separately. It is easy to see that when only ML-constraints are used, the dendrogram can reach all the way up to a single cluster, no matter how mergers are done. The following illustrative example shows that with CL-constraints, if mergers are not done correctly, the dendrogram may stop prematurely.

Example: Consider a set S with $4k$ nodes. To describe the CL constraints, we will think of S as the union of four pairwise disjoint sets X, Y, Z and W, each with k nodes. Let $X = \{x_1, x_2, \ldots, x_k\}$, $Y = \{y_1, y_2, \ldots, y_k\}$, $Z = \{z_1, z_2, \ldots, z_k\}$ and $W = \{w_1, w_2, \ldots, w_k\}$. The CL-constraints are as follows. *(a)* There is a CL-constraint for each pair of nodes $\{x_i, x_j\}$, $i \neq j$, *(b)* There is a CL-constraint for each pair of nodes $\{w_i, w_j\}$, $i \neq j$, *(c)* There is a CL-constraint for each pair of nodes $\{y_i, z_j\}$, $1 \leq i, j \leq k$.

Assume that the distance between each pair of nodes in S is 1. Thus, nearest-neighbor mergers may lead to the following feasible clustering with $2k$ clusters: $\{x_1, y_1\}$, $\{x_2, y_2\}$, $\ldots$, $\{x_k, y_k\}$, $\{z_1, w_1\}$, $\{z_2, w_2\}$, $\ldots$, $\{z_k, w_k\}$. This collection of clusters can be seen to be irreducible in view of the given CL constraints. However, a feasible clustering with k clusters is possible: $\{x_1, w_1, y_1, y_2, \ldots, y_k\}$, $\{x_2, w_2, z_1, z_2, \ldots, z_k\}$, $\{x_3, w_3\}$, $\ldots$, $\{x_k, w_k\}$. Thus, in this example, a carefully constructed dendrogram allows k additional levels. □

When only the ϵ-constraint is considered, the following lemma points out that there is only one irreducible configuration; thus, no premature stoppages are possible. In proving this lemma, we will assume that the distance function is symmetric.

Lemma 1. *If S is a set of nodes to be clustered under an ϵ-constraint. Any irreducible and feasible collection C of clusters for S must satisfy the following two conditions.*

(a) C contains at most one cluster with two or more nodes of S.
(b) Each singleton cluster in C contains a node x with no ϵ-neighbors in S.

Proof: Suppose C has two or more clusters, say C_1 and C_2, such that each of C_1 and C_2 has two or more nodes. We claim that C_1 and C_2 can be merged without violating the ϵ-constraint. This is because each node in C_1 (C_2) has an ϵ-neighbor in C_1 (C_2) since C is feasible and distances are symmetric. Thus, merging C_1 and C_2 cannot violate the ϵ-constraint. This contradicts the assumption that C is irreducible and the result of Part (a) follows. The proof for Part (b) is similar. Suppose C has a singleton cluster $C_1 = \{x\}$ and the node x has an ϵ-neighbor in some cluster C_2. Again, C_1 and C_2 can be merged without violating the ϵ-constraint. □

Lemma 1 can be seen to hold even for the combination of ML and ϵ constraints since ML constraints cannot be violated by merging clusters. Thus, no matter how clusters are merged at the intermediate levels, the highest level of the dendrogram will always correspond to the configuration described in the above lemma when ML and ϵ constraints are used. In the presence of CL-constraints, it was pointed out through an example that the dendrogram may stop prematurely if mergers are not carried out carefully. It is easy to extend the example to show that this behavior occurs even when CL-constraints are combined with ML-constraints or an ϵ-constraint.

6 Conclusion and Future Work

Our paper made two significant theoretical results. Firstly, the feasibility problem for *unspecified* k is studied and we find that clustering under all four types (ML, CL, ϵ and δ) of constraints is **NP**-complete; hence, creating a feasible dendrogram is intractable. These results are fundamentally different from our earlier work [4] because the feasibility problem and proofs are quite different. Secondly, we proved under some constraint

types (i.e. cannot-link) that traditional agglomerative clustering algorithms give rise to dead-end (irreducible) solutions. If there exists a feasible solution with k_{max} clusters then the traditional agglomerative clustering algorithm may not get all the way to a feasible solution with k_{min} clusters even though there exists feasible clusterings for each value between k_{max} and k_{min}. Therefore, the approach of joining the two "nearest" clusters may yield an incomplete dendrogram. How to perform clustering when dead-end feasible solutions exist remains an important open problem we intend to study.

Our experimental results indicate that small amounts of labeled data can improve the dendrogram quality with respect to cluster purity and "tightness" (as measured by the distortion). We find that the cluster-level δ constraint can reduce computational time between two and four fold by effectively creating a pruned dendrogram. To further improve the efficiency of agglomerative clustering we introduced the γ constraint, that allows the use of the triangle inequality to save computation time. We derived best case and expected case analysis for this situation which our experiments verified. Additional future work we will explore include constraints to create balanced dendrograms and the important asymmetric distance situation.

References

1. S. Basu, A. Banerjee, R. Mooney, Semi-supervised Clustering by Seeding, 19th *ICML*, 2002.
2. S. Basu, M. Bilenko and R. J. Mooney, Active Semi-Supervision for Pairwise Constrained Clustering, 4th *SIAM Data Mining Conf.*. 2004.
3. P. Bradley, U. Fayyad, and C. Reina, "Scaling Clustering Algorithms to Large Databases", 4th *ACM KDD Conference.* 1998.
4. I. Davidson and S. S. Ravi, "Clustering with Constraints: Feasibility Issues and the k-Means Algorithm", *SIAM International Conference on Data Mining*, 2005.
5. I. Davidson and S. S. Ravi, "Towards Efficient and Improved Hierarchical Clustering with Instance and Cluster-Level Constraints", Tech. Report, CS Department, SUNY - Albany, 2005. Available from: www.cs.albany.edu/∼davidson
6. C. Elkan, Using the triangle inequality to accelerate k-means, *ICML*, 2003.
7. M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman and Co., 1979.
8. M. Garey, D. Johnson and H. Witsenhausen, "The complexity of the generalized Lloyd-Max problem", *IEEE Trans. Information Theory*, Vol. 28,2, 1982.
9. D. Klein, S. D. Kamvar and C. D. Manning, "From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering", *ICML* 2002.
10. M. Nanni, Speeding-up hierarchical agglomerative clustering in presence of expensive metrics, *PAKDD 2005*, LNAI 3518.
11. T. J. Schafer, "The Complexity of Satisfiability Problems", *STOC*, 1978.
12. K. Wagstaff and C. Cardie, "Clustering with Instance-Level Constraints", *ICML*, 2000.
13. D. B. West, *Introduction to Graph Theory*, Second Edition, Prentice-Hall, 2001.
14. K. Yang, R. Yang, M. Kafatos, "A Feasible Method to Find Areas with Constraints Using Hierarchical Depth-First Clustering", *Scientific and Stat. Database Management Conf.*, 2001.
15. O. R. Zaiane, A. Foss, C. Lee, W. Wang, On Data Clustering Analysis: Scalability, Constraints and Validation, *PAKDD*, 2000.
16. Y. Zho & G. Karypis, "Hierarchical Clustering Algorithms for Document Datasets", *Data Mining and Knowledge Discovery*, Vol. 10 No. 2, March 2005, pp. 141–168.

Cluster Aggregate Inequality and Multi-level Hierarchical Clustering

Chris Ding and Xiaofeng He

Lawrence Berkeley National Laboratory,
Berkeley, California 94720, USA

Abstract. We show that (1) in hierarchical clustering, many linkage functions satisfy a cluster aggregate inequality, which allows an exact $O(N^2)$ multi-level (using mutual nearest neighbor) implementation of the standard $O(N^3)$ agglomerative hierarchical clustering algorithm. (2) a desirable close friends cohesion of clusters can be translated into kNN consistency which is guaranteed by the multi-level algorithm; (3) For similarity-based linkage functions, the multi-level algorithm is naturally implemented as graph contraction. The effectiveness of our algorithms is demonstrated on a number of real life applications.

1 Introduction

Agglomerative hierarchical clustering (AHC) is developed in 1960's and is widely used in practice. AHC produces a tree describing the hierarchical cluster structure. Such a comprehensive description of the data is quite useful for broad areas of applications. For example, in bioinformatics research, AHC is most commonly used for clustering genes in a DNA gene microarray expression data, because the resulting hierarchical cluster structure is readily recognizable by biologists. The phylogenetic tree (similar to binary clustering tree) of organisms is often built using the UPGMA (unweighted pair group method average) AHC algorithm. In social sciences, the hierarchical cluster structure often reveals gradual evolving social relationships that help explain complex social issues. Another application of AHC is in classification tasks on a large dataset using support vector machine [12]. The hierarchical cluster structure allows one to use most detailed local representation near the decision boundaries where support vectors lie; but as one moves away from the decision boundaries, the centroid representation of progressively larger clusters can be used.

Besides hierarchical clustering, many other clustering algorithms have been developed (see recent survey and text books [5,1,3]). K-means clustering is perhaps the most commonly used method and is well developed. The gaussian mixture model using EM algorithm directly improves over the K-means method by using a probabilistic model of cluster membership of each object. Both algorithms can be viewed as a global objective function optimization problem. A related set of graph clustering algorithms are developed that partition nodes into two sub-clusters based on well-motivated clustering objective functions [8]. They

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 71–83, 2005.

are typically applied in a top-down fashion (see also[7]), and thus complement the bottom-up AHC.

Standard AHC scales as $O(N^3)$. A detailed description of AHC and complete references can be found in [4,9]. A number of efficient implementation based on approximations have been proposed [10,6]. Several recent studies propose to integrate hierarchical clustering with additional information [2,11] or summary statistics[13].

In this paper, we focus on making AHC scale to large data set. Over the last 40 years, the basic AHC algorithm remains unchanged. The the basic algorithm is an iterative procedure; at each iteration, among all pairs of current clusters, the pair with largest linkage function value (smallest distance) are selected and merged.

We start with a key observation. In the AHC algorithm, at each iteration, one may merge all mutual-nearest-neighbor (1mn) pairs (defined by the linkage function) simultaneously in the same iteration. As long as the linkage function satisfies a "cluster aggregate inequality", this modified algorithm of simultaneously merge all 1mn-pairs at each iteration produces identical clustering results as the standard AHC algorithm. The cluster aggregate inequality is satisfied by most common linkage functions (see §2.2). This modified algorithm provides a natural multi-level implementation of the AHC, where at each level we merge all 1mn pairs. This [illegible] (MLHC) algorithm provides an order-N speedup over the standard AHC.

Next we propose "close friends" cohesion as a desirable feature for clustering, which requires that for every member in a cluster, its closest friend is also in the same cluster. We show that the MLHC guarantees the close-friends cohesion, a desirable feature for clustering. We further extend this cluster membership cohesion idea to (mutual) nearest neighbor consistency, and show that MLHC improves this KNN consistency compare to other clustering algorithms. (§3)

When the linkage function is expressed in similarity (in contrast to distance or dissimilarity), the new algorithm is identical to multi-level [illegible]. Graph contraction provides a natural framework for hierarchical clustering (§4).

The effectiveness of our algorithms is demonstrated on a number of real life applications: DNA gene expressions for lung cancer, global climate pattern, and internet newsgroups (§5).

2 Multi-level Hierarchical Clustering (MLHC)

2.1 Algorithm

The standard agglomerative hierarchical clustering is a bottom-up process. During each step, we merge two [illegible] clusters C_p and C_q which are closest, or [illegible] among all pairs of current clusters:

$$\min_{<pq>} d(C_p, C_q).$$

where $d(\cdot,\cdot)$ is the dissimilarity-based (distance) linkage function between C_p and C_q. Many researches have studied the effects of different choice of [illegible] functions [4].

At each step of AHC with p current clusters, $p-1$ new linkage functions need be computed, and we have total $O(p^2)$ pairwise linkage functions. It takes p^2 comparisons to search for the pair with the largest linkage. This is repeated $N-1$ times. The total computation is

$$N_{\text{search}}^{\text{AHC}} = N^2 + (N-1)^2 + \cdots + 2^2 + 1^2 = O(N^3/3).$$

The new MLHC algorithm is motivated by the following observation. In each iterative step in AHC, when all pairwise linkage functions are computed, we can form all mutual nearest neighbor pairs (1mn-pairs) of current clusters using the linkage as the distance metric. Two objects (i,j) are a 1mn-pair if j is the nearest neighbor of i and vice versa.

In this perspective, the standard AHC merges only the 1mn-pair with largest linkage value. It is then natural to ask if we may also merge all other 1mn-pairs simultaneously. Will this produces the same results? An analysis shows that if the linkage function satisfies a "cluster aggregate inequality", then the clustering results remain the same as the standard AHC.

This observation suggests a simultaneous merging algorithm which we call MLHC. At each level, all 1mn-pairs are identified and merged (not just the pair with largest linkage value). This is repeated until all objects are merged into one cluster. The total number of level is about $log_2 N$. Thus the required computation is approximately

$$N_{\text{search}}^{\text{MLHC}} = N^2 + (N/2)^2 + (N/4)^2 + 2^2 + 1^2 = O(4N^2/3).$$

The new algorithm speedups by a factor of order-N.

2.2 Cluster Aggregate Inequality

In this section, we show that the simultaneous merging of all 1mn-pairs in MLHC produces identical clustering results as the standard AHC, provided the linkage function satisfies a [illegible]

Definition. [illegible] of the linkage function. Suppose we have three current clusters A, B, C. We try to merge B, C into a new cluster $B+C$. The cluster aggregate inequality is a property of the linkage function that the merged cluster $B+C$ is "no closer" to A than either one of its individual members B or C. More precisely, for distance (dissimilarity) based linkage $d(\cdot,\cdot)$, the cluster aggregate inequality is

$$d_{A,B+C} \geq \min(d_{A,C}, d_{A,B}) \tag{1}$$

for any triple (A, B, C).

What kind of linkage functions satisfy the cluster aggregate inequality? It is interesting to see that most commonly used linkage functions satisfy cluster

aggregate inequality. Consider the four similarity-based linkage function. (i) the single linkage, defined as the closest distance among points in A, B, (ii) the complete linkage, defined as the fartherest distance among points in A, B, (iii) the average linkage, defined as the average of all distances among points in A, B, (iv) the minimum variance linkage.

$$d^{\text{sgl}}_{A,B} = \min_{i\in A\ j\in B} d_{ij} \tag{2}$$

$$d^{\text{cmp}}_{A,B} = \max_{i\in A\ j\in B} d_{ij} \tag{3}$$

$$d^{\text{avg}}_{A,B} = \frac{d(A,B)}{n_A\, n_B} \tag{4}$$

$$d^{\text{min-var}}_{A,B} = \frac{n_A\, n_B}{n_A + n_B} ||\mathbf{c}_A - \mathbf{c}_B||^2. \tag{5}$$

Theorem 1. The single link, the complete link and average link satisfy the strong cluster aggregate inequality.

Proof. For single link, one can easily see that

$$d^{\text{sgl}}_{A,B+C} = \min_{i\in A; j\in B+C} d_{ij} = \min(\min_{i\in A; j\in B} d_{ij}, \min_{i\in A; j\in C} d_{ij}) = \min(d^{\text{sgl}}_{A,B}, d^{\text{sgl}}_{A,C})$$

Thus the equality in Eq.(1) hold for single link. With same reasoning, one can see the inequality holds for complete linkage.

For average link, we have

$$\begin{aligned}
d^{\text{avg}}_{A,B+C} &= \sum_{i\in A; j\in B+C} \frac{d_{ij}}{|A||B+C|} \\
&= \sum_{i\in A; j\in B} \frac{d_{ij}}{|A||B+C|} + \sum_{i\in A; j\in C} \frac{d_{ij}}{|A||B+C|} \\
&= \frac{|B|}{|B+C|} d^{\text{avg}}_{A,B} + \frac{|C|}{|B+C|} d^{\text{avg}}_{A,C} \\
&\geq \frac{|B|}{|B+C|} \min(d^{\text{avg}}_{A,B}, d^{\text{avg}}_{A,C}) + \frac{|C|}{|B+C|} \min(d^{\text{avg}}_{A,B}, d^{\text{avg}}_{A,C}) \\
&= \min(d^{\text{avg}}_{A,B}, d^{\text{avg}}_{A,C})
\end{aligned}$$

□

2.3 Equivalence of MLHC and AHC

Cluster aggregate inequality plays a fundamental role in hierarchical clustering. It is similar to the triangle inequality in metric space: for any three vectors $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ in the Hilbert space with the distance metric $d(\cdot,\cdot)$, the metric must satisfies the triangle inequality

$$d_{ik} \leq d_{ij} + d_{jk}.$$

Triangle inequality plays a fundamental role in deriving properties of the metric Space. Now we prove the main results of this paper.

Theorem 2. If the linkage function satisfies the cluster aggregate inequality, the clustering trees produced by MLHC is identical to that produced by standard AHC.

Proof. We view the linkage function as a distance metric and build all 1mn-pairs based on the linkage function. We show that the AHC is iteratively merging 1mn-pairs, which is same as MLHC. The details is broken into two features of AHC below. □

We first note a simple feature of AHC:

Feature 1. The closest pair must be a 1mn-pair.

Proof. Suppose this is not true, i.e., the closest pair is (a, b), but a is not the closest neighbor of b. There must exist another point c which is the closest neighbor of a. Then the pair (a, c) must be the closest pair, but this contradicts the fact that (a, b) is the closest pair. Thus a must be the closest neighbor of b. Similarly, b must be the closest neighbor of a. □

Next, we prove a key feature of AHC. This shows the essence of Theorem 2.

Feature 2. If the linkage function satisfies the cluster aggregate inequality, then any 1mn-pair must be preserved and will merge eventually in standard AHC.

Proof. Suppose at certain iteration, the current clusters are listed as (C_{j_1}, C_{j_2}), $(C_{j_3}, C_{j_4}), C_{j_5}, (C_{j_6}, C_{j_7}), \cdots$, where 1mn-pair is indicated by the parenthesis. In AHC, the 1mn-pair with largest linkage value , say (C_{j_6}, C_{j_7}), is merged. Due to the cluster aggregate inequality, the newly merged cluster $C_{(j_6,j_7)}$ will be "no closer" to any other current clusters. Thus the 1mn of $C_{(j_6,j_7)}$ can not be any member of the current remaining 1mn-pairs, say C_{j_1}.

This can seen as follows. By construction, neither C_{j_6} nor C_{j_7} is closer to C_{j_1} than C_{j_2} does. Due to the cluster aggregate inequality, the newly merged cluster $C_{(j_6,j_7)}$ will be "no closer" to C_{j_1} than either C_{j_6} or C_{j_7} does. Thus 1mn of $C_{(j_6,j_7)}$ can not be C_{j_1}.

This guarantees that the 1mn-pair (C_{j_1}, C_{j_2}) will never be broken by a merged cluster. Thus in the next iteration, either a current 1mn-pair, say (C_{j_3}, C_{j_4}), is merged, or the newly-merged $C_{(j_6,j_7)}$ is merged with a singleton cluster, say C_{j_5}. Therefore, the 1mn-pair (C_{j_1}, C_{j_2}) will preserve and eventually merge at some later iteration. □

We give a simple example to illustrating some of the concepts. In Figure 1(b), we have 5 objects. They form two 1mn-pairs (a, b), (d, e) and one isolated object c. We do the standard AHC. Suppose (a, b) has the largest linkage value. So a,b are first merged into $(a+b)$. We assert that the 1mn-pair (d, e) must be preserved and will merge in later stages in AHC. This is done in two stages. First, we show that d cannot be the nearest neighbor of $(a + b)$, i.e.,

$$d(d, a + b) > d(d, e). \tag{6}$$

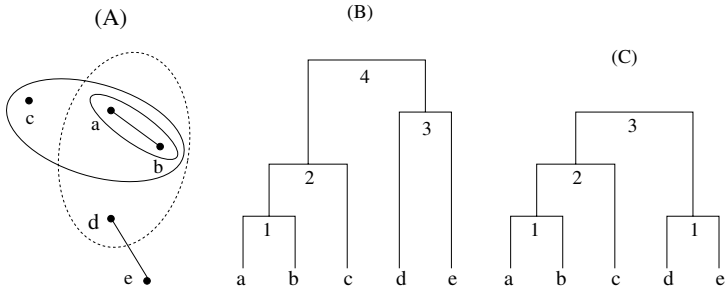


Fig. 1. (A) Dataset of 5 objects. 1mn-pairs are linked by a line. A merge is indicated by an elongated solid circle. (B) Dendrogram using AHC. Numbers indicate order of the merge. (C) Dendrogram using MLHC.

The fact that (d, e) is a 1mn-pair implies

$$d(d, a) > d(d, e), \tag{7}$$

$$d(d, b) > d(d, e), \tag{8}$$

$$d(d, c) > d(d, e). \tag{9}$$

From the cluster aggregate inequality, $d(d, a + b) \geq \min[d(d, a), d(d, b)]$. From this, and makeing use of Eqs.(7, 8), we obtain Eq.(6).

In the next round of AHC, either (i) the pair $(a+b, c)$ has the largest linkage value, or (ii) the pair (d, e) has the largest linkage value. If (ii) holds, our assertion is proved. Suppose (i) holds. Then $(a+b, c)$ are merged into $(a+b+c)$. Now we show that d cannot be the nearest neighbor of $(a + b + c)$, i.e.,

$$d(d, a + b + c) > d(d, e). \tag{10}$$

From the cluster aggregate inequality, $d(d, a + b + c) \geq \min[d(d, a + b), d(d, c)]$ From this, together with Eqs.(6, 9), we obtain Eq.(10). Therefore, (d, e) is the pair with the largest linkage value. Thus (d, e) are merged into $(d + e)$.

This example shows how 1mn pairs are preserved in AHC. Thus any cluster merge in MLHC (those 1mn-pairs) will also occur in AHC. There are total $N-1$ cluster merges in both MLHC and AHC. So any cluster merge in AHC also occur in MLHC.

Both AHC and MLHC algorithms lead to the same binary cluster tree. The difference is the sequential order they are merged. This is illustrated in Figure 1.(B,C). If we represent the tree hight by the standard linkage function value for each merge, the dendrograms of the two algorithm remains identical.

We emphasize that the equivalence of MLHC and AHC only requires 1mn-pair preservation during AHC, as shown in Feature 2 in the above. Therefore, cluster aggregate inequality is a [illegible] condition for 1mn-pair preservation.

For a given dataset, it is possible that a particular linkage function maybe not satisfy the generic cluster aggregate inequality for all possible triples (i, j, k),

but the 1mn-pair preservation holds during AHC and thus MLHC is equivalent to AHC.

In summary, these analysis not only shows that MLHC is equivalent to AHC, thus providing a $O(n)$ speedup; but also brought out a new insight for AHC, i.e., 1mn-pair preservation during AHC. This leads to close-friends cohesion.

3 "Close Friends" Cohesion

One of the fundamental concept of data clustering is that members of the same cluster have high association with each other. One way to characterize the within cluster association is the cohesion of the cluster members via the preservation of "close friends". Suppose we divide 100 people into 10 teams. The cohesion of each team is greatly enhanced if for any member in a team, his/her close friends are also in the same team.

It is interesting to note that this close-friends cohesion is guaranteed by MLHC, if we interprete 1mn-pair relationship as close friends; by construction, this cohesion is guaranteed at all levels. We say the clustering results satisfy 1mn-consistency if for each object in a cluster, its 1mn is also in the same cluster.

By Theorem 2, clusters produced by the standard AHC also enjoy the same 1mn-consistency as in MLHC. We summarize this important result as

Theorem 3. In MLHC, 1mn-consistency is fully guaranteed. In agglomerative hierarchical clustering, if the linkage function satisfy the cluster aggregate inequality, 1mn-consistency is fully guaranteed.

3.1 Cluster Membership kNN Consistency

1mn describe the "most close" friend. Beyond the closest friend, it is desirable that other less close friends are also in the same cluster. This will increase cohesiveness of the clustering.

We thus further extend the "close friends" into k-nearest-neighbor, the cohesiveness of a cluster becomes the following knn consistency:

Cluster Membership kNN Consistency: For any data object in a cluster, its k-nearest neighbors are also in the same cluster.

Note that the relationship of "close friend" is not symmetric. Although a's closest friend is b, b's closest friend could be c, not a. Thus the "mutual closest friend" implies the tightest friendship. Thus k-Mutual-Nearest-Neighbor Consistency is more desirable.

3.2 Enforcing kNN Consistency

In general, clustering algorithms perform global optimizations, such as K-means algorithm, will gives final clusters with a smaller degree of cluster knn and kmn consistency than the bottom hierarchical clustering. This is because that in global optimizations, nearest-neighbor relations are not considered explicitly.

In HC, a pair of clusters are merged if their linkage value is high. Thus clusters being merged are typically very similar to each other. Therefore the nearest-neighbor local information are utilized to some extent, leading to higher degree of knn consistency.

What about other knn/kmn consistency? First we note that cluster knn consistency defines a "transitive relationship". If x_1 is a member of cluster C_1, and x_2 is the 1nn of x_1, then by cluster 1nn consistency, x_2 should also be a member of C_1. This transitive relation implies that 100% 1nn consistency can be achieved only if entire connected component of the 1nn graph are in C_1. To generate clusters that guarantee 100% knn consistency, at each level of the MLHC, we must first generate knn-graph, identify all connected components, and for each CC, merge all current clusters into one cluster.

Because for any object, its 2nn set always include its 1nn set. Thus 2nn consistency guarantees 1nn consistency. Similarly, because any knn set include kmn set, knn consistency implies (k-1)-nn consistency.

4 Similarity-Based Hierarchical Clustering: Multi-level Graph Contraction

The above discussion uses the distance-based linkage. All results there can be easily translate into similarity-based linkage function.

For similarity-based linkage we select the pair with the largest linkage to merge: $\max_{<pq>} s(C_p, C_q)$, where $s(C_p, C_q)$ is the aggregate similarity between clusters C_p, C_q. Let the initial pairwise similarity are $W = (w_{ij})$. The aggregate similarity has a simple form, $s(C_p, C_q) = \sum_{i \in C} \sum_{j \in C} w_{ij}$.

Cluster aggregate inequality using similarity-based linkage can be written as

$$s(A, B + C) \leq \max[s(A, B), s(A, C)] \tag{11}$$

Consider the following similarity-based linkage functions. (i) the single linkage, defined as the largest similarity among points in A, B, (ii) the complete linkage, defined as the smallest similarity among points in A, B, (iii) the average linkage, defined as the average of all similarities among points in A, B,

$$s_{\text{single}}(A, B) = \max_{i \in A \ j \in B} s_{ij}, \ s_{\text{complete}}(A, B) = \min_{i \in A \ j \in B} s_{ij}, \ s_{\text{avg}}(A, B) = \frac{s(A, B)}{|A||B|}.$$

With similar analysis as in the case of distance-based clustering, we can proof

Theorem 4. All three above similarity-based linkage functions satisfy the cluster aggregate inequality. The similarity based linkage functions have an advantage that merging two cluster become graph node contraction. Defining the similarity between two objects as the weight on an edge between them, this forms a similarity graph. Thus the multi-level hierarchical clustering naturally become multi-level graph contraction of the similarity graph. Many well-known results in graph theory can be applied.

Merging two current clusters into a new cluster corresponds to contracting two nodes i, j into a new node k and with edge e_{ij} being eliminated. Weights of the graph are updated according to standard graph contraction procedure. Let $W^{(t)}, W^{(t+1)}$ be the weights of the similarity graph at steps $t, t+1$. The updated weights for contracting the edge e_{ij} and merging nodes i, j into k are

$$\begin{cases} w_{kk}^{(t+1)} = w_{ii}^{(t)} + w_{jj}^{(t)} + w_{ij}^{(t)} \\ w_{kp}^{(t+1)} = w_{ip}^{(t)} + w_{jp}^{(t)}, \quad \forall p \notin \{i, j, k\} \\ w_{pq}^{(t+1)} = w_{pq}^{(t)}, \qquad \forall p, q \notin \{i, j, k\} \end{cases}$$

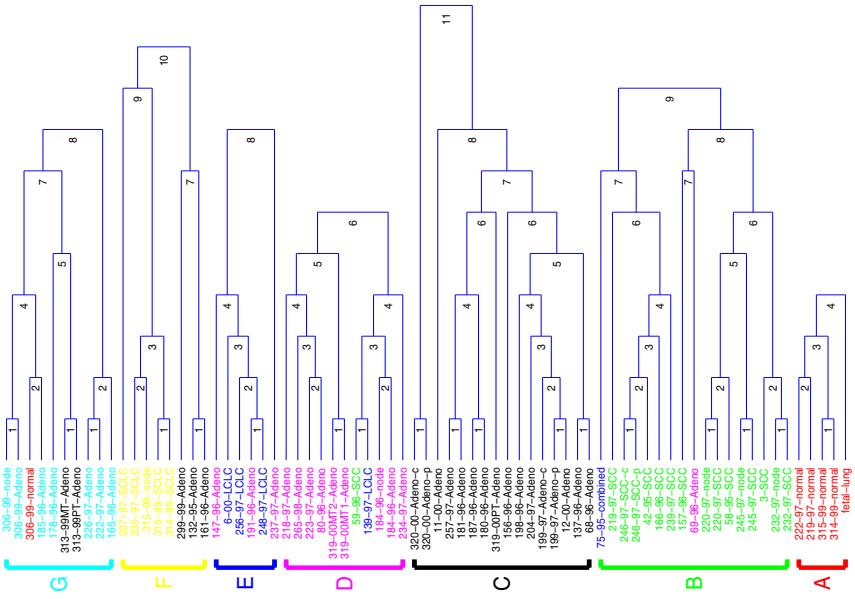


Fig. 2. MLHC clustering of lung cancer gene expressions

Using the above graph contraction operation, AHC and MLHC can be described very succinctly. AHC can be viewed as contracting one edge (with heaviest weight) at each step. The steps are repeated until all nodes merge into one. This takes exactly $N-1$ steps. MLHC can be viewed as contracting all 1mn-pairs simultaneously at each step. It is repeated about $O(\log_2 N)$ times. Now the speedup of MLHC over AHC is clear. At each step, all-pair linkage function computation is necessary. But the number of steps required in AHC is $N-1$, and it is $O(\log_2 N)$ in MLHC.

5 Experiments

5.1 DNA Gene Expression for Lung Cancer

The DNA gene expressions of lung cancer patients (available online: http://genome-www.stanford.edu/lung_cancer/adeno/) contains 73 samples of 67 lung tumors from patients whose clinical course was followed for up to 5 years. The samples comprise of 916 DNA clones representing 835 unique genes. The samples are classified into 5 groups by visual examination (41 Adenocarcinomas (ACs), 16 squamous cell carcinomas (SCCs), 5 large cell lung cancers(LCLCs), 5 small cell lung cancer (SCLCs) and 5 normal tissue with one fetal lung tissue.). The largest group ACs is further classified into three smaller groups. The purpose is to see if we can recover this 7 groups using unsupervised learning method, i.e., the hierarchical clustering method. The Pearson correlations c_{ij} among tissue samples are computed first, and the similarity metric is defined as $s_{ij} = \exp(5c_{ij})$. We use MLHC and obtain the cluster structure as shown in Fig.2.

As the Figure shows, at 1st level, after an all-pair computation, 18 1mn-pairs are formed and merged. At 2nd level, 11 1mn-pairs are formed and merged. Total 11 levels of merges are required to obtain 7 clusters. In contrast, for standard AHC, We need 66 levels of merge steps to obtain 7 clusters. The clustering result is give in the confusion matrix $= \begin{bmatrix} 5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 15 & \cdot & 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & 16 & \cdot & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & 9 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 3 & 3 & \cdot & \cdot \\ \cdot & \cdot & 3 & \cdot & \cdot & 5 & \cdot \\ 1 & \cdot & 2 & \cdot & \cdot & \cdot & 7 \end{bmatrix}$ where $T = (t_{ij})$, t_{ij} is the number of data points which are observed to be in cluster i, but was computed via the clustering method to belong to cluster j. The accuracy is defined as $Q = \sum_k t_{kk}/N = 82\%$. indicating the effectiveness of the clustering algorithm.

5.2 Climate Pattern

We tested MLHC on global precipitation data as shown in Fig.3. Regularly-spaced data points cover the surface of earth. Each data point is a 402-dimensional vector containing seasonal means over 100 years and geometric information: longitude and latitude. Similarity between two points are based two factors: (1) precipitation pattern similarity computed as Euclidean distance and (2) geometric closeness based on simple physical distance. The obtained stable regions (shown in different color and symbols) correlate well with continents, say, in Australia, south Americas.

5.3 Internet Newsgroups

We apply MLHC on Internet newsgroup articles. A 20-newsgroup dataset is from www.cs.cmu.edu /afs/cs/project/theo-11/www/naive-bayes.html. 1000 words are selected according to the mutual information between words and documents in unsupervised manner. Word - document matrix is first constructed using standard `tf.idf` term weighting. Cosine similarity between documents is used. We focus on two sets of 5-newsgroup combinations listed below:

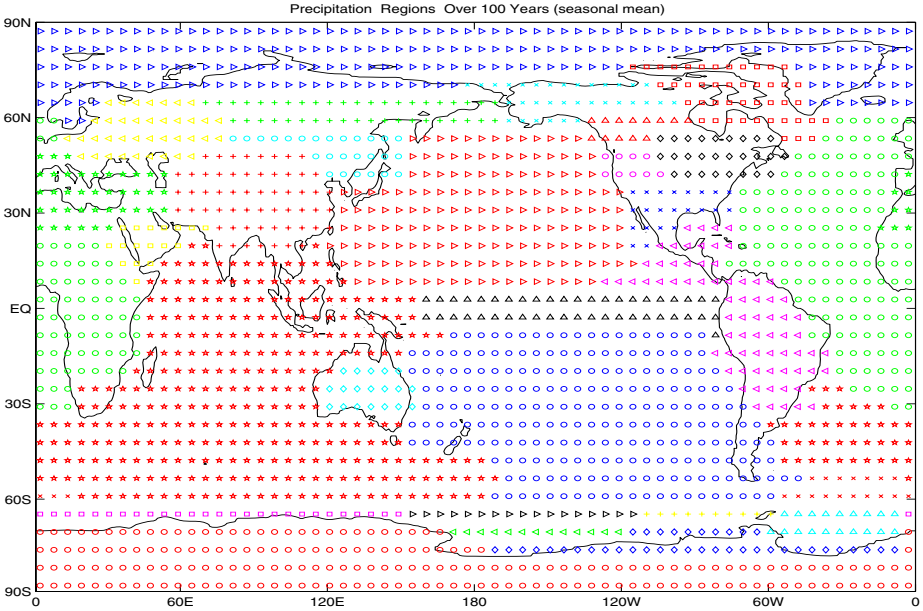


Fig. 3. Global precipitation pattern based on seasonal means over 100 years

A5:		B5:	
NG2:	comp.graphics	NG2:	comp.graphics
NG9:	rec.motorcycles	NG3:	comp.os.ms-windows
NG10:	rec.sport.baseball	NG8:	rec.autos
NG15:	sci.space	NG13:	sci.electronics
NG18:	talk.politics.mideast	NG19:	talk.politics.misc

In A5, clusters overlap at medium level. In B5, clusters overlap substantially. Table 1 contains the results of MLHC. To accumulate sufficient statistics, for each newsgroup combination, we generate 10 datasets, each of which is a random sample of documents from the newsgroups (with 100 documents per newsgroup). The results in the table are the average over these 10 random sampled datasets. For comparison purpose, we also run K-means clustering. For each dataset, we run 10 K-means clustering from random seeds for cluster centroids and selecte the best result as determined by the K-means objective function value. Results are given in Table 1. Note that because percentage consistency results are close to 1, we give inconsistency = 1 - consistency.

From Table 1, the MLHC results have better clustering accuracy (last column of Table 1) compared to K-means clustering. More important is MLHC always provides clustering with better kmn cluster membership consistency. For 1mn-consistency, MLHC is perfect since this is guaranteed by MLHC. With this, it is not surprising that MLHC has substantially better 1nn-consistency than K-means method, about half as smaller. In all categories, MLHC has better knn/kmn consistency than K-means .

Table 1. Fractional knn and kmn inconsistency and clustering accuracy (last column) for newsgroup datasets A5 and B5. For dataset A5, 1nn inconsistency is 16.2% for K-means and 8.5% for MLHC.

	1nn	2nn	3nn	1mn	2mn	3mn	Accuracy
A5							
K-means	16.2	28.4	37.8	6.4	14.5	23.0	75.2%
MLHC	8.5	24.1	36.4	0	6.9	16.6	77.6%
B5							
K-means	23.1	39.4	50.6	8.5	21.6	32.8	56.3%
MLHC	10.2	28.9	45.0	0	9.3	21.3	60.7%

6 Summary

In this paper, we propose a modification of the standard AHC algorithm that allow an order-N faster implementation. The modification is based on the recognition that all 1mn-pairs in each iteration of AHC can be merged if the linkage function satisfies the cluster aggregate inequality. This leads to the multi-level hierarchical clustering algorithm. Many commonly used linkage functions satisfy this inequality and thus will benefit from this modification. We propose "close friends" cohesion as important feature of clustering and show that it is fully guarantees in the algorithm. This is further extended to cluster membership KNN consistency. Experiments on newsgroup show that kNN consistency is satisfied much better by MLHC than widely used algorithms such as K-means .

References

1. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification, 2nd ed.* Wiley, 2000.
2. B. Fung, K. Wang, and M. Ester. Large hierarchical document clustering using frequent itemsets. *Proc. SIAM Data Mining Conf*, 2003.
3. T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning.* Springer Verlag, 2001.
4. A.K. Jain and R.C. Dubes. *Algorithms for clustering data.* Prentice Hall, 1988.
5. A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999.
6. S.Y. Jung and T.-S. Kim. An agglomerative hierarchical clustering using partial maximum array and incremental similarity computation method. *Proc. SIAM Conf. on Data Mining*, pages 265–272, 2001.
7. G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32:68–75, 1999.
8. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE. Trans. on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.
9. S. Theodoridis and K. Koutroumbas. *Pattern Recognition.* Academic Press, 1999.

10. E.M. Voorhees. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing and Management*, 22:465–476, 1986.
11. H Xiong, M. Steinbach, P-N. Tan, and V. Kumar. Hicap:hierarchial clustering with pattern preservation. *Proc. SIAM Data Mining Conf*, pages 279–290, 2004.
12. H. Yu, J. Yang, and J. Han. Classifying large data sets using svms with hierarchical clusters. In *Proc. ACM Int'l Conf Knowledge Disc. Data Mining (KDD)*, pages 306–315, 2003.
13. T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. *Proc. ACM Int'l Conf. Management of Data (SIGMOD)*, pages 103–114, 1996.

Ensembles of Balanced Nested Dichotomies for Multi-class Problems

Lin Dong[1], Eibe F an [1], and S efan K a e [2]

[1] Department of Computer Science, University of Waikato, New Zealand
{ld21, eibe}@cs.waikato.ac.nz
[2] Department of Computer Science, Technical University of Munich, Germany
kramer@in.tum.de

Abstract. A system of nested dichotomies is a hierarchical decomposition of a multi-class problem with c classes into $c-1$ two-class problems and can be represented as a tree structure. Ensembles of randomly-generated nested dichotomies have proven to be an effective approach to multi-class learning problems [1]. However, sampling trees by giving each tree equal probability means that the depth of a tree is limited only by the number of classes, and very unbalanced trees can negatively affect runtime. In this paper we investigate two approaches to building balanced nested dichotomies—*class-balanced* nested dichotomies and *data-balanced* nested dichotomies—and evaluate them in the same ensemble setting. Using C4.5 decision trees as the base models, we show that both approaches can reduce runtime with little or no effect on accuracy, especially on problems with many classes. We also investigate the effect of caching models when building ensembles of nested dichotomies.

1 Introduction

Many eal-wo ld cla i ca ion oble a e l i-cla oble : hey involve a no inal cla va iable ha ha o e han wo val e . The e a e ba ically wo a oache fo ac ling hi y e of oble . One i o ada he lea ning algo i h o deal wi h l i-cla oble di ec ly, and he o he i o c ea e eve al wo-cla oble and fo a l i-cla edic ion ba ed on he e-dic ion ob ained f o he wo-cla oble . The la e a oach i a ealing beca e i doe no involve any change o he nde lying wo-cla lea ning algo i h . Well- nown exa le of hi y e of a oach a e e o -co ec ing o code [2] and ai wi e cla i ca ion [3], and hey of en e l in igni - can inc ea e in acc acy.

Recen ly, i ha been hown ha en e ble of ne ed dicho o ie a e a o i ing al e na ive o ai wi e cla i ca ion and anda d e o -co ec ing o code . In ex e i en wi h a deci ion ee lea ne and logi ic eg e - ion, hei e fo ance wa le de enden on he ba e lea ne ed, and hey yield obabili y e i a e in a na al and well-fo nded way if he ba e lea ne can gene a e wo-cla obabili y e i a e [1].

A d awbac of en e ble of ne ed dicho o ie , a lea co a ed o ai wi e cla i ca ion, i he igni can inc ea e in n i e. Al ho gh ai wi e cla i ca- ion e i e a lying he ba e lea ne $c*(c-1)/2$ i e fo a lea ning oble

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 84–95, 2005.

wi h *c* cla e , each lea ning oble i ch alle han he o iginal oble beca e only da a f o he elevan ai of cla e i con ide ed [3]. A ing a lea ning algo i h ha cale linea ly in he n be of in ance , and a - ing ha eve y cla ha he a e n be of in ance , he ove all n i e fo ai wi e cla i ca ion i linea in he n be of cla e .[1]

B ilding a ingle y e of ne ed dicho o ie in he a e e ing al o e- i e i e linea in he n be of cla e in he wo ca e, b he algo i h be a lied a xed, e - eci ed n be of i e o b ild an en e ble of ee (10 o 20 en e ble e be we e fo nd o be gene ally cien o achieve axi acc acy on he UCI da a e inve iga ed in [1]).

In hi a e we a e loo ing a a oache o ed cing he i e e i ed o b ild an en e ble of ne ed dicho o ie (END). Mo e eci cally, we o- o e o y e of ne ed dicho o ie (ECBND o EDBND , e ec ively). U ing C4.5 a he ba e lea ne , we how ha hey can i ove n i e, e ecially on oble wi h any cla e , wi h li le o no effec on cla i ca ion acc acy. We al o inve iga e he effec of caching odel : he a e wo-cla lea ning oble can occ l i le i e in an en e ble and i a e en e o e- e wo-cla ba e odel ha have al eady been b il fo evio y e of ne ed dicho o ie .

The a e i c ed a follow . In Sec ion 2 we di c he ba ic hod of b ilding END , o wo odi ed ve ion of he algo i h (ECBND and EDBND), and he e of caching odel . Sec ion 3 e en e i ical e l , ob ained f o 21 l i-cla UCI da a e , and eve al a i cial do ain wi h a va ying n be of cla e . Sec ion 4 a i e o nding .

2 Balanced Nested Dichotomies

A y e of ne ed dicho o ie i a a i ical odel ha i ed o deco o e a l i-cla oble in o l i le wo-cla oble (e.g. [4] in od ce i a a e hod fo e fo ing l i-cla logi ic eg e ion). The deco o i ion can be e e en ed a a bina y ee (Fig e 1). Each node of he ee o e a e of cla label , he co e onding aining da a and a bina y cla i e . A he ve y beginning, he oo node con ain he whole e of he o iginal cla label co e onding o he l i-cla cla i ca ion oble . Thi e i hen li in o wo b e . The e wo b e of cla label a e ea ed a wo " e a" cla e and a bina y cla i e i lea ned fo edic ing he . The aining da a e i li in o wo b e co e onding o he wo e a cla e and one b e of aining da a i ega ded a he o i ive exa le while he o he b e i ega ded a he nega ive exa le . The wo cce o node of he oo inhe i he wo b e of he o iginal cla label wi h hei co e onding aining da a e and a ee i b il by a lying hi oce ec ively. The oce nally each a leaf node if he node con ain only one cla label.

[1] Pairwise classification is actually even more beneficial when the base learner's run-time is worse than linear in the number of instances.

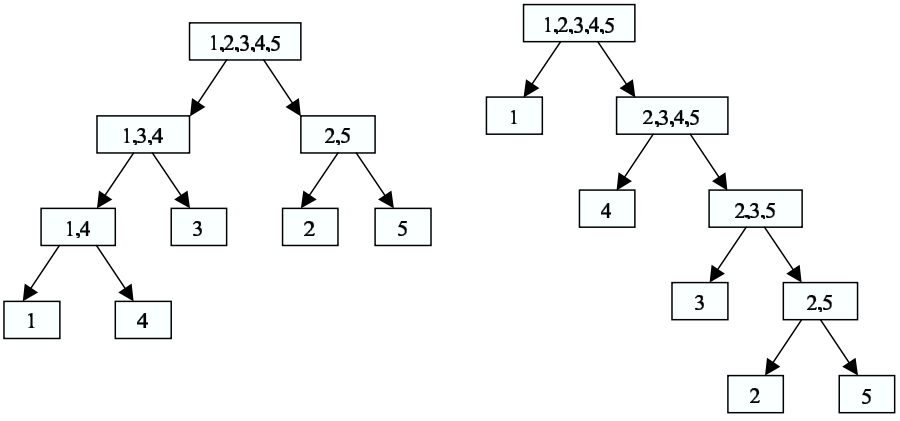


Fig. 1. Two different nested dichotomies for a classification problem with five classes

I i obvio . ha fo, any given c-cla . , oble , he , ee con ain c leaf node (one fo, each cla .) and $c-1$ in e, nal node . Each in e, nal node con ain a bina, y cla . i e, . A nice fea , e of . ing a . y e of ne ed dicho o ie fo, . l i-cla . , oble . i ha i yield cla . , obabili y e i a e in a . , aigh fo, wa, d fa hion. A . . ing he individ al wo-cla . cla . i e, . b il by he ba e lea, ne, , od ce cla . , obabili y e i a e fo, he co, , e onding wo-cla . , oble . —one fo, each b, anch ex ending f, o he co, , e onding in e, nal node—we can ob ain a cla . , obabili y e i a e fo, a a, ic la, leaf node (i.e. he cla . label in he . l i-cla . , oble ha i a . ocia ed wi h hi leaf node) by . i ly . l i lying oge he, he , obabili y e i a e ob ained f, o he bina, y cla . i e, . along he a, ic la, a h f, o he , oo node o ha leaf [1].

Howeve, , he, e i a , oble wi h he a lica ion of ne ed dicho o ie o . anda, d l i-cla . , oble . : he, e a, e any o . ible , ee . , c , e fo, a given . e of cla . e , and in he ab ence of , io, nowledge abo whe he, a a, ic la, deco o i ion i . o, e a , o , ia e, i i no clea, which one o . e. Fig , e 1 . how wo diffe, en . y e of ne ed dicho o ie fo, a ve-cla . , oble . The , oble i ha wo diffe, en , ee will of en lead o diffe, en , edic ion beca . e he bina, y cla . i e, . fo, each node a, e dealing wi h diffe, en wo-cla . , oble . . The . elec ion of he , ee . , c , e will infl ence he cla . i ca ion , e l . . Given hi ob e, va ion and he . cce . of en e ble lea, ning in yielding acc , a e , edic-ion , i . a e . en e o . e all o . ible ne ed dicho o ie fo, a given , oble and ave, age hei, , obabili y e i a e . Unfo, na ely hi i infea ible beca . e he n . be, of o . ible . y e of ne ed dicho o ie fo, a c-cla . , oble i $(2c-3)!!$ [1]. Hence i i nece . a, y o a e a . b e .

F, an and K, a e, [1] . a le , ando ly f, o he . ace of all o . ible , ee , giving each , ee e al , obabili y. Howeve, , i i no clea, whe he, hi i he be a , oach. In he ab ence of , io, nowledge, any . a ling che e ha doe no give , efe, en ial , ea . en o a a, ic la, cla . can be con ide, ed a . i able candida e. The , oble wi h , ando . a ling ba ed on a nifo, . di , ib ion

Table 1. Comparison of the number of possible trees

Number of classes	Number of nested dichotomies	Number of class-balanced nested dichotomies
2	1	1
3	3	3
4	15	3
5	105	30
6	945	90
7	10,395	315
8	135,135	315
9	2,027,025	11,340
10	34,459,425	113,400
11	654,729,075	1,247,400
12	13,749,310,575	3,742,200

ove, ee i ha he ee de h i only li i ed by he n be of cla e , and dee ee can a e a long i e o b ild. Con ide he ca e whe e he ee i a li , a in he econd ee hown in Fig e 1, and a e he wo la ge cla e a e e a a ed o la (cla e 2 and 5 in he exa le). Then all bina y lea ning oble will involved he wo la ge cla e , inc ing a high co a ional co fo he oce of b ilding he bina y cla i e in he ee.

2.1 Class-Balanced Nested Dichotomies

In ligh of he e ob e va ion we con ide wo diffe en a ling a egie in hi a e . The e hod i ba ed on balancing he n be of cla e a each node. In ead of a ling f o he ace of all o ible ee , we a le f o he ace of all ee , and b ild an en e ble of balanced ee . The advan age of hi e hod i ha he de h of he ee i g a an eed o be loga i h ic in he n be of cla e . We call hi an en e ble of cla -balanced ne ed dicho o ie (ECBND).

The n be of o ible cla -balanced ne ed dicho o ie i obvio ly alle han he o al n be of ne ed dicho o ie . The following ec ence ela ion de ne he n be of o ible cla -balanced ee :

$$T(c) = \begin{cases} \frac{1}{2}\binom{c}{c/2}T(\frac{c}{2})T(\frac{c}{2}) & : \text{ if } i \text{ even} \\ \binom{c}{(c+1)/2}T(\frac{c+1}{2})T(\frac{c-1}{2}) & : \text{ if } i \text{ odd} \end{cases}$$

whe e $T(1) = 1$ and $T(2) = 1$.

Table 1 how he n be of o ible y e of ne ed dicho o ie fo o 12 cla e fo he cla -balanced (CBND) and he ncon ained ca e (ND). I how ha a non- ivial n be of CBND can be gene a ed fo cla i ca ion oble wi h ve o o e cla e .

Fig e 2 how he algo i h fo b ilding a y e of cla -balanced ne ed dicho o ie . A each node he e of cla e i li in o e al i e b e (of co e, if he n be of cla e i odd, he i e will no be exac ly e al), and he

```
method buildClassBalancedNestedDichotomies(Dataset D, Set of classes C)

  if |C| = 1 then return
  P = subset of C, randomly chosen from all subsets of size ⌊|C|/2⌋
  N = C \ P
  D_p = all instances in D apart from those pertaining to classes in P
  buildClassBalancedNestedDichotomies(D_p, P)
  D_n = all instances in D apart from those pertaining to classes in N
  buildClassBalancedNestedDichotomies(D_n, N)
  D' = a two-class version of D created based on N and P
  classifierForNode = buildClassifier(D')
```

Fig. 2. Algorithm for generating class-balanced nested dichotomies

ba e lea, ning algo, i h i a lied o he da a co, e onding o he e wo b e . The algo, i h hen , ec , e n il only one cla i lef . I i a lied , e ea edly wi h diffe, en , ando n be, eed o gene, a e a co i ee of , ee .

2.2 Data-Balanced Nested Dichotomies

The, e i a o en ial , oble wi h he cla -balanced a , oach: o e l i-cla , oble a, e ve, y nbalanced and o e cla e a, e ch o, e o lo han o he, . In ha ca e a cla -balanced , ee doe no i ly ha i i al o , , (i.e. ha i exhibi abo he a e n be, of in ance in he wo cce o, node of an in e, nal node). Thi can nega ively affec , n i e if he ba e lea, ning algo, i h ha i e co lexi y wo, e han linea, in he n be, of in ance . Hence we al o con ide, a i le algo, i h fo, b ilding da a-balanced ne ed dicho o ie in hi a e, . No e ha hi e hod viola e he condi ion ha he a ling che e ho ld no be bia ed owa, d a a, ic la, cla : ba ed on hi che e, leaf node fo, la, ge, cla e will be loca ed highe, in he , ee , c , e. De i e hi o en ial d, awbac we decided o inve iga e hi che e e i, ically beca e i i di c l o ay how i o, an hi condi ion i in , ac ice.

Fig , e 3 how o , algo, i h fo, b ilding a y e of da a-balanced ne ed dicho o ie . I , ando ly a ign cla e o wo b e n il he i e of he , aining da a in one of he b e exceed half he o al a o n of , aining da a a he node. One o iva ion fo, ing hi i le algo, i h wa ha i i i o, an o ain ain a deg, ee of , ando ne in he a ign en of cla e o b e in o, de, o , e e, ve dive, i y in he co i ee of , ando ly gene, a ed y e of ne ed dicho o ie . Given ha we a, e ai ing fo, an en e ble of ne ed dicho o ie i wo ld no be advi able, even if i we, e co a ionally fea ible, o ai fo, an o i balance beca e hi wo ld eve, ely , e , ic he n be, of , ee ha can be gene, a ed. Even wi h o , i le algo, i h dive, i y ffe, when he cla di , ib ion i ve, y nbalanced. Howeve, , i i di c l o de, ive a gene, al ex , e ion fo, he n be, of , ee ha can o en ially gene, a ed by hi e hod beca e hi n be, de end on he cla di , ib ion in he da a e .

```
method buildDataBalancedNestedDichotomies(Dataset D, List of classes C)

  if |C| = 1 then return
  C = random permutation of C
  D_p = ∅, D_n = ∅
  do
    if (|C| > 1) then
      add all instances from D pertaining to first class in C to D_p
      add all instances from D pertaining to last class in C to D_n
      remove first and last class from C
    else
      add all instances from D pertaining to remaining class in C to D_p
      remove remaining class from C
  while (|D_p| < ⌊|D|/2⌋) and (|D_n| < ⌊|D|/2⌋)
  if ((|D_p| ≥ ⌊|D|/2⌋) then
     add instances from D pertaining to remaining classes in C to D_n
  else
     add instances from D pertaining to remaining classes in C to D_p
  P = all classes present in D_p, N = all classes present in D_n
  buildDataBalancedNestedDichotomies(D_p, P)
  buildDataBalancedNestedDichotomies(D_n, N)
  D' = a two-class version of D created based on N and P
  classifierForNode = classifier learned by base learner from D'
```

Fig. 3. Algorithm for generating data-balanced nested dichotomies

2.3 Computational Complexity

The o iva ion fo ing balanced ne ed dicho o ie i ha hi ed ce n- i e. In he following we analy e he co a ional co lexi y of co le ely ando and balanced ne ed dicho o ie . Le c be he n be of cla e in he da a e , and n be he n be of aining in ance . Fo i lici y, a e ha all cla e have an a oxi a ely e al n be of in ance in he (i.e. ha he n be of in ance in each cla i a oxi a ely n/c). We a e ha he i e co lexi y of he ba e lea ning algo i h i linea in he n be of aining in ance , and ha we can igno e he effec of he n be of a ib e .

In he wo ca e, a co le ely ando y e of ne ed dicho o ie can degene a e in o a li , and he o al n i e fo b ilding a l i-cla cla i e ba ed on hi ind of c e and he above a ion i

$$\sum_{i=0}^{c-2} \frac{c-i}{c} n = \frac{n}{c} \sum_{i=0}^{c-2} c - i = \frac{n}{c} \left((c-1)c - \sum_{i=0}^{c-2} i \right) = \frac{n}{c} \left((c-1)c - \frac{(c-2)(c-1)}{2} \right)$$
$$> \frac{n}{c} \left((c-1)c - \frac{c(c-1)}{2} \right) = \frac{(c-1)}{2} n.$$

Hence he wo -ca e i e co lexi y i linea in he n be of in ance and cla e .

Le now con ide he balanced ca e. A ing c i even, we have $\log c$ laye of in e nal node . In each laye , all he aining da a need o be oce ed

(beca . e he nion of all . b e . in each laye, i he o, iginal da a e). Given ha we have a . . ed ha he ba e lea, ne, . cale linea, ly in he n . be, of in ance , he ove, all , n i e beco e $n \log c$, i.e. i i loga, i h ic in he n . be, of cla . e and linea, in he n . be, of in ance .

A . . ing a ba e lea, ning algo, i h who e i e co lexi y i wo, . e han linea, , he advan age of he balanced . che e beco e even . o, e , ono nced (beca . e he . i e of he . b e . of da a con ide, ed a each node dec, ea e . o, e ic ly in hi . che e). No e al o ha he a . . ion of evenly di , ib ed cla . e i no . , ic ly nece . a, y. Thi can be . een by con ide, ing he wo, . ca e, whe, e one cla . ha al o all he in ance . The wo, . -ca e i e co lexi y fo, he nbalanced ca e , e ain linea, in he n . be, of cla . e in hi . i a ion, and he one fo, he balanced ca e loga, i h ic in he n . be, of cla . e .

Howeve, , in he ca e of a . ewed cla . di , ib ion i i o . ible o i , ove on he cla . -balanced . che e when he ba e lea, ning algo, i h ' . , n i e i wo, e han linea, . In ha ca e i . a e . en e o a e o divide he n . be, of in ance a evenly a o . ible a each node, . o a o , ed ce he . axi . a o n of da a con ide, ed a a node a ic ly a o . ible. Thi i why we have inve iga ed he da a-balanced a , oach di c . . ed above.

2.4 Caching Models

The, e i a f , he, o o, ni y o i , ove he , aining i e fo, en e ble of ne ed dicho o ie . I a, i e f, o he fac ha en e ble . e be, . . ay . ha, e . o e wo-cla . . , oble . . Con ide, Fig , e 1. In bo h , ee , a cla . i e, ha o be lea, ned ha . e a, a e cla . e 2 and 5. The e cla . i e, . will be iden ical beca . e hey a, e ba ed on exac ly he . a e da a. I i no . en ible o b ild he wice. Con e en ly we can cache . odel ha have been b il in a ha h able and , ed ce co a ional co lexi y f , he, .

A ex lained by F, an and K, a e, [1], he, e a, e $(3^c - (2^{c+1} - 1))/2$ o - . ible wo-cla . . , oble . fo, a c-cla . . , oble , i.e. g, ow h i ex onen ial in he n . be, of cla . e . Hence we can ex ec ha caching only . a e a diffe, ence fo, , ela ively . . all n . be, of cla . e . If we con ide, balanced dicho o ie , he n . be, of o . ible wo-cla . . , oble . i , ed ced. Con e en ly caching will be . o, e bene cial in he balanced ca e.

3 Experiments

In he following we e i, ically inve iga e he effec of o , , o o ed . odi ca- ion on , n i e and acc , acy. We . ed 21 . l i-cla . UCI da a e . [5]. The n . be, of cla . e va, ie f, o 3 o 26. We al o e, fo, ed . o e ex e, i en . wi h a, i cial da a ha exhibi . a la, ge, n . be, of cla . e . Fo, each . che e, we . ed 10 en e ble . e be, . (i.e. 10 . y e . of ne ed dicho o ie a, e gene, a ed and hei , obabili y e i a e a, e ave, aged o fo, . a , edic ion). J48, he i le- en a ion of he C4.5 deci ion , ee lea, ne, [6] f, o he We a wo, bench [7] wa . ed a he ba e lea, ne, fo, he ex e, i en . .

Table 2. Effect of model caching on ENDs for UCI datasets

Dataset	Number of classes	Number of instances	Training time for ENDs w/o caching	with caching	
iris	3	150	0.03 ± 0.02	0.01 ± 0.00	
balance-scale	3	625	0.28 ± 0.06	0.09 ± 0.04	•
splice	3	3190	4.56 ± 0.32	1.37 ± 0.14	•
waveform	3	5000	38.78 ± 0.65	11.42 ± 0.96	•
lymphography	4	148	0.06 ± 0.02	0.04 ± 0.02	•
vehicle	4	846	1.87 ± 0.11	1.08 ± 0.16	•
hypothyroid	4	3772	3.13 ± 0.47	1.75 ± 0.29	•
anneal	6	898	0.99 ± 0.08	0.82 ± 0.13	•
zoo	7	101	0.07 ± 0.02	0.06 ± 0.02	
autos	7	205	0.40 ± 0.05	0.36 ± 0.05	
glass	7	214	0.30 ± 0.03	0.27 ± 0.03	
segment	7	2310	6.61 ± 0.27	5.87 ± 0.37	•
ecoli	8	336	0.25 ± 0.04	0.23 ± 0.04	
optdigits	10	5620	72.53 ± 3.30	68.70 ± 3.00	•
pendigits	10	10992	49.30 ± 2.00	47.07 ± 2.12	•
vowel	11	990	4.21 ± 0.11	4.04 ± 0.16	•
arrhythmia	16	452	21.14 ± 1.01	20.76 ± 1.09	
soybean	19	683	1.02 ± 0.07	0.99 ± 0.06	
primary-tumor	22	339	0.63 ± 0.06	0.63 ± 0.06	
audiology	24	226	0.74 ± 0.06	0.74 ± 0.05	
letter	26	20000	317.53 ± 11.44	315.74 ± 11.53	

All ex e i en al e l a e ave age f o 10 n of a i ed 5-fold c o -valida ion (UCI da a e) o 3-fold c o -valida ion (a i cial da a). We al o e o anda d devia ion fo he 50 (UCI da a) o 30 (a i cial da a) individ- al e i a e . R n i e wa ea ed on a achine wi h a Pen i 4 3 GH oce o nning he Java Ho S o Clien VM (b ild 1.4.2_03) on Lin x, and i e o ed in econd . We e ed fo igni can diffe ence ing he co ec ed e a led - e [8].

3.1 Applying Caching to ENDs

In hi ec ion we di c he effec of caching individ al cla i e in an en e ble of ne ed dicho o ie . Table 2 ha he ave age aining i e fo END wi h and wi ho caching ba ed on a ha h able. Signi can i ove en in aining i e ob ained by caching a e a ed wi h a •.

The e l how ha he n i e dec ea e igni can ly fo 14 of he 21 UCI da a e (of co e, acc acy e ain iden ical). The i ove en i e ecially obvio on da a e wi h a all n be of cla e and a la ge n be of in- ance . Wi h a all n be of cla e one i o e li ely o enco n e he a e bina y cla i e in diffe en y e of ne ed dicho o ie in he en e ble. Wi h a la ge n be of in ance , o e i e i aved by avoiding eb ilding he a e bina y cla i e . Fo in ance, he aining i e on he wavefo da a e , which

Table 3. Comparison of training time on UCI datasets

Dataset	Number of classes	Training time ENDs	ECBNDs	EDBNDs
iris	3	0.01 ± 0.00	0.03 ± 0.02	0.02 ± 0.00
balance-scale	3	0.09 ± 0.04	0.09 ± 0.04	0.09 ± 0.04
splice	3	1.37 ± 0.14	1.45 ± 0.12	1.33 ± 0.20
waveform	3	11.42 ± 0.96	11.31 ± 0.56	11.24 ± 1.10
lymphography	4	0.04 ± 0.02	0.03 ± 0.02	0.03 ± 0.00
vehicle	4	1.08 ± 0.16	0.51 ± 0.06 •	0.52 ± 0.05 •
hypothyroid	4	1.75 ± 0.29	0.86 ± 0.12 •	0.92 ± 0.22 •
anneal	6	0.82 ± 0.13	0.63 ± 0.09 •	0.50 ± 0.13 •
zoo	7	0.06 ± 0.02	0.06 ± 0.03	0.06 ± 0.02
autos	7	0.36 ± 0.05	0.26 ± 0.04 •	0.25 ± 0.05 •
glass	7	0.27 ± 0.03	0.21 ± 0.04 •	0.20 ± 0.04 •
segment	7	5.87 ± 0.37	4.88 ± 0.34 •	4.98 ± 0.41 •
ecoli	8	0.23 ± 0.04	0.20 ± 0.03	0.21 ± 0.04
optdigits	10	68.70 ± 3.00	55.17 ± 1.91 •	55.03 ± 2.16 •
pendigits	10	47.07 ± 2.12	37.95 ± 1.53 •	38.40 ± 1.52 •
vowel	11	4.04 ± 0.16	3.62 ± 0.11 •	3.70 ± 0.12 •
arrhythmia	16	20.76 ± 1.09	19.20 ± 1.08 •	17.39 ± 1.56 •
soybean	19	0.99 ± 0.06	0.87 ± 0.06 •	0.85 ± 0.07 •
primary-tumor	22	0.63 ± 0.06	0.54 ± 0.06 •	0.54 ± 0.06 •
audiology	24	0.74 ± 0.05	0.64 ± 0.08 •	0.63 ± 0.09 •
letter	26	315.74 ± 11.53	273.45 ± 16.75 •	274.07 ± 16.84 •

ha 5000 in ance and only 3 cla e , dec ea e d a a ically f o 38.78 econd o 11.42 econd by ing ha h able . On he o he hand, fo he a hy h ia da a e , which ha 16 cla e and only 452 in ance , he aining i e dec ea e only ligh ly, f o 21.14 econd o 20.76 econd . F o Table 2, we al o ee ha he e i no igni can i ove en fo he aining i e when he n be of cla e exceed 11. The chance o enco n e he a e bina y cla i e in ho e i a ion beco e li i ed a he e a e o any o ible wo-cla oble . Mo eove , he n be of in ance in he e da a e (excl ding he le e da a) i all o ha he i e aved by ing ha h able i no no iceable. We al o e fo ed ex e i en wi h a i cial da a wi h even o e cla e and he e wa e en ially no diffe ence in n i e on ha da a.

3.2 Comparing ENDs, ECBNDs, and EDBNDs

A we have een, caching doe no hel when he e a e any cla e . In he following we will ee ha ing balanced ne ed dicho o ie hel in ho e ca e . We will loo a aining i e and hen he effec on acc acy.

Training time. Table 3 how he aining i e fo END , cla -balanced END (ECBND), and da a-balanced END (EDBND), on he UCI da a e . Model caching wa a lied in all h ee ve ion of END . A • indica e a ignif-ican ed c ion in n i e co a ed o END .

Table 4. Comparison of training time on artificial datasets

Number of classes	Number of instances	Training time ENDs	ECBNDs	EDBNDs
10	820	0.60 ± 0.09	0.58 ± 0.07	0.58 ± 0.07
20	1390	1.50 ± 0.12	1.42 ± 0.08	1.44 ± 0.09
30	1950	2.72 ± 0.11	2.31 ± 0.12 •	2.33 ± 0.12 •
40	2410	3.87 ± 0.16	3.18 ± 0.14 •	3.24 ± 0.13 •
50	3090	5.55 ± 0.23	4.54 ± 0.17 •	4.57 ± 0.20 •
60	3660	7.48 ± 0.29	5.86 ± 0.17 •	5.90 ± 0.25 •
70	4560	10.41 ± 0.36	8.23 ± 0.28 •	8.35 ± 0.30 •
80	5010	12.32 ± 0.43	9.56 ± 0.33 •	9.67 ± 0.31 •
90	5840	15.75 ± 0.53	12.62 ± 0.44 •	12.78 ± 0.34 •
100	6230	18.25 ± 0.61	13.61 ± 0.38 •	13.98 ± 0.44 •
150	9590	40.65 ± 1.90	27.63 ± 0.77 •	28.19 ± 0.65 •
200	12320	66.41 ± 2.95	42.37 ± 1.30 •	42.70 ± 1.30 •

The ˌe l . . how ha . ing cla . -balanced ne ed dicho o ie ˌe l . in . igni can ly ˌed ced ˌaining i e on 14 of he 21 da a e . . U ing he da a-balanced . che e al o hel . : EDBND a ˌe . igni can ly . o ˌe e cien han END on 14 da a e . , j . li e ECBND . Co a ˌed o cla . -balanced ˌee , da a-balanced ˌee a ˌe . igni can ly . o ˌe e cien on one da a e (a ˌ hy h ia). (Thi info ˌ a ion i no incl ded in Table 3.) Thi da a e ha an ex ˌe ely nbal-anced cla . di ˌib ion and hi i why he da a-balanced a ˌoach hel . .

The advan age of he balanced . che e i ˌe ˌic ed o da a e . wi h . o ˌe han 3 cla . e . On h ˌee-cla . da a e . , all ne ed dicho o ie a ˌe cla . -balanced, . o we wo ld no ex ec any . igni can diffe ˌence be ween END and ECBND . The ex ˌe i en al ˌe l . bea ˌ hi o .

Table 4 . how he ˌaining i e fo ˌ o ˌ 12 a ˌ i cial da a e . . To gene ˌ a e he e da a e . we . ed a cl . e ˌ gene ˌ a o ˌ and va ˌ ied he n . be ˌ of cl . e ˌ . f ˌ o 10 o 200. In ance in he . a e cl . e ˌ we ˌ e a . igned he . a e cla . label. Each in ance in he e da a e . con i . of one boolean a ˌib e and wo n . e ˌ ic a ˌib e . The a ˌib e val e ˌ ange we ˌ e . e o be diffe ˌ en b co ld ove ˌ la . A ˌib e val e we ˌ e gene ˌ a ed ˌ ando ly wi hin each cl . e ˌ . The n . be ˌ of in ance in each cl . e ˌ (i.e. cla .) wa al o ˌ ando ly gene ˌ a ed and va ˌ ied be ween 20 and 110.

The ˌe l . on he a ˌ i cial da a e . . how ha he balanced . che e exhibi a . igni can advan age in e ˌ . . of ˌ nning i e when 30 o ˌ . o ˌ e cla . e a ˌ e ˌ e en in he da a. The ˌ e wa no . igni can diffe ˌ ence in ˌ nning i e fo ˌ he wo balanced . che e (cla . -balanced v . . da a-balanced) on any of he da a e . . Thi indica e ha he cla . di ˌib ion in o ˌ a ˌ i cial da a e . i no . ewed eno gh fo ˌ he da a-balanced a ˌoach o hel .

Accuracy. I ˌove en . in ˌ n i e a ˌ e le . . ef l if hey affec acc ˌacy in a . igni can fa hion. Hence i i i o ˌ an o eval a e he effec of o ˌ ˌ o o ed . odi ca ion on acc ˌacy. Table 5 . how he e i a ed acc ˌacy fo ˌ END , ECB-ND , and EDBND on he UCI da a e . . We can . ee ha he ˌ e i no da a e wi h a . igni can diffe ˌ ence in acc ˌacy fo ˌ END and ECBND . . Thi i he

Table 5. Comparison of accuracy on UCI datasets

Dataset	Number of classes	Percent correct ENDs	ECBNDs	EDBNDs
iris	3	94.13 ± 3.84	94.13 ± 3.72	94.27 ± 3.81
balance-scale	3	79.92 ± 2.37	79.49 ± 2.41	79.78 ± 2.31
splice	3	94.75 ± 1.01	94.55 ± 0.98	93.07 ± 1.33 •
waveform	3	77.89 ± 1.88	77.53 ± 1.91	77.85 ± 2.06
lymphography	4	77.73 ± 7.47	76.63 ± 6.35	76.90 ± 6.93
vehicle	4	73.20 ± 2.92	72.36 ± 2.30	72.36 ± 2.30
hypothyroid	4	99.54 ± 0.26	99.51 ± 0.27	99.54 ± 0.28
anneal	6	98.63 ± 0.80	98.44 ± 0.75	98.53 ± 0.62
zoo	7	93.66 ± 5.67	93.87 ± 4.61	93.88 ± 4.50
autos	7	76.20 ± 6.11	74.83 ± 6.62	75.32 ± 7.10
glass	7	72.82 ± 7.42	73.51 ± 6.17	72.25 ± 6.84
segment	7	97.45 ± 0.83	97.35 ± 0.80	97.39 ± 0.87
ecoli	8	85.60 ± 4.11	85.36 ± 4.06	84.88 ± 4.13
optdigits	10	96.99 ± 0.49	97.14 ± 0.45	97.18 ± 0.50
pendigits	10	98.59 ± 0.27	98.76 ± 0.25	98.76 ± 0.26
vowel	11	88.31 ± 2.66	89.98 ± 2.47	89.24 ± 2.79
arrhythmia	16	72.59 ± 3.24	72.82 ± 4.11	71.51 ± 3.55
soybean	19	93.90 ± 1.63	94.49 ± 1.69	94.36 ± 1.78
primary-tumor	22	44.72 ± 5.04	46.28 ± 4.61	45.96 ± 4.62
audiology	24	78.46 ± 5.44	79.66 ± 5.12	79.48 ± 5.23
letter	26	94.33 ± 0.37	94.50 ± 0.36	94.51 ± 0.35

Table 6. Comparison of accuracy on artificial datasets

Number of classes	Number of instances	Percent correct ENDs	ECBNDs	EDBNDs
10	820	78.08 ± 1.94	78.34 ± 2.35	78.32 ± 2.32
20	1390	77.79 ± 1.87	77.21 ± 1.44	77.47 ± 1.66
30	1950	77.09 ± 1.61	76.93 ± 1.53	76.85 ± 1.46
40	2410	76.64 ± 1.24	76.56 ± 1.39	76.46 ± 1.24
50	3090	76.26 ± 1.09	76.17 ± 1.26	76.25 ± 1.19
60	3660	76.43 ± 1.08	76.33 ± 1.04	76.37 ± 0.95
70	4560	73.58 ± 1.12	73.27 ± 0.97	73.50 ± 0.90
80	5010	75.85 ± 1.06	75.61 ± 0.94	75.71 ± 0.87
90	5840	76.41 ± 0.84	76.40 ± 0.91	76.41 ± 0.87
100	6230	76.59 ± 0.77	76.54 ± 0.73	76.50 ± 0.85
150	9590	75.92 ± 0.66	75.89 ± 0.72	75.86 ± 0.62
200	12320	75.89 ± 0.51	75.67 ± 0.51	75.73 ± 0.49

de i ed o co e. Fo EDBND , he e i one h ee-cla da a e (lice) whe e he acc acy i igni can ly ed ced co a ed o END . The lice da a ha a ewed cla di ib ion, whe e one cla ha abo half he in ance and he e i evenly di ib ed a ong he e aining wo cla e . We ea ed he dive i y of he h ee y e of en e ble on hi da a e ing he a a

. a i ic. Thi . a i ic can be . ed o . ea , e ag, ee en be ween ai . of en- . e ble . e be, . [9]. Fo, EDBND , he . ean a a val e ove, all ai . , . ea , ed on he , aining da a, wa 0.96, which wa indeed highe, han he . ean a a val e fo, END and ECBND (0.94 and 0.93 , e ec ively). Thi indica e ha , ed c ion in dive, i y i he , ea on fo, he d, o in e, fo, ance.

Table 6 . how he . a e info, a ion fo, he a, i cial da a e . . In hi ca e he, e i no a . ingle da a e whe, e he, e i a . igni can diffe, ence in acc , acy be ween any of he . che e .

4 Conclusions

En e ble of ne ed dicho o ie have , ecen ly been . hown o be a ve, y , o i ing . e a lea, ning . che e fo, . l i-cla . , oble . . They , od ce acc , a e cla . i - ca ion and yield cla . , obabili ie e i a e in a na , al way. In hi a e, we have . hown ha i i o . ible o i , ove he , n i e of hi . e a lea, n- ing . che e wi ho affec ing acc , acy. A . i le way o i , ove , n i e fo, , oble . wi h a . . all n . be, of cla . e i o cache wo-cla . . odel and , e- . e he in diffe, en . e be, . of an en e ble of ne ed dicho o ie . On , oble . wi h . any cla . e we have . hown ha . ing cla . -balanced ne ed dicho o ie . igni can ly i , ove , n i e, wi h no . igni can change in acc , acy. We have al o , e en ed a da a-balanced . che e ha can hel o i , ove , n i e f , he, when he, e a, e . any cla . e and he cla . di , ib ion i highly . ewed.

References

1. Frank, E., Kramer, S.: Ensembles of nested dichotomies for multi-class problems. In: Proc Int Conf on Machine Learning, ACM Press (2004) 305–312
2. Dietterich, T., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research **2** (1995) 263–286
3. Fürnkranz, J.: Round robin classification. Journal of Machine Learning Research **2** (2002) 721–747
4. Fox, J.: Applied Regression Analysis, Linear Models, and Related Methods. Sage (1997)
5. Blake, C., Merz, C.: UCI repository of machine learning databases. University of California, Irvine, Dept. of Inf. and Computer Science (1998) [www.ics.uci.edu/∼mlearn/MLRepository.html].
6. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Los Altos, CA (1992)
7. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (2000)
8. Nadeau, C., Bengio, Y.: Inference for the generalization error. Machine Learning **52** (2003) 239–281
9. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning **40** (1998) 139–157

Protein Sequence Pattern Mining with Constraints

Pedro Gabriel Ferreira* and Paulo J. Azevedo**

University of Minho, Department of Informatics,
Campus of Gualtar, 4710-057 Braga, Portugal
{pedrogabriel, pja}@di.uminho.pt

Abstract. Considering the characteristics of biological sequence databases, which typically have a small alphabet, a very long length and a relative small size (several hundreds of sequences), we propose a new sequence mining algorithm (*gIL*). *gIL* was developed for linear sequence pattern mining and results from the combination of some of the most efficient techniques used in sequence and itemset mining. The algorithm exhibits a high adaptability, yielding a smooth and direct introduction of various types of features into the mining process, namely the extraction of rigid and arbitrary gap patterns. Both breadth or a depth first traversal are possible. The experimental evaluation, in synthetic and real life protein databases, has shown that our algorithm has superior performance to state-of-the art algorithms. The use of constraints has also proved to be a very useful tool to specify user interesting patterns.

1 Introduction

In the development of sequence pattern mining algorithms, two communities can be considered: the [illegible] and the [illegible] community. The algorithms from the Data Mining community inherited some characteristics from the association rule mining algorithms. They are best suited for data with many (from hundred of thousands to millions) sequences with a relative small length (from 10 to 20), and an alphabet of thousands of events, e.g. [9,7,11,1]. In the bioinformatics community, algorithms are developed in order to be very efficient when mining a small number of sequences (in the order of hundreds) with large lengths (few hundreds). The alphabet size is typically very small (ex: 4 for DNA and 20 for protein sequences). We emphasize the algorithm Teiresias [6] as a standard.

The major problem with Sequence pattern mining is that it usually generates too many patterns. When databases attain considerable size or when the average

* Supported by a PhD Scholarship (SFRH/BD/13462/2003) from Fundação Ciência e Tecnologia.

** Supported by Fundação Ciência e Tecnologia - Programa de Financiamento Plurianual de Unidades de I & D, Centro de Ciências e Tecnologias da Computação - Universidade do Minho.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 96–107, 2005.

length of the sequences is very long, the mining process becomes computationally expensive or simply infeasible. This is often the case when we are mining biological data like proteins or DNA. Additionally, the user interpretation of the results turns out to be a very hard task since the interesting patterns are blurred into the huge amount of outputted patterns. The solution to this problem can be achieved through the definition of alternative interesting measures besides support, or with user imposed restrictions to the search space. When properly integrated in the mining process these restrictions reduce the computation demands in terms of time and memory, allowing to deal with datasets that are otherwise potentially untractable. These restrictions are expressed through what is typically called as [illegible]. The use of Constraints enhances the database queries. The runtime reduction grants the user with the opportunity to interactively refine the query specification. This can be done until an expected answer is found.

2 Preliminaries

We consider the special case of linear sequences databases. A database D is as a collection of linear sequences. A [illegible] is a sequence composed by successive atomic elements, generically called events. Examples of this type of databases are protein or DNA sequences or website navigation paths. The term [illegible] is used to make the distinction from the transactional sequences, that consist in sequences of [illegible] (usually called as [illegible]). Given a sequence S, S' is subsequence of S if S' can be obtained by deleting some of the events in S. A sequence pattern is called a [illegible] if it is found to be subsequence of a number of sequences in the dataset greater or equal to a specified threshold value. This value is called [illegible], σ, and is defined as an user parameter. The [illegible] represents the list of sequence identifiers where the pattern occurs. The cardinality of this list corresponds to the support of that pattern.

Considering patterns in the form $A_1 - x(p_1, q_1) - A_2 - x(p_2, q_2) - ... A_n$, a sequence pattern is an [illegible] when a variable (zero or more) number of gaps exist between adjacent events in the pattern, i.e. $p_i \leq q_i, \forall i$. Typically a variable gap with n minimum and m maximum number of gaps is described as $-x(n, m)-$. In the sequences $< 1\ 5\ 3\ 4\ 5 >$ and $< 1\ 2\ 2\ 3 >$ exists an arbitrary gap pattern $1-x(1, 2)-3$. A [illegible] is a pattern where gaps contain a fixed size for all the database occurrences of the sequence pattern, i.e. $p_i = q_i, \forall i$. To denote a rigid gap the $-r(n)-$ notation is used, where n is the size of the gap. The $1-r(2)-3$ is a pattern of length 4, in the sequences $< 1\ 2\ 5\ 3\ 4\ 5 >$ and $< 1\ 1\ 6\ 3 >$. Each gap position is denoted by the "." (wildcard) symbol, meaning that it matches any symbol of the alphabet. A pattern belongs to one of three classes: [illegible] or [illegible]. A sequence pattern is [illegible] if it is not contained in any other pattern, and [illegible] when all its extensions have an inferior support than itself. The [illegible] refers to when all the patterns are enumerated. When extending a sequence pattern $S =< s_1\ s_2 \ldots s_n >$, with a new event s_{n+1}, then

S is called a [illegible] and $S' =< s_1\ s_2\ \ldots s_n\ s_{n+1} >$ the [illegible]. If an event b occurs after a in a certain sequence, we denoted it as: $a \rightarrow b$, and a is called the [illegible], $pred(a \rightarrow b) = a$, and b the [illegible], $succ(a \rightarrow b) = b$. The pair is frequent if it occurs in at least σ sequences of the database.

Constraints represent an efficient way to prune the search space [9,10]. Considering the user's point of view, it also enables to focus the search on more interesting sequence patterns. The most common and generic types of constraints are:

- [illegible]: restricts the set of the events ([illegible]) that may appear in the sequence patterns,
- [illegible]: defines the ([illegible]) minimum distance or the maximum distance ([illegible]) that may occur between two adjacent events in the sequence patterns,
- [illegible]: defines the maximum distance ([illegible]) between the first and the last event of the sequence patterns.
- [illegible]: determines that the extracted patterns start with the specified events ([illegible]).

Another useful feature in sequence mining, in particular to protein pattern mining, is the use of [illegible]. When used during the mining process an event can be substituted by another event belonging to the same set. A [illegible] hierarchy of relations can be represented through substitution sets.

Depending on the target application of the frequent sequence patterns other measures of interest and scoring can be applied as posterior step of the mining process. Since the closed and the maximal patterns are not necessarily the most interesting we designed our algorithm in order to find all the frequent patterns. From the biological point of view, rigid patterns allow to find more well conserved regions, while arbitrary patterns permit the cover of a large number of sequences in the database.

The problem we address in this paper can be formulated as follow: given a database D of linear sequences, a minimum support, σ, and the optional parameters [illegible], find [illegible] the [illegible] or [illegible] frequent sequence patterns that respect the defined constraints.

3 Algorithm

The proposed algorithm uses a Bottom-Up search space enumeration and a combination of frequent pairs of events to extend and find all the frequent sequence patterns. The algorithm is divided in two phases: [illegible] and [illegible]. Since the frequent sequences are obtained from the set of frequent pairs, the first phase of the algorithm consists in traversing all the sequences in the database and building two auxiliary data structures. The first structure contains the set of all pairs of events found in the database. Each pair representation points to the sequences where they appear (through a sequence identifier bitmap). The second data structure consists of a vertical representation

of the database. It contains the positions or offsets of the events in the sequences where they occur. This information is required to ensure that the order of the events along the data sequence is respected. Both data structures are thought for quick information retrieval. At the end of the scanning phase we obtain a map of all the pairs of events present in the database and a vertical format representation of the original database. In the second phase the pairs of events are successively combined to find all the sequence patterns. These operations are fundamentally based on two properties:

Property 1 (Anti-Monotonic). [illegible]

Property 2 (Sequence Transitive Extension). [illegible] $S =< s_1 \ldots s_n >$, C_S [illegible] O_S [illegible] S [illegible] C_S [illegible] $P = (s_j \rightarrow s_m)$, C_P [illegible] O_P [illegible] $succ(P)$ [illegible] C_P [illegible] $succ(S) = pred(P)$, [illegible] $s_n = s_j$, [illegible] $E =< s_1 \ldots s_n s_m >$ [illegible] C_E, [illegible] $C_E = \{X : \forall X \ in \ C_S \cap C_P, O_P(X) > O_S(X)\}$

Hence, the basic idea is to successively extend a frequent pair of events with another frequent pair, as long as the predecessor of one pair is equal to the successor of the other. This joining step is sound provided that the above mentioned properties (1 and 2) are respected. The joining of pairs combined with a breadth first or a depth first traversal yields all the frequent sequences patterns in the database.

3.1 Scanning Phase

The first phase of the algorithm consists in the following procedure: For each sequence in D, obtain all ordered pairs of events, without repetitions. Consider the sequence 5 in the example database of table 1(a). The obtained pairs are: $1 \rightarrow 2$, $1 \rightarrow 3$, $1 \rightarrow 4$, $2 \rightarrow 2$, $2 \rightarrow 3$, $2 \rightarrow 4$ and $3 \rightarrow 4$. During the determination of the pairs of events the first auxiliary data structure, that consists of an N-bidimensional matrix, is built and updated. N corresponds to the size of the alphabet. The N^2 cells in the matrix correspond to the N^2 possible combinations of pairs. We call this structure the [illegible]. Each $Cell(i, j)$ contains the information relative to the pair $i \rightarrow j$. This information consists of a bitmap that indicates the presence (1) or the absence (0) in the respective sequence

Table 1. (a) Parameters used in the synthetic data generator; (b) Properties of the proteins datasets

Symbol	Meaning
S	Number of Sequences (x 10^3)
L	Avg. Length of the sequences
R	Alphabet Size
P	Distribution Skewness

DataSet	NumSeq	AlphabetSize	AvgLen	MinLen	MaxLen
Yeast	393	21	256	15	1859
PSSP	396	22	158	21	577
nonOM	60	20	349	53	1161
mushroom	8124	120	23	23	23

(i-th bit corresponds to the sequence i in D) and an integer that contains the support count. This last value allows a fast support checking. For each pair $i \rightarrow j$ we update the respective $Cell(i, j)$ in the Bitmap Matrix, by activating the bit corresponding to the sequence where the pair occurs and incrementing the support counter. As an example, for the pair $1 \rightarrow 3$, the $Cell(1, 3)$ is represented in figure 1(b):

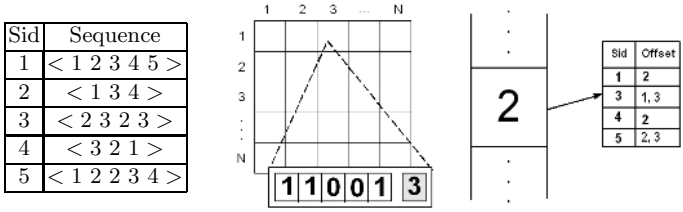


Fig. 1. (a) An example database; (b) Content of the Cell(1,3) in the Bitmap Matrix; (c) Representation of event 2 in the Offset Matrix

This means that the pair occurs in the database sequence 1, 2 and 5 and has a support of 3. Simultaneously, as each event in the database is being scanned, a second data structure called is also being built. Conceptually, this data structure consists of an adjacency matrix that will contain all the offset (positions) of all the events in the entire database. Each event is a key that points to a list of pairs $<$, $>$, where is a list of all the positions of the event in the sequence . Thus, the Offset Matrix is a vertical representation of the database. Figure 1(c) shows the information stored in the Offset Matrix for the event 2.

3.2 Sequence Extension Phase

We start by presenting how arbitrary gap patterns are extracted. In section 3.4 we will show how easily our algorithm can be adapted to extract rigid gap patterns. For implementing the extension phase we present two tests (algorithms) that conjunctively are necessary and sufficient conditions to consider as frequent a new extended sequence.

This is a quick test that implements property 1. The function gets the correspondent bitmaps of S and P. The intersection operation is also very fast and simple and the support function retrieves the support of the intersection bitmap. This test allows the verification of a condition for the extended sequence to be frequent. A second test is necessary

input : ( ); ( ); ( )
= ( ) and = ( )
= ∩
if ( ) ≥ , then return OK.

Algorithm 1: Support Test

```
   input : ,( ); ( ); ( . ); ( .)
1          = ( ,);
2     =     ( );
3     = 0;
4  foreach Sid in seqLst do
5          = offsetLst( , );
6          = offsetLastEvent( , );
7          = offsetStartEvent( , );
8      if ∃ ∈ , then
9              =     + 1;
10         if ( − ) then = ( − )
11         if ( − ) then = ( − )
12     end
13             ( , );
14 end
15 if     ≥  then
16     return OK;
17 end
```

Algorithm 2: Order Test

to guarantee that the order of the events is kept along the sequences that $C_{S'}$ bitmap points to.

Algorithm 2 assumes that, for each frequent sequence, additional information besides the sequence event list is kept during the extension phase. Namely, the corresponding bitmap that for the case exposed in algorithm 1 will be $C_{S'}$ if S' is determined to be frequent. Also two offset lists in the form $< \ , \ >$ are kept. One will contain the offset of the last event of the sequence, , and will be used for the "Order Test". The second, , contains the offset of the first event of the sequence pattern in all the Sid where it appears. This will be used when the verification of the window constraint is performed. In the Order Test, given a bitmap resulted from the support test, the function returns the list of the sequence identifiers for the bitmap. The function returns a list of offset values of the event in the respective Sid. For each sequence identifier it is tested whether the extension pair has an offset greater than the offset value of the extended sequence. This implements the computation of C_E and the offsetList of $succ(E)$ as in property 2. At line 13 the $diffTest$ function performs a simple test to check whether the minimum support is still reachable. At the end of the procedure (lines 15 to 17) it is tested whether the order of the extended sequence pattern is respected in a sufficient number of database sequences. In the positive case the extended sequence is considered frequent. Given algorithm 1 and 2, property 3 guarantees the necessary and sufficient conditions to safely extend a base sequence into a frequent one.

Property 3 (Frequent Extended Sequence). σ, $S =< E_1 \ldots E_n >$, $|S| \geq 2$, $P = E_k \rightarrow E_w$, $E_n = E_k$, $S' =< E_1 \ldots E_n \ g_{n,k} E_k >$, $g_{n,k} = -x(n,m) - r(n) -$ OK

3.3 Space Search Traversal

Guided by the Bitmap Matrix the search space can be traversed using two possible approaches: or . For both cases the set of the frequent

sequences starts as the set of frequent pairs. In the depth first mode it starts with a sequence of size 2 that is successively expand until it can not be further extended. Then we backtrack and start extending another sequence. The advantage of this type of traversal is that we don't need to keep all the intermediary frequent sequence information, in contrast with the breadth first traversal where all the information of the sequences size k need to be kept before the sequences of size k+1 are generated. This yields is some cases, a significant memory reduction.

3.4 Rigid Gap Patterns

The algorithm described in 2 is designed to mine arbitrary gap patterns. Using gIL to mine rigid gap patterns requires only minor changes in the Order Test algorithm. Lines 4 to 11 in algorithm 2 are rewritten in algorithm 3. In this algorithm, first it is collected (in gapLst) the size of all the gaps for a certain sequence extension. Next, for each gap size it is tested whether the extended sequence is frequent. One should note that for rigid gap patterns, two sequence patterns with the same events are considered different if the gaps between the events have different size, e.g., $< 1 \cdot \cdot\ 2 >$ is different from $< 1 \cdot \cdot \cdot\ 2 >$.

```
 1  foreach Sid in seqLst do
 2          = offsetLst(     ,     );
 3          = offsetLastEvent(     ,     );
 4          = offsetStartEvent(     ,     );
 5      if ∃    ∈     ,              then
 6                =     −     ;                .     (     );
 7      end
 8  end
 9  foreach R in gapLst do
10      foreach Sid in seqLst do
11          Repeat Step 2 to 4;
12          if ∃    ∈     , (     −     ) =     then
13                    =         + 1;
14          end
15      end
16      if     ≥     then
17          return OK;
18      end
19  end
```

Algorithm 3: Algorithm changes to mine rigid gap patterns

4 Constraints

The introduction of constraints in the gIL algorithm like [illegible] is a straightforward process and translates into a considerable performance gain. These efficiency improvements are naturally expected since (depending on the values of the constraints) the search space can be greatly reduced. The introduction of [illegible] is also very easy to achieve. Implementing events exclusion constraint and substitution sets turns out to be a natural operation. Simple changes in the Bitmap Matrix (that guides the sequence extension) and in the Offset Matrix (discriminates the positions of the events in every sequence where they occur) enable this implementations. The new features are introduced between the scanning phase and the sequence extension phase. The min/max gap and window constraints constitute an additional to be applied when the sequence is extended.

4.1 Events Exclusion, Start Events and Substitution Sets

The event exclusion constraint is applied by traversing the rows and columns of the Bitmap Matrix where the excluded events occurs. At that positions the support[1] count variable in the respective cells is set to zero. Start events constraints are also straightforwardly implemented by allowing extensions only to the events in $StartEventSets$.

When substitution sets are activated, one or more sets of equivalent events are available. For each set of equivalent events one has to form the union of the rows (horizontal union) and columns (vertical union) in the Bitmap Matrix, where those events occur. The vertical union is similar to the horizontal union. Moreover, for all the equivalent events, one needs to pairwisely intersect the sequences where they occur and then perform the union of the offsetLists for the intersected sequences. This results in the new offsetLists of the equivalent events.

4.2 Min / Max Gap and Window Size

These constraints are trivially introduced in the "Order Test". In algorithm 2, the test in line 8 is extended with three additional tests: $(X - Y) < maxGap\ AND\ (X - Y) > minGap\ AND\ (X - W) < windowSize$.

5 Experimental Evaluation

We evaluated our algorithm along different variables using two collections of synthetic and real datasets. To generate the synthetic datasets we developed a sequence generator based on the Zipfian distribution. This generator receives the following parameters (see table 1(a)): number of sequences, average length of the sequences, alphabet size and a parameter p that expresses the skewness of the distribution. This generator has allowed us to generate sequences with a relative small alphabet. The evaluated variables for this datasets were: [illegible], [illegible] ([illegible]), and [illegible]. Additionally, we tested the mushroom dataset used at the FIMI workshop [4]. To represent real life data, we used several datasets of proteins. The Yeast ([illegible]) dataset is available at [5] and PSSP used for protein secondary structure prediction [3]. We also used a subset of the non Outer Membrane proteins obtained from [8]. The properties for this datasets are summarized in table 1(b). It is interesting to notice that, for all datasets, gIL's scanning phase time is residual (less than 0.4 seconds).

Since gIL finds two types of patterns we performed evaluation against two different algorithms. Both are in memory algorithms, assuming that the database completely fits into main memory. For the arbitrary gap patterns from the all patterns class we compared gIL with the SPAM [1] algorithm. SPAM has shown to outperform SPADE [11] and PrefixSpan [7] and is a state-of-the-art algorithm

[1] Future interactions on this dataset still have the Bitmap Matrix intact since the bitmaps remain unchanged.

in transactional sequence pattern mining. The datasets suffer a conversion into the transactional dataset format, in order to be processed by SPAM. In this conversion each customer is considered as a sequence and each [illegible] contains a unique item (event).

For the rigid gap patterns we compared gIL with Teiresias [6], a well known algorithm from the bioinformatics community. It can be obtained from [2]. It is, as far as we know, the most complete and efficient algorithm for mining closed (called "most specific" in their paper) frequent rigid gap patterns. Closed patterns are a subset of all frequent sequence patterns. In this sense, gIL (which derives all patterns) tackles a more general problem and consequently considers a much larger search space than Teiresias. Besides minimum support, Teiresias uses two additional parameters. L and W are respectively the number of non-wild cards events in a pattern and the maximum spanning between two consecutive events. Since gIL starts by enumerating patterns with size 2, we will set L=2 and W to the maxGap value. All the experiments[2] were performed using exact discovery, i.e. without the use of substitution sets, and on a 1.5GHz Intel Centrino machine with 512MB of main memory, running windows XP Professional. The applications were written in C/C++ language.

5.1 Arbitrary Gap Patterns Evaluation

We start by comparing the efficiency of SPAM with the gIL algorithm without constraints. In figure 2(a) and 2(b) we tested different values of support for two datasets of $1K$ and $2K$ respectively. The sequences have an average length of 60 and an alphabet of 20 events. It was clear in these two experiments that for relative smaller dataset sizes and lower support values gIL becomes more efficient than SPAM. Figure 2(c) shows the scalability of the algorithms in respect to the dataset size for a support of 30%. This graphic shows that gIL scales well in relation to the dataset size.

In order to test a dataset with different characteristics, namely larger alphabet size, small length and greater dataset size, we used the Mushroom dataset, see figure 3(a). In figure 3(b) we have runtimes of gIL for datasets with one thousand sequences and different values of average sequence length. It was imposed a maxGap constraint of 15. As we observed during all the experiments, there is a critical point in the support variation, typically between 10% and 20%, that translates into an explosion of number of frequent patterns. This leads to an exponential behaviour in the algorithm's runtime. Even so, we can see that gIL shows similar behaviour for the different values of sequence length. Figure 3(c) measures the relative performance time, i.e. the ratio between the mining time with constraints and without constraints. These values were obtained for a support of 70%. Runtime without constraints was 305 seconds. It describes the behaviour of the algorithm when decreasing the maxGap and the Window values.

[2] Further details and results can be obtained from an extended version of this paper.

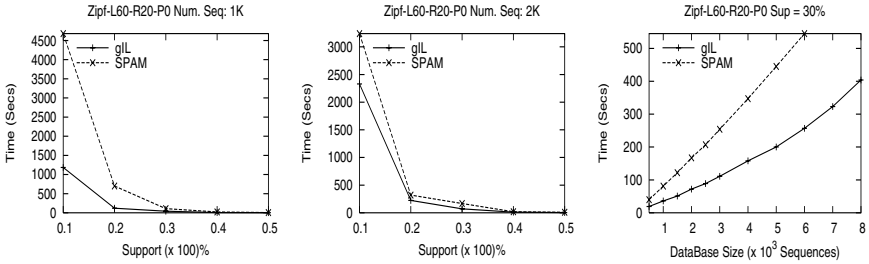


Fig. 2. (a) Support variation with Zipf database size=1K; (b) Support variation with Zipf database size=2K; (c) Scalability of gIL w.r.t. database size with a support of 30%

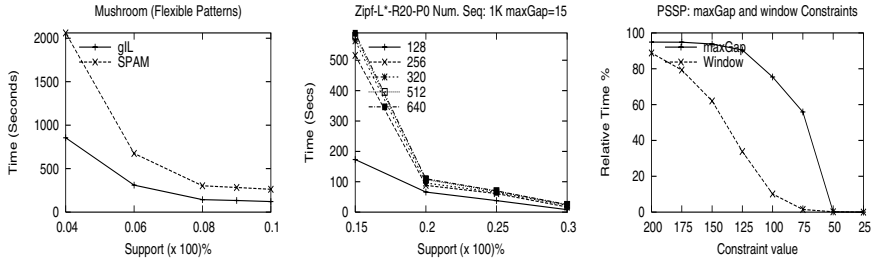


Fig. 3. (a) Support variation for the Mushroom dataset; (b) Scalability of gIL w.r.t sequence size for different support values (c) Performance evaluation using maxgap and windowgap constraints

In respect to memory usage both algorithms showed a low memory demand for all the datasets. For the Mushroom dataset which was the most demanding in terms of memory, SPAM used a maximum of 9 MB for a support of 4% and gIL a constant memory usage of 26 MB for all the support values. gIL shows a constant and support independent memory usage since once the data structures are built for a given dataset they remain unchanged.

5.2 Rigid Gap Patterns Evaluation

In order to assess the performance of gIL in the mining of rigid gap patterns we compared it with Teiresias [6], for different proteins datasets. In figure 4(a) and 4(b) the Yeast dataset was evaluated for two values of maxGap(W), 10 and 15. The results showed that gIL outperforms Teiresias by an order of magnitude. When comparing the performance of the algorithms in relation to the PSSP (figure 4(c)) and the nonOM (figure 5(a)) datasets, for a maxGap of 15, gIL outperforms Teiresias by a factor of 2 in the first case. This difference becomes more significant in the second case. The nonOM dataset has a greater average sequence length, but a small dataset size. This last characteristic results into a smaller bitmap length yielding a significant performance improvement. As

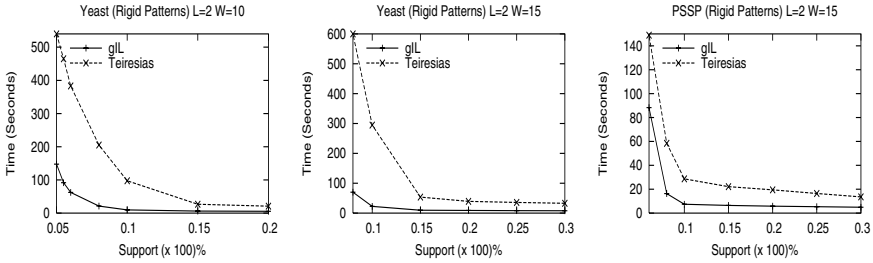


Fig. 4. (a) Support variation for the Yeast dataset, with L=2 and W(maxGap) = 10; (b) Support variation for the Yeast dataset, with L=2 and W(maxGap) = 15; (c) Support variation for the PSSP dataset, with L=2 and W(maxGap) = 15

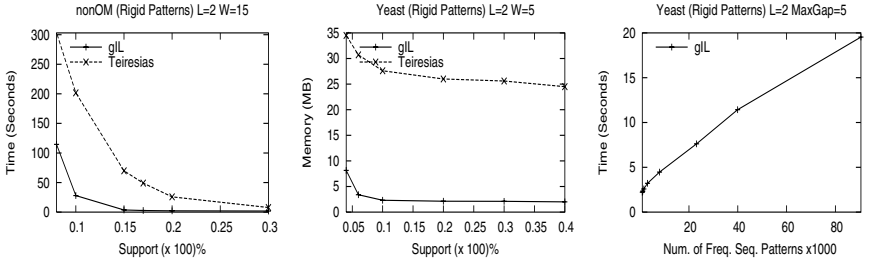


Fig. 5. (a) Support variation for the nonOM dataset, with L=2 and W(maxGap) = 15; (b) Memory usage for the Yeast dataset, with L=2 and W(maxGap) = 5; (c) Scalability of gIL w.r.t number of sequences for the Yeast dataset

we already verified in the arbitrary gap experiments, gIL memory usage maintains nearly constant for all the tested support values (figure 5(b)). Figure 5(c) shows the linear scalability of gIL in relation to the number of frequent sequence patterns.

6 Conclusions

We presented an algorithm called , , suitable to work with databases of linear sequences with a long average length and a relative small alphabet size. Our experiments showed that for the particular case of the proteins datasets, gIL exhibits superior performance to state-of-the-art algorithms. The algorithm has a high adaptability, and thus it was easily changed to extract two different types of patterns: arbitrary and rigid gap patterns. Furthermore, the data organization allows a straightforward implementation of constraints and substitution sets. These features are pushed directly into the mining process, which in some cases enables the mining in useful time of otherwise untractable problems. In this sense gIL is an interesting and powerful algorithm to be applied in a broader range of domains and in particular suitable for biological data. Thus, even when

performing extensions an event at a time (using a smart combination of some of the most efficient techniques that have been used in the task of itemset and sequence mining) one can obtain an algorithm that efficiently handles the explosive nature of pattern search, inherent to the biological sequence datasets.

References

1. J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the 8th SIGKDD International Conference on KDD and Data Mining, 2002.*
2. IBM Bioinformatics. Teiresias. http://www.research.ibm.com/bioinformatics/.
3. James Cuff and Geoffrey J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. In *PROTEINS: Structure, Function, and Genetics*, number 34. WILEY-LISS, INC, 1999.
4. Fimi. Fimi workshop 2003 (mushroom dataset). http://fimi.cs.helsinki.fi/fimi03.
5. GenBank. yeast (saccharomyces cerevisiae). www.maths.uq.edu.au.
6. A.Floratos I. Rigoutsos. Combinatorial pattern discovery in biological sequences: the teiresias algorithm. *Bioinformatics*, 1(14), January 1998.
7. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining sequential patterns efficiently by prefix projected pattern growth. In *Proceedings of the International Conference on Data Engineering, ICDE 2001.*
8. Psort. Psort database. http://www.psort.org/.
9. Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings 5th International Conference on Extending DataBase Technology*, 1996.
10. Mohammed J. Zaki. Sequence mining in categorical domains: Incorporating constraints. In *In Proceedings of 9th International Conference on Information and Knowledge Management, CIKM 2000.*
11. Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60, 2001.

An Adaptive Nearest Neighbor Classification Algorithm for Data Streams

Yan-Nei Law and Carlo Zaniolo

Computer Science Dept., UCLA, Los Angeles, CA 90095, USA
{ynlaw, zaniolo}@cs.ucla.edu

Abstract. In this paper, we propose an incremental classification algorithm which uses a multi-resolution data representation to find adaptive nearest neighbors of a test point. The algorithm achieves excellent performance by using small classifier ensembles where approximation error bounds are guaranteed for each ensemble size. The very low update cost of our incremental classifier makes it highly suitable for data stream applications. Tests performed on both synthetic and real-life data indicate that our new classifier outperforms existing algorithms for data streams in terms of accuracy and computational costs.

1 Introduction

A significant amount of recent research has focused on mining data streams for applications such as financial data analysis, network monitoring, security, sensor networks, and many others [3,8]. Algorithms for mining data streams have to address challenges not encountered in traditional mining of stored data: at the physical level, these include fast input rates and unending data sets, while, at the logical level, there is the need to cope with concept drift [18]. Therefore, classical classification algorithms must be replaced by, or modified into, incremental algorithms that are fast and light and gracefully adapt to changes in data statistics [17,18,5].

Related Works. Because of their good performance and intuitive appeal, decision tree classifiers and nearest neighborhood classifiers have been widely used in traditional data mining tasks [9]. For data streams, several decision tree classifiers have been proposed—either as single decision trees, or as ensembles of such trees. In particular, VFDT [7] and CVFDT [10] represent well-known algorithms for building single decision tree classifiers, respectively, on stationary, and time-changing data streams. These algorithms employ a criterion based on Hoeffding bounds to decide when a further level of the current decision tree should be created. While this approach assures interesting theoretical properties, the time required for updating the decision tree can be significant, and a large amount of samples is needed to build a classifier with reasonable accuracy. When the size of the training set is small, the performance of this approach can be unsatisfactory.

Another approach to data stream classification uses ensemble methods. These construct a set of classifiers by a base learner, and then combine the predictions

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 108–120, 2005.

of these base models by voting techniques. Previous research works [17,18,5] have shown that ensembles can often outperform single classifiers and are also suitable for coping with concept drift. On the other hand, ensemble methods suffer from the drawback that they often fail to provide a simple model and understanding of the problem at hand [9].

In this paper, we focus on building nearest neighbor (NN) classifiers for data streams. This technique works well in traditional data mining applications, is supported by a strong intuitive appeal, and it rather simple to implement. However, the time spent for finding the exact NN can be expensive and, therefore, a significant amount of previous research has focused on this problem. A well-known method for accelerating the nearest neighbor lookup is to use k-d trees [4]. A k-d tree is a balanced binary tree that recursively splits a d-dimensional space into smaller subregions. However, the tree can become seriously unbalanced by massive new arrivals in the data stream, and thus lose the ability of expediting the search. Another approach to NN classifiers attempts to provide approximate answers with error bound guarantees. There are many novel algorithms [11,12,13,14] for finding approximate K-NN on stored data. However, to find the $(1+\epsilon)$-approximate nearest neighbors, these algorithms must perform multiple scans of the data. Also, the update cost of the dynamic algorithms [11,13,14] depends on the size of the data set, since the entire data set is needed for the update process. Therefore, they are not suitable for mining data streams.

Our ANNCAD Algorithm. In this paper, we introduce an Adaptive NN Classification Algorithm for Data-streams. It is well-known that when data is non-uniform, it is difficult to predetermine K in the KNN classification [6,20]. So, instead of fixing a specific number of neighbors, as in the usual KNN algorithm, we adaptively expand the nearby area of a test point until a satisfactory classification is obtained. To save the computation time for finding adaptive NN, we first preassigning a class to every subregion (cell). To achieve this, we decompose the feature space of a training set and obtain a multi-resolution data representation. There are many decomposition techniques for multi-resolution data representations. The averaging technique used in this paper can be thought of Haar Wavelets Transformation [16]. Thus, information from different resolution levels can then be used for adaptively preassigning a class to every cell. Then we determine to which cell the test point belongs, in order to predict its class. Moreover, because of the compact support property inherited from wavelets, the time spent updating a classifier when a new tuple arrives is a small constant, and it is independent of the size of the data set. Unlike VDFT, which requires a large data set to decide whether to expand the tree by one more level, ANNCAD does not have this restriction.

In the paper, we use grid-based approach for classification. The main characteristic of this approach is the fast processing time and small memory usage, which is independent of the number of data points. It only depends on the number of cells of each dimension in the discretized space, which is easy to adjust in order to fulfill system constraints. Therefore, this approach has been widely employed in clustering problem. Some examples of novel clustering algorithms

are [19], [1] and [15]. However, there is not much work using this approach for classification.

Paper Organization. In this paper, we present our algorithm ANNCAD and discuss its properties in §2. In §3, we compare ANNCAD with some existing algorithms. The results suggest that ANNCAD will outperform existing algorithms. Finally, conclusions and suggestions for future work will be given in §4.

2 ANNCAD

In this section, we introduce our proposed algorithm ANNCAD, which includes four main stages: (1) Quantization of the Feature Space; (2) Building classifiers; (3) Finding predictive label for a test point by adaptively finding its neighboring cells; (4) Updating classifiers for newly arriving tuples. This algorithm only read each data tuple at most once, and only requires a small constant time to process it. We then discuss its properties and complexity.

2.1 Notation

We are given a set of d-dimensional data D with attributes $X_1, X_2, ..., X_d$. For each $i = 1, ..., d$, the domain of X_i is bounded and totally ordered, and ranges over the interval $[L_i, H_i)$. Thus, $X = [L_1, H_1) \times ... \times [L_d, H_d)$ is the feature space containing our data set D.

Definition 1. g g^d $\Delta x_i = (H_i - L_i)/g$ i^{th}

Let $B_{i_1,...,i}$ denote the block:

$$[L_1 + (i_1 - 1)\Delta x_1, L_1 + i_1 \Delta x_1) \times ... \times [L_d + (i_d - 1)\Delta x_d, L_d + i_d \Delta x_d).$$

Alternatively, we denote $B_{i_1,...,i}$ by $B_{\mathbf{i}}$, with $\mathbf{i} = (i_1, ..., i_d)$ the unique identifier for the block. Then, two blocks $B_{\mathbf{k}}$ and $B_{\mathbf{h}}$, $\mathbf{k} \neq \mathbf{h}$, are said to be if $|k_i - h_i| \leq 1$, for each $i = 1, ..., d$. In this case, $B_{\mathbf{k}}$ is said to be a neighbor of $B_{\mathbf{h}}$. $Ctr_{B_{\mathbf{i}}}$ denotes the center of block $B_{\mathbf{i}}$, computed as the average of its vertices:

$$Ctr_{B_{\mathbf{i}}} = (L_1 + (i_1 - 1/2)\Delta x_1, ..., L_d + (i_d - 1/2)\Delta x_d).$$

Definition 2. x $B_{\mathbf{i}}$ x $B_{\mathbf{i}}$ x $Ctr_{B_{\mathbf{i}}}$

Note that the distance in Def. 2 can be any kind of distance. In the following, we use Euclidean distance to be the distance between a point and a block.

2.2 Quantization of the Feature Space

The first step of ANNCAD is to partition the feature space into a discretized space with g^d blocks as in Def. 1. It is advisable to choose different sizes of grid according to system resource constraints and desirable fineness of a classifier. For each nonempty block, we count the number of training points contained in it for each class. Now we get the distribution of the data entities in each class. To decide whether we need to start with a finer resolution feature space, we then count the number of training points that do not belong to the majority class of its block as a measure of the training error. We then calculate the coarser representations of the data by averaging the 2^d corresponding blocks in the next finer level. We illustrate the above process by Example 1.

A set of 100 two-class training points in the 2-D unit square is shown in Fig. 1(a). There are two classes for this data set, where a circle (resp. triangle) represents a training point of class I (resp. II). First we separate the training points of each class, discretize them using a 4×4 grid and count the number of training points for each block to get the data distribution of each class (see Fig. 1(b)). Moreover, Fig. 1(c)-(d) show the coarser representations of the data.

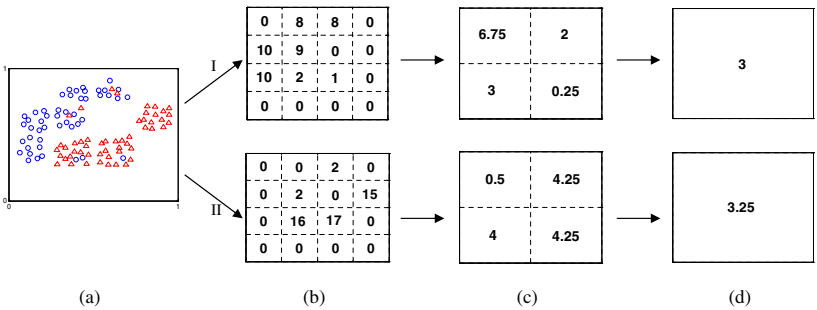


Fig. 1. Multi-resolution representation of a two-class data set

Due to the problem of the curse of dimensionality, the storage amount is exponential in the number of dimensions. To deal with this, we store the nonempty blocks in the leaf nodes of a B^+-tree using their z-values [21] as keys. Thus the required storage space is much smaller and is bounded by $O(\min(N, g^d))$ where N is the number of training samples. For instance, in Fig. 1, we only need to store information for at most 8 blocks even though there are 100 training points in the 4×4 blocks feature space. To reduce space usage, we may only store the data array of the finest level and calculate the coarser levels on the fly when building a classifier. On the other hand, to reduce time complexity, we may precalculate and store the coarser levels. In the following discussion, we assume that the system stores the data representation of each level.

2.3 Building a Classifier and Classifying Test Points

The main idea of ANNCAD is to use a multi-resolution data representation for classification. Notice that the neighborhood relation strongly depends on the quantization process. This will be addressed in next subsection by building several classifier ensembles using different grids obtained by subgrid displacements. Observe that in general, the finer level the block can be classified, the shorter distance between this block and the training set. Therefore, to build a classifier and classify a test point (see Algorithms 1 and 2), we start with the finest resolution for searching nearest neighbors and progressively consider the coarser resolutions, in order to find nearest neighbors adaptively.

We first construct a single classifier as a starting point (see Algorithm 1). We start with setting every block to have a default tag U (Non-visited). In the finest level, we classify any nonempty block with its majority class label. We then classify any nonempty block of every lower level as follows: We label the block by its majority class label if the majority class label has more points than the second majority class by a threshold percentage. If not, we use a specific tag M (Mixed) to label it.

Algorithm 1. BuildClassifier($\{\mathbf{x}, y\}|\mathbf{x}$ is a vector of attributes, y is a class label.)

Quantize the feature space containing $\{\mathbf{x}\}$
Label majority class for each nonempty block in the finest level
For each level $i = \log(g)$ downto 1
  For each nonempty block B
    If $|$majority $c_a| - |$2nd majority $c_b| >$ threshold %, label class c_a
    else label tag M
Return Classifier

Algorithm 2. TestClass(test point: $\mathbf{t}$)

For each level $i = \log(g) + 1$ downto 1
  If label of $B^i(\mathbf{t}) <> U$ /*$B^i(\mathbf{t})$ is nonempty */
    If label of $B^i(\mathbf{t}) <> M$, class of $\mathbf{t}$ = class of $B^i(\mathbf{t})$
    else class of $\mathbf{t}$ = class of NN of $B^{i+1}(\mathbf{t})$ /*$B^{i+1}(\mathbf{t})$ contains $\mathbf{t}$ in level $i+1$*/
    Break
Return class label for $\mathbf{t}$, $B^i(\mathbf{t})$

We build a classifier for the data set of Example 1 and set the threshold value to be 80%. Fig. 2(a), (b) and (c) show the class label of each nonempty block in the finest, intermediate and coarsest resolution respectively.

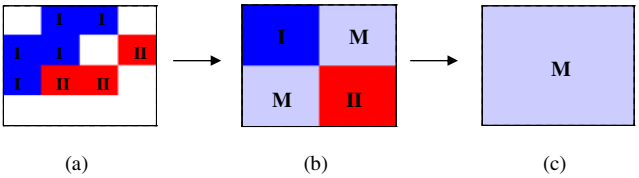


Fig. 2. Hierarchical structure of classifiers

For each level i, a test point $\mathbf{t}$ belongs to a unique block $B^i(\mathbf{t})$. We search from the finest to the coarsest level until reaching a nonempty block $B^i(\mathbf{t})$. If the label of $B^i(\mathbf{t})$ is one of the classes, we label the test point by this class. Otherwise, if $B^i(\mathbf{t})$ has tag M, we find the nearest neighbor block of $B^{i+1}(\mathbf{t})$ where $B^{i+1}(\mathbf{t})$ is a block containing $\mathbf{t}$ in level $i+1$. To reduce the time spent, we only consider the neighbors of $B^{i+1}(\mathbf{t})$ which belong to $B^i(\mathbf{t})$ in level i. It is very easy to access these neighbors as they are also neighbors of $B^{i+1}(\mathbf{t})$ in the B^+-tree with their z-values as keys. Note that $B^{i+1}(\mathbf{t})$ must be empty, otherwise we should classify it at level $i+1$. But some of the neighbors of $B^{i+1}(\mathbf{t})$ must be nonempty as $B^i(\mathbf{t})$ is nonempty. We simply calculate the distance between test point $\mathbf{t}$ and each neighbor of $B^{i+1}(\mathbf{t})$ and label $\mathbf{t}$ by the class of NN.

We use the classifier built in Example 2 to classify a test point $\mathbf{t} = (0.6, 0.7)$. Starting with the finest level, we found that the first nonempty block containing $\mathbf{t}$ is $[0.5, 1) \times [0.5, 1)$ (see Fig. 3(b)). Since it has tag M, we calculate the distance between $\mathbf{t}$ and each nonempty neighboring block in the next finer level ($[0.75, 1) \times [0.5, 0.75), [0.5, 0.75) \times [0.75, 1)$). Finally, we get the nearest neighboring block $[0.75, 1) \times [0.5, 0.75)$ and label $\mathbf{t}$ to be class I (see Fig.

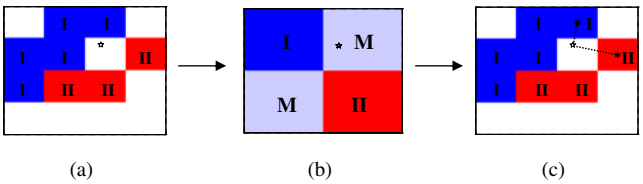


Fig. 3. Hierarchical classifier access

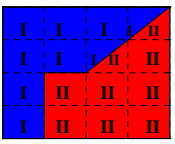


Fig. 4. The combined classifier

3(c)). When we combine the multi-resolution classifier of each level, we get a classifier for the whole feature space (see Fig. 4).

2.4 Incremental Updates of Classifiers

The main requirement of a data stream classification algorithm is that it is able to update classifiers incrementally and effectively when a new tuple arrives. Moreover, updated classifier should be adapt to concept drift behaviors. In this subsection, we present incremental update process of ANNCAD for a stationary data, without re-scanning the data and discuss an exponential forgetting technique to adapt to concept drifts.

Because of the compact support property, arrival of a new tuple only affects the blocks of the classifier in each level containing this tuple. Therefore, we only need to update the data array of these blocks and their classes if necessary. During the update process, the system may run out of memory as the number of nonempty blocks may increase. To deal with this, we may simply remove the finest data array, multiple the entries of the remaining coarser data arrays by 2^d, and update the quantity g. A detailed description of updating classifiers can be found in Algorithm 3. This solution can effectively meet the memory constraint.

Algorithm 3. UpdateClassifier(new tuple: $\mathbf{t}$)

For each level $i = \log(g) + 1$ downto 1
  Add $\delta_t / 2^{d \times (\log(g)+1-i)}$ to data array Φ^i
    /*δ_t is a matrix with value 1 in the corr. entry of t and 0 elsewhere.*/
  If i is the finest level, label $B^i(\mathbf{t})$ with the majority class
  else if |majority c_a| − |2nd majority c_b| > threshold %, label $B^i(\mathbf{t})$ by c_a
    else label $B^i(\mathbf{t})$ by tag M
If memory runs out,
  Remove the data array of level $\log(g) + 1$
  For each level $i = \log(g)$ downto 1, $\Phi^i = 2^d \cdot \Phi^i$
  Label each nonempty block of the classifier in level $\log(g)$ by its majority class
  Set $g = g/2$
Return updated classifier

Exponential Forgetting. If the concept of the data changes over time, a very common technique called exponential forgetting may be used to assign less weight to the old data to adapt to more recent trend. To achieve this, we multiply an exponential forgetting factor λ to the data array, where $0 \leq \lambda \leq 1$. For each level i, after each time interval t, we update the data array Φ^i to be:

$$\Phi^i|_{(n+1)t} \leftarrow \lambda \Phi^i|_{n \cdot t}$$

where $\Phi^i|_{n \cdot t}$ is the data array at time $n \cdot t$. Indeed, if there is no concept change, the result of classifier will not be affected. If there is a concept drift, the classifier

can adapt to the change quickly since the weight of the old data is exponentially decreased. In practice, an exponential forgetting technique is easier to implement than a sliding window because we need extra memory buffer to store the data of the most current window for implementing the sliding window.

2.5 Building Several Classifiers Using Different Grids

As mentioned above, the neighborhood relation strongly depends on the quantization process. For instance, consider the case that there is a training point u which is close to the test point v but they are located in different blocks. Then the information on u may not affect the classification of v.

To overcome the problem of initial quantization process, we build several classifier ensembles starting with different quantization space. In general, to build n^d different classifiers, each time we shift $\frac{1}{n}$ of the unit length of feature space for a set of selected dimensions. Fig. 5 shows a reference grid and its 3 different shifted grids for a feature space with 4×4 blocks. For a given test point $\mathbf{t}$, we use these n^d classifiers to get n^d class labels and selected blocks $B^i(\mathbf{t})$ of $\mathbf{t}$ in each level i, starting from the finest one. We then choose the majority class label. If there is tie, we calculate the distance between each selected block $B^i(\mathbf{t})$ with majority class label and $\mathbf{t}$ to find the closest one. Algorithm 4 shows this classifying process using n^d classifiers.

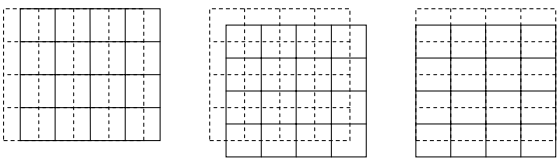


Fig. 5. An example of 4 different grids for building 4 classifiers

---

Algorithm 4. Testn^dClass(objects: $\mathbf{t}$)

For each level $i = \log(g) + 1$ downto 1
  Get the label of $\mathbf{t}$ for each classifier
  If there is a label <> U, choose the majority label
    If there is a tie, label $\mathbf{t}$ by class of $B^i(\mathbf{t})$ with closest center to $\mathbf{t}$
    Break
Return class label for $\mathbf{t}$

The following theorem shows that the approximation error of finding nearest neighbors decreases as the number of classifier ensembles increases.

Theorem 1. [illegible] x [illegible] Y [illegible] x [illegible] n^d [illegible] $z \notin Y$ [illegible] $(x, y) < (1 + \frac{1}{n-1}) * (x, z)$ [illegible] $y \in Y$

For simplicity, we consider the case when $d = 1$. This proof works for any d. For $d = 1$, we build n classifiers, where each classifier i use the grid that is shifted $\frac{i}{n}$ unit length from the original grid. Let ϵ be the length of a block. Consider a test point x, x belongs to an interval I_k for classifier k. Note that $[x - \frac{n-1}{n}\epsilon, x + \frac{n-1}{n}\epsilon] \subset \bigcup I_k \subset [x - \epsilon, x + \epsilon]$. Hence, the distance between x and its nearest neighbor that we found must be less than ϵ. Meanwhile, the points that we do not consider should be at least $\frac{n-1}{n}\epsilon$ far away from x. If $z \notin Y$, $\frac{dist(x,y)}{dist(x,z)} < \frac{\epsilon}{(n-1)\epsilon/n} = (1 + \frac{1}{n-1})$ for every $y \in Y$.

The above theorem shows that the classification result using one classifier does not have any guarantee about the quality of the nearest neighbors that it found because the ratio of approximation error will tend to infinity. When n is large enough, the set of training points selected by those classifier ensembles are exactly the set of training points with distance ϵ from the test point. To achieve an approximation error bound guarantee, theoretically we need an exponential number of classifiers. However, in practice, we only use two classifiers to get a good result. Indeed, experiments in §3 show that few classifiers can obtain a significant improvement at the beginning. After this stage, the performance will become steady even though we keep increasing the number of classifiers.

2.6 Properties of ANNCAD

As ANNCAD is a combination of multi-resolution and adaptive nearest neighbors techniques, it inherits both their properties and their advantages.

- The locality property allows a fast update. As a new tuple arrival only affects the class of the block containing it in each level, the incremental update process only costs a constant time (number of levels).
- We may set a threshold value for classifying decisions to remove noise.
- This algorithm makes it easy to build multi-resolution classifiers. Users can specify the number of levels to efficiently control the fineness of the classifier. Moreover, one may optimize the system resource constraints and easy to adjust on the fly when the system runs out of memory.
- Let g, N and d be the number of blocks of each dimension, training points and attributes respectively. The time spent on building a classifier is $O(\min(N, g^d))$ with constant factor $\log(g)$. For the time spent on classifying a test point, the worst case complexity is $O(\log_2(g) + 2^d)$ where the first part is for classifying a test point using classifiers and the second part is for finding its nearest neighbor which is optional. Also, the time spent for updating classifiers when a new tuple arrives is $\log_2(g) + 1$. Comparing with the time spent in VFDT, our method is more attractive.

3 Performance Evaluation

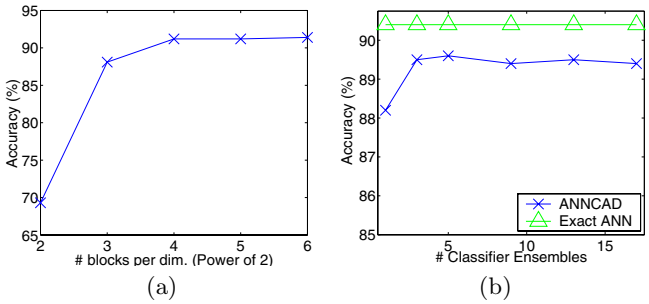
In this section, we first study the effects on parameters for ANNCAD by using two synthetic data sets. We then compare ANNCAD with VFDT and CVFDT

on three real-life data sets. To illustrate the approximation power of ANNCAD, we include the results of , which computes ANN exactly, as controls. : For each test point t, we search the area within 0.5 block side length distance. If the area is nonempty, we classify t as the majority label of all these points in this area. Otherwise, we expand the searching area by doubling the radius until we get a class for t. Note that the time and space complexities of are very expensive making it impractical to use.

3.1 Synthetic Data Sets

The aim of this experiment is to study the effect on the initial resolution for ANNCAD. In this synthetic data set, we consider a 3-D unit cube. We randomly pick 3k training points and assign those points which are inside a sphere with center (0.5, 0.5, 0.5) and radius 0.5 to be class 0, and class 1 otherwise. This data set is effective to test the performance of a classifier as it has a curve-like decision boundary. We then randomly draw 1k test points and run ANNCAD starting with different initial resolution and 100% threshold value. In Fig. 6(a), the result shows that a finer initial resolution gets a better result. This can be explained by the fact that we can capture a curve-like decision boundary if we start with a finer resolution. On the other hand, as discussed in last section, the time spent for building a classifier increases linearly for different resolutions. In general, we should choose a resolution according to system resource constraints.

The aim of this experiment is to study the effect on number of classifier ensembles for ANNCAD. As in the previous experiment, we randomly pick 1k training examples and assign them labels. We then randomly draw 1k test points and test them based on the voting result of these classifiers. We set $16 \times 16 \times 16$ blocks for the finest level and 100% threshold value. In Fig. 6(b), the result shows that having more classifiers will get a better result in the beginning. The performance improvement becomes steady even though we keep increasing the number of classifiers. It is because there is no further information given when we increase the number of classifiers. In this experiment, we only use 2 or 3 classifiers to obtain a competitive result with the (90.4%).



(a) (b)

Fig. 6. Effect on initial resolutions and number of classifiers

3.2 Real Life Data Sets

The aim of this set of experiments is to compare the performance of ANNCAD with that of VFDT and CFVDT on stationary and time-changing real-life data sets respectively. We first used a letter recognition data set from the UCI machine learning repository web site [2]. The objective is to identify a black-and-white pixel displays as one of the 26 English alphabet. In this data set, each entity is a pixel display for an English alphabet and has 16 numerical attributes to describe its pixel displays. The detail description of this data set is provided in [2]. In this experiment, we use 15k tuples for training set with 5% noise added and 5k for test set. We obtain noisy data by randomly assigning a class label for 5% training examples. For ANNCAD, we set g for the initial grid to be 16 units and build two classifiers. Moreover, since VFDT needs a very large training set to get a fair result, we rescan the data sets up to 500 times for VFDT. So the data set becomes 7,500,000 tuples. In Fig. 7(a), the performance of ANNCAD dominates that of VFDT. Moreover, ANNCAD only needs one scan to achieve this result, which shows that ANNCAD even works well for a small training set.

The second real life data set we used is the Forest Cover Type data set which is another data set from [2]. The objective is to predict forest cover type (7 types). For each observation, there are 54 variables. Neural network (backpropagation) was employed to classify this data set and got 70% accuracy, which is the highest one recorded in [2]. In our experiment, we used all the 10 quantitative variables. There are 12k examples for training set and 90k examples for testing set. For ANNCAD, we scaled each attribute to the range $[0, 1)$. We set g for the initial grid to be 32 units and build two classifiers. As the above experiment, we rescan the training set up to 120 times for VFDT, until its performance becomes steady. In Fig. 7(b), the performance of ANNCAD dominates that of VFDT. These two experiments show that ANNCAD works well in different kinds of data sets.

We further tested ANNCAD in the case when there are concept drifts in data set. The data we used was extracted from the census bureau database [2]. Each observation represents a record of an adult and has 14 attributes including age, race etc. The prediction task is to determine whether a person makes over 50K a year. Concept drift is simulated by grouping records with same race (Amer-Indian-Eskimo(AIE), Asian-Pac-Islander(API), Black(B), Other(O), White(W)). The distribution of training tuples of each race is shown in Fig. 7(c). Since the models for different races of people should be different, concept drifts are introduced when $n = 311, 1350, 4474, 4746$. In this experiment, we used the 6 continuous attributes. We used 7800 examples for learning and tested the classifiers for every 300 examples. For ANNCAD, we build two classifiers and set λ to be 0.98 and g for the initial grid to be 64 units. We scaled the attribute values as mentioned in the previous experiment. The results are shown in Fig. 7(c). The curves show that ANNCAD keeps improving in each region. Also, as mentioned in §2.6, computations required for ANNCAD are much lower than CVFDT.

Moreover, notice that ANNCAD works almost as well as [illegible] on these three data sets, which demonstrates its excellent approximation ability.

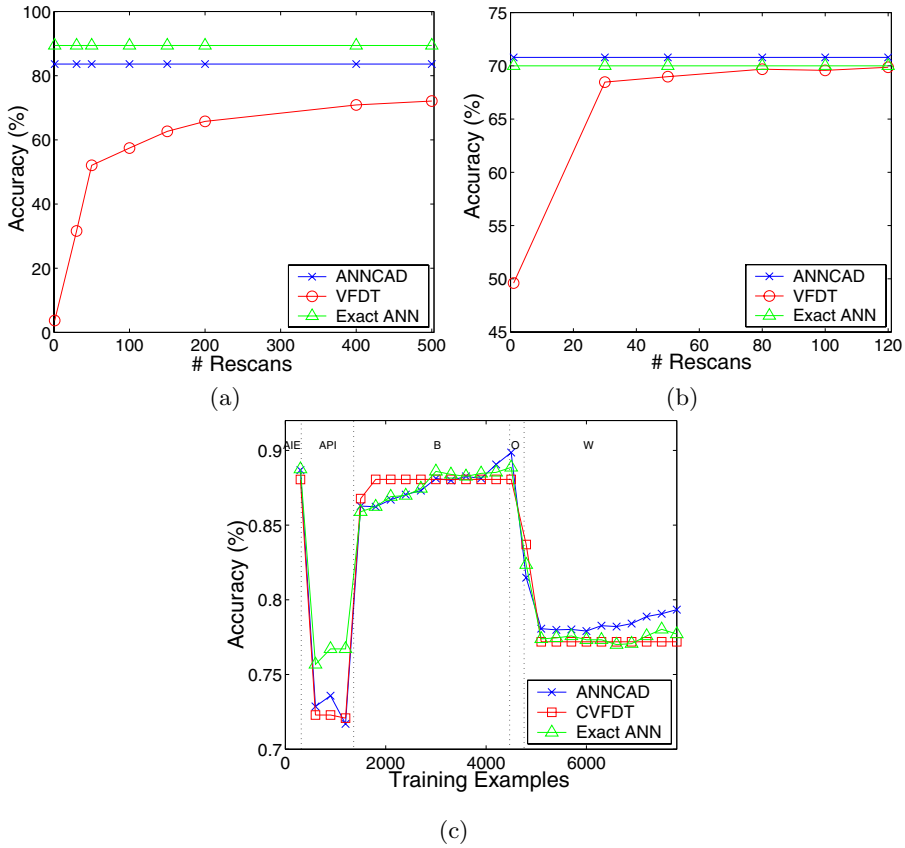


Fig. 7. Three real-life data sets:(a) Letter Recognition (b) Forest Covertype (c) Census

4 Conclusion and Future Work

In this paper, we proposed an incremental classification algorithm ANNCAD using a multi-resolution data representation to find adaptive nearest neighbors of a test point. ANNCAD is very suitable for mining data streams as its update speed is very fast. Also, the accuracy compares favorably with existing algorithms for mining data streams. ANNCAD adapts to concept drift effectively by the exponential forgetting approach. However, the very detection of sudden concept drift is of interest in many applications. The ANNCAD framework can also be extended to detect concept drift–e.g. changes in class label of blocks is a good indicator of possible concept drift. This represents a topic for our future research.

Acknowledgement. This research was supported in part by NSF Grant No. 0326214.

References

1. R. Agrawal, J. Gehrke, D. Gunopulos and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *SIGMOD 1998: 94–105.*
2. C. L. Blake and C. J. Merz. UCI Repository of machine learning databases. `http://www.ics.uci.edu/∼mlearn/MLRepository.html`.
3. B. Babcock, S. Babu, R. Motawani and J. Widom. Models and issues in data stream systems. *PODS 2002: 1–16.*
4. J. Bentley. Multidimensional binary search trees used for associative searching. *Communication of the ACM 18(9): 509–517 (1975).*
5. F. Chu and C. Zaniolo. Fast and light boosting for adaptive mining of data streams. *PAKDD 2004: 282–292.*
6. C. Domeniconi, J. Peng and D. Gunopulos, Locally adaptive metric nearest-neighbor classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence 24(9): 1281–1285 (2002).*
7. P. Domingos and G. Hulten. Mining high-speed data streams. *KDD 2000: 71-80.*
8. L. Golab and M. Özsu. Issues in data stream management. *ACM SIGMOD 32(2): 5–14 (2003).*
9. J. Han and M. Kamber. *Data Mining – Concepts and Techniques (2000).* Morgan Kaufmann Publishers.
10. G. Hulten, L. Spence and P. Domingos. Mining time-changing data streams. *KDD 2001: 97–106.*
11. P. Indyk, R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. *STOC 1998: 604–613.*
12. P. Indyk. Dimensionality reduction techniques for proximity problems. *ACM-SIAM symposium on Discrete algorithms 2000: 371–378.*
13. P. Indyk. High-dimensional computational geometry. *Dept. of Comput. Sci., Stanford Univ., 2001.*
14. E. Kushilevitz, R. Ostrovsky, Y. Rabani. Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces. *SIAM J. Comput. 30(2): 457–474 (2000).*
15. G. Sheikholeslami, S. Chatterjee and A. Zhang. WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases. *VLDB 1998: 428–439.*
16. G. Strang and T. Nguyen. *Wavelets and Filter Banks (1996).* Wellesley-Cambridge Press.
17. W. Street and Y. Kim. A streaming ensemble algorithm (SEA) for large-scale classification. *SIGKDD 2001: 377–382.*
18. H. Wang, W. Fan, P. Yu and J. Han. Mining concept-drifting data streams using ensemble classifiers. *SIGKDD 2003: 226–235.*
19. W. Wang, J. Yang and R. Muntz. STING: A Statistical Information Grid Approach to Spatial Data Mining *VLDB 1997: 186–195.*
20. D. Wettschereck and T. Dietterich. Locally Adaptive Nearest Neighbor Algorithms. *Advances in Neural Information Processing Systems 6: 184–191 (1994).*
21. C. Zaniolo, S. Ceri, C. Faloutsos, R. Snodgrass, V. Subrahmanian and R. Zicari. *Advanced Database Systems (1997).* Morgan Kaufmann Press.

Support Vector Random Fields for Spatial Classification

Chi-Hoon Lee, Russell Greiner, and Mark Schmidt

Department of Computing Science,
University of Alberta,
Edmonton AB, Canada
{chihoon, greiner, schmidt}@cs.ualberta.ca

Abstract. In this paper we propose Support Vector Random Fields (SVRFs), an extension of Support Vector Machines (SVMs) that explicitly models spatial correlations in multi-dimensional data. SVRFs are derived as Conditional Random Fields that take advantage of the generalization properties of SVMs. We also propose improvements to computing posterior probability distributions from SVMs, and present a local-consistency potential measure that encourages spatial continuity. SVRFs can be efficiently trained, converge quickly during inference, and can be trivially augmented with kernel functions. SVRFs are more robust to class imbalance than Discriminative Random Fields (DRFs), and are more accurate near edges. Our results on synthetic data and a real-world tumor detection task show the superiority of SVRFs over both SVMs and DRFs.

1 Introduction

The task of classification has traditionally focused on data that is "independent and identically distributed" (iid), in particular assuming that the class labels for different data points are conditionally independent (ie. knowing that one patient has cancer does not mean another one will). However, real-world classification problems often deal with data points whose labels are correlated, and thus the data violates the iid assumption. There is extensive literature focusing on the 1-dimensional 'sequential' case (see [1]), where correlations in the labels of data points in a linear sequence exist, such as in strings, sequences, and language. This paper focuses on the more general 'spatial' case, where these correlations exist in data with two-dimensional (or higher-dimensional) structure, such as in images, volumes, graphs, and video.

Classifiers that make the iid assumption often produce undesirable results when applied to data with spatial dependencies in the labels. For example, in the task of image labeling, a classifier could classify a pixel as 'face', even if all adjacent pixels were classified as 'non-face'. This problem motivates the use of Markov Random Fields (MRFs) and more recently Conditional Random Fields (CRFs) for spatial data. These classification techniques augment the performance of an iid classification technique (often a Mixture Model for MRFs, and Logistic Regression for CRFs) by taking into account spatial class dependencies.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 121–132, 2005.

Support Vector Machines (SVMs) are classifiers that have appealing theoretical properties [2], and have shown impressive empirical results in a wide variety of tasks. However, this technique makes the critical iid assumption. This paper proposed an extension to SVMs that considers spatial correlations among data instances (as in Random Field models), while still taking advantage of the powerful discriminative properties of SVMs. We refer to this technique as Support Vector Random Fields (SVRFs)

The remaining sections of this paper are organized as follows. Section 2 formalizes the task and reviews related methods for modeling dependencies in the labels of spatial data. Section 3 reviews Support Vector Machines, and presents our Support Vector Random Field extension. Experimental results on synthetic and real data sets are given in Sect. 4, while a summary of our contribution is presented in Sect. 5.

2 Related Work

The challenge of performing classification while modeling class dependencies is often divided into two perspectives: Generative and Discriminative models [1]. Generative classifiers learn a model of the joint probability, $p(x, y) = p(x|y)p(y)$, of the features x and corresponding labels y. Predictions are made using Bayes rule to compute $p(y|x)$, and finding an assignment of labels maximizing this probability. In contrast, discriminative classifiers model the posterior $p(y|x)$ directly without generating any prior distributions over the classes. Thus, discriminative models solely focus on maximizing the conditional probability of the labels, given the features. For many applications, discriminative classifiers often achieve higher accuracy than generative classifiers [1]. There has been much related work on using random field theory to model class dependencies in generative and more recently discriminative contexts [3,4]. Hence, we will first review [illegible] (typically formulated as a generative classifier), followed by [illegible] (a state-of-the-art discriminative classifier built upon the foundations of Markov Random Fields).

2.1 Problem Formulation

In this work, we will focus on the task of classifying elements (pixels or regions) of a two-dimensional image, although the methods discussed also apply to higher-dimensional data. An image is represented with an M by N matrix of elements. For an instance $X = (x_{11}, x_{12}, \ldots, x_{1N}, \ldots, x_{M1}, x_{M2}, \ldots, x_{MN})$, we seek to infer the most likely joint class labels:

$$Y^* = (y^*_{11}, y^*_{12}, \ldots, y^*_{1N}, \ldots, y^*_{M1}, y^*_{M2}, \ldots, y^*_{MN})$$

If we assume that the labels assigned to elements are independent, the following joint probability can be formulated: $P(Y) = \prod_{i=1}^{M} \prod_{j=1}^{N} P(y_{ij})$. However, conditional independency does not hold for image data, since spatially adjacent elements are likely to receive the same labels. We therefore need to explicitly

consider this local dependency. This involves addressing three important issues: How should the optimal solution be defined, how are spatial dependencies considered, and how should we search the (exponential size) configuration space.

2.2 Markov Random Fields (MRFs)

(MRFs) provide a mathematical formulation for modeling local dependencies, and are defined as follows [3]:

Definition 1. $S - \{i\}$ $y_{S-\{i\}}$ $S-\{i\}$ N_i i

$P(Y) > 0$
$P(y_i|y_{S-\{i\}}) = P(y_i|y_N)$

Condition 2 (Markovianity) states that the conditional distribution of an element y_i is dependent only on its neighbors. Markov Random Fields have traditionally sought to maximize the joint probability $P(Y^*)$ (a generative approach). In this formulation, the posterior over the labels given the observations is formulated using Bayes' rule as:

$$P(Y|X) \propto P(X|Y)P(Y) = P(Y)\prod_{i}^{n} P(x_i|y_i) \tag{1}$$

In (1), the equivalence between MRFs and Gibbs Distributions [5] provides an efficient way to factor the prior $P(Y)$ over cliques defined in the neighborhood Graph G. The prior $P(Y)$ is written as

$$P(Y) = \frac{\exp(\sum_{c\in C} V_c(Y))}{\sum_{Y'\in\Omega} exp(\sum_{c\in C} V_c(Y'))} \tag{2}$$

where $V_c(Y)$ is a clique potential function of labels for clique $c \in C$, C is a set of cliques in G, and Ω is the space of all possible labelings. From (1) and (2), the target configuration Y^* is a realization of a locally dependent Markov Random Field with a specified prior distribution. Based on (1) and (2) and using Z to denote the (normalizing) "partition function", if we assume Gaussian likelihoods then the posterior distribution can be factored as:

$$P(Y|X) = \frac{1}{Z}\exp\Big[\sum_{i\in S}\log(P(x_i|y_i)) + \sum_{c\in C} V_c(Y_c)\Big] \tag{3}$$

The Gaussian assumption for $P(X|Y)$ in (1) allows straightforward Maximum Likelihood parameter estimation. Although there have been many approximation

algorithms designed to find the optimal Y^*, we will focus on a local method called [illegible] [5], written as:

$$y_i^* = \arg\max_{y \in L} P(y_i | y_N \; x_i) \tag{4}$$

Assuming Gaussians for the likelihood and a pairwise neighborhood system for the prior over labels, (4) can be restated as:

$$y_i^* = \arg\max_{y \in L} \frac{1}{Z_i} \exp\Big[\log(P(x_i|y_i)) + \sum_{j \in N} \beta y_i y_j\Big] \tag{5}$$

where β is a constant and L is a set of class labels.

This concept has proved to be applicable in a wide variety of domains where there exists correlations among neighboring instances. However, the generative nature of the model and the assumption that the likelihood is Gaussian can be too restrictive to capture complex dependencies between neighboring elements or between observations and labels. In addition, the prior over labels is completely independent from the observations, thus the interactions between neighbors are not proportional to their similarity.

2.3 Conditional Random Fields (CRFs)

[illegible] avoid the Gaussian assumption by using a model that seeks to maximize the conditional probability of the labels given the observations $P(Y^*|X)$ (a discriminative model), and are defined as follows [1]:

Definition 2. [illegible] $G = (S, E)$ [illegible] Y [illegible] (X, Y) [illegible] Y [illegible] y_i [illegible] $P(y_i|X, y_{s \setminus i}) = P(y_i|X, y_N)$

This model alleviates the need to model the observations $P(X)$, allowing the use of arbitrary attributes of the observations without explicitly modeling them. CRFs assume a 1-dimensional chain-structure where only adjacent elements are neighbors. This allows the factorization of the joint probability over labels. Discriminative Random Fields (DRFs) extend 1-dimensional CRFs to 2-dimensional structures [6]. The conditional probability of the labels Y in the Discriminative Random Field framework is defined as:

$$P(Y|X) = \frac{1}{Z} \exp\Big(\sum_{i \in S} A_i(y_i, X) + \sum_{i \in S} \sum_{j \in N} I_{ij}(y_i, y_{j,X})\Big) \tag{6}$$

A_i is the 'Association' potential that models dependencies between the observations and the class labels, while I_i is the 'Interaction' potential that models dependencies between the labels of neighboring elements (and the observations). Note that this is a much more powerful model than the assumed Gaussian Association potential and the indicator function used for the Interaction potential

(that doesn't consider the observations) in MRFs. Parameter learning in DRFs involves maximizing the log likelihood of (6), while inference uses ICM [6].

DRFs are a powerful method for modeling dependencies in spatial data. However, several problems associated with this method include the fact that it is hard to find a good initial labeling and stopping criteria during inference, and it is sensitive to issues of class imbalance. Furthermore, for some real-world tasks the use of logistic regression as a discriminative method in DRFs often does not produce results that are as accurate as powerful classification models such as Support Vector Machines (that make the iid assumption).

3 Support Vector Random Fields (SVRFs)

This section presents Support Vector Random Fields (SVRFs), our extension of SVMs that allows the modelling of non-trivial two-dimensional (or higher) spatial dependencies using a CRF framework. This model has two major components: The [illegible] potential function and the [illegible] potential function. The [illegible] function captures relationships between the observations and the class labels, while the [illegible] function models relationships between the labels of neighboring data points and the observations at data points. Since the selection of the observation-matching potential is critical to the performance of the model, the Support Vector Random Field model employs SVMs for this potential, providing a theoretical and empirical advantage over the logistic model used in DRFs and the Gaussian model used in MRFs, that produce unsatisfactory results for many tasks. SVRFs can be formulated as follows:

$$P(Y|X) = \frac{1}{Z} \exp\Big\{ \sum_{i \in S} \log(O(y_i, \Upsilon_i(X))) + \sum_{i \in S} \sum_{j \in N} V(y_i, y_j, X) \Big\} \tag{7}$$

In this formulation, $\Upsilon_i(X)$ is a function that computes features from the observations X for location i, $O(y_i, \Upsilon_i(X))$ is the observation-potential, and $V(y_i, y_j, X)$ is the local-consistency potential. The pair-wise neighborhood system is defined as a local dependency structure. In this work, interactions between pixels with a Euclidean distance of 1 were considered (ie. the radius 1 von Neumann neighborhood). We will now examine these potentials in more detail.

3.1 Observation-Matching

The observation-matching potential seeks to find a posterior probability distribution that maps from the observations to corresponding class labels. DRFs employ a Generalized Linear Models (GLM) for this potential. However, GLMs often do not estimate appropriate parameters. This is especially true in image data where feature sets may have a high number of dimensions and/or several features have a high degree of correlation. This can cause problems in parameter estimation and approximations to resolve these issues may not produce optimal parameters [7].

Fortunately, the CRF framework allows a flexible choice of the observation-matching potential function. We overcome the disadvantages of the GLM by employing a Support Vector Machine classifier, seeking to find the margin maximizing hyperplane between the classes. This classifier has appealing properties in high-dimensional spaces and is less sensitive to class imbalance. Furthermore, due to the properties of error bounds, SVMs tends to outperform GLMs, especially when the classes overlap in the feature space (often the case with image data). Parameter estimation for SVMs involves optimizing the following Quadratic Programming problem for the training data x_i (where, is a constant that bounds the misclassification error):

$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i}^{N} \sum_{j}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$subject \;\; to \;\; 0 \leq \alpha_i \leq C \;\; and \;\; \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{8}$$

Consequently, the decision function, given the parameters α_i for the l training instances and bias term b, is (for a more thorough discussion of SVMs, we refer to [2]): $f(x) = \sum_{i=1}^{l} (\alpha_i y_i x \cdot x_i) + b$

Unfortunately, the decision function $f(x)$ produced by SVMs measures distances to the decision boundary, while we require a posterior probability function. We adopted the approach of [8] to convert the decision function to a posterior probability function. This approach is efficient and minimizes the risk of overfitting during the conversion, but has some ambiguities and potential difficulties in numerical computation. We have addressed these issues in our approach, which will be briefly outlined here.

We estimate a posterior probability from the Support Vector Machine decision function using the sigmoid function:

$$O(y_i = 1, \Upsilon_i(X)) = \frac{1}{1 + \exp(Af(\Upsilon_i(X)) + B)} \tag{9}$$

The parameters A and B are estimated from training data represented as pairs $(f(\Upsilon_i(X)), t_i)$, where $f(\cdot)$ is the Support Vector Machine decision function, and t_i denotes a relaxed probability that $y_i = 1$ as in (9). We could set $t_i = 1$, if the class label at i is 1(ie. $y_i = 1$). However, this ignores the possibility that $\Upsilon_i(X)$ has the opposite class label (ie. -1). Thus, we employed the relaxed probability: $t_i = \frac{N_+ + 1}{N_+ + 2}$, if $y_i = 1$, and $t_i = \frac{1}{N_- + 2}$, if $y_i = -1$ (N_+ and N_- being the number of positive and negative class instances). By producing the new forms of training instances, we can solve the following optimization problem to estimate parameters:

$$\min - \sum_{i=1}^{l} \Big[t_i \log p(\Upsilon_i(X)) + (1 - t_i) \log(1 - p(\Upsilon_i(X))) \Big] \tag{10}$$

where

$$p(\Upsilon_i(X)) = \frac{1}{1 + exp(Af(\Upsilon_i(X)) + B)}$$

[8] adopted a Levenberg-Marquardt approach to solve the optimization problem, finding an approximation of the Hessian matrix. However, this may cause incorrect computations of the Hessian matrix (especially for unconstrained optimizations [7]). Hence, we employed Newton's method with backtracking line search to solve the optimization. In addition, in order to avoid overflows and underflows of *exp* and *log* functions, we reformulate Eq.10 as follows:

$$-\Big(t_i \log p(\Upsilon_i(X)) + (1 - t_i)\log(1 - p(\Upsilon_i(X)))\Big)$$

$$= t_i(Af(\Upsilon_i(X)) + B) + \log(1 + \exp(-Af(\Upsilon_i(X)) - B)) \tag{11}$$

3.2 Local-Consistency

In MRFs, local-consistency considers correlations between neighboring data points, and is considered to be observation independent. CRFs provide more powerful modelling of local-consistency by removing the assumption of observation independence. In order to use the principles of CRFs for local-consistency, an approach is needed that penalizes discontinuity between pairwise sites. For this, we use a linear function of pairwise continuity:

$$V(y_i, y_j, X) = y_i y_j \nu^T \Phi_{ij}(X) \tag{12}$$

$\Phi_{ij}(X)$ is a function that computes features for sites i and j based on observations X. As opposed to DRFs, which penalize discontinuity by considering the absolute difference between pairwise observations [6], our approach introduces a new mapping function $\Phi(\cdot)$ that encourages continuity in addition to penalizing discontinuity (using $\max(\Upsilon(X))$ to denote the vector of max values for each feature):

$$\Phi_{ij}(X) = \frac{\max(\Upsilon(X)) - \mid \Upsilon_i(X) - \Upsilon_j(X) \mid}{\max(\Upsilon(X))} \tag{13}$$

3.3 Learning and Inference

The proposed model needs to estimate the parameters of the observation-matching function and the local-consistency function. Although we estimate these parameters sequentially, our model outperforms the simultaneous learning approach of DRFs and significantly increases its computational efficiency.

The parameters of the Support Vector Machine decision function are first estimated by solving the Quadratic Programming problem in (8) (using SVMlight [9]). We then convert the decision function to a posterior function using (10) and the new training instances. Finally, we adopted pseudolikelihood [3] to estimate the local consistency parameters ν, due to its simplicity and fast computation. For training on l pixels from K images, pseudolikehood is formulated as:

$$\widehat{\nu} = \arg\max_{\nu} \prod_{k=1}^{K} \prod_{i=1}^{l} P(y_i^k | y_N^k\ , X^k, \nu) \tag{14}$$

As in [6], to ensure that the log-likelihood is convex we assume that ν is Gaussian and compute the local-consistency parameters using its log likelihood $l(\widehat{\nu})$:

$$l(\widehat{\nu}) = \arg\max_{\nu} \sum_{k=1}^{K} \sum_{i=1}^{l} \left\{ O_i^n + \sum_{j \in N} V(y_i^k, y_j^k, X^k) - \log(z_i^k) \right\} - \frac{1}{2\tau} \nu^T \nu \quad (15)$$

In this model, z_i^k is a partition function for each site i in image k, and τ is a regularizing constant. Equation (15) is solved by gradient descent, and note that the observation matching function acts as a constant during this process. Due to the employment of SVMs, the time complexity of learning is $O(S^2)$, where S is the number of pixels to be trained, although in practice it is much faster.

The inference problem is to infer an optimal labeling Y^* given a new instance X and the estimated model parameters. We herein adopted the Iterated Conditional Modes (ICM) approach described in Section 2.2 [5], that maximizes the local conditional probability iteratively. For our proposed model and [6], ICM is expressed as,

$$y_i^* = \arg\max_{y \in L} P(y_i | y_N, X) \quad (16)$$

Although ICM is based on iterative principles, it often converges quickly to a high quality configuration, and each iteration has time complexity $O(S)$.

4 Experiments

We have evaluated our proposed model on synthetic and real-world binary image labeling tasks, comparing our approach to Logistic Regression, SVMs, and DRFs for these problems. Since class imbalance was present in many of the data sets, we used the Jaccard measure to quantify performance: $f = \frac{TP}{TP+FP+FN}$, where TP is the number of true positives, FP denotes the number of false positives, and FN tallies false negatives.

4.1 Experiments on Synthetic Data

We evaluated the four techniques over 5 synthetic binary image sets. These binary images were corrupted by zero mean Gaussian noise with unit standard deviation, and the task was to label the foreground objects (see the first and second columns in Fig. 1). Two of the sets contained balanced class labels (and), while the other three contained imbalanced classes. The five 150 image sets were divided into 100 images for training and 50 for testing. Example results and aggregate scores are shown in Fig. 1. Note that the last 4 columns illustrate the outcomes from each technique– SVMs, Logistic Regression (LR), SVRFs, and DRFs.

Logistic Regression and subsequently DRFs performed poorly in all three imbalanced data sets (, , and). In these cases, SVMs outperformed these methods and consequently our proposed SVRFs outperformed

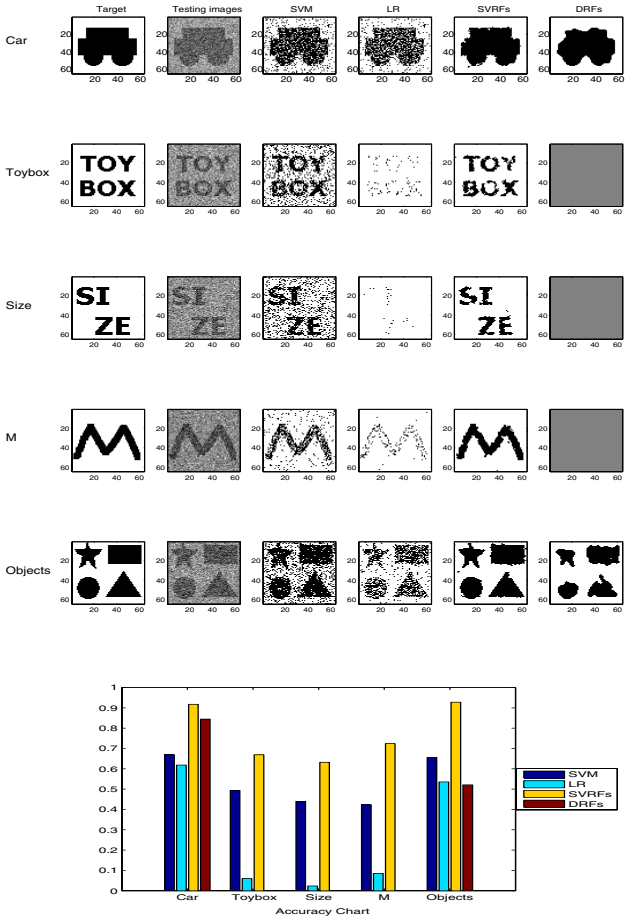
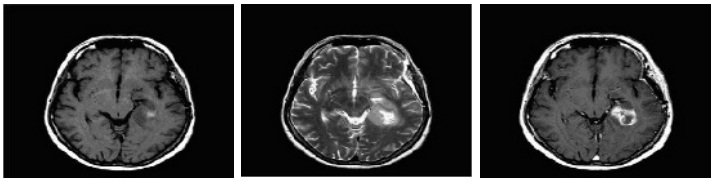


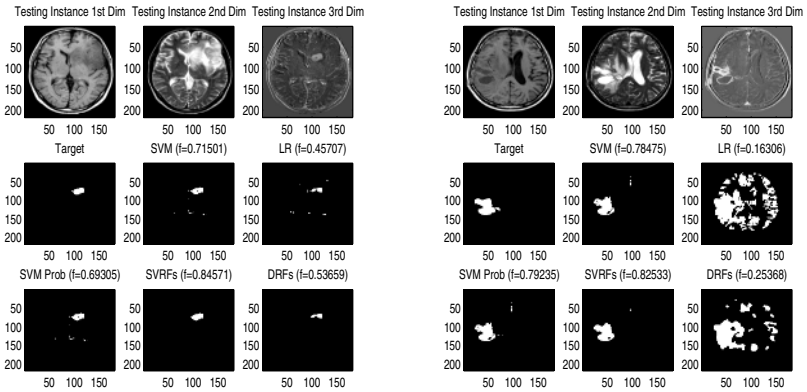
Fig. 1. Average scores on synthetic data sets

SVMs. In the first balanced data set ( ), DRFs and SVRFs both significantly outperformed SVMs and Logistic Regression (the iid classifiers). However, DRFs performed poorly on the second balanced data set ( ). This is due to DRFs simultaneous parameter learning, that tends to overestimate the local-consistency potential. Since the observation-matching is underweighted, edges become degraded during inference (there are more edge areas in the data). Terminating inference before convergence could reduce this, but this is not highly desirable for automatic classification. Overall, our Support Vector Random Field model demonstrated the best performance on all data sets, in particular those with imbalanced data and a greater proportion of edge areas.



(a) T1 (b) T2 (c) T1-Contrast

Fig. 2. A multi-spectral MRI

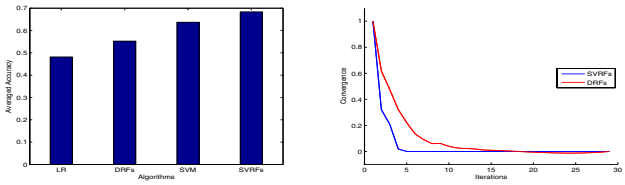


(a) Example 1 (b) Example 2

Fig. 3. An example of the classification result

4.2 Experiments on Real Data

We applied our model to the real-world problem of tumor segmentation in medical imaging. We focused on the task of brain tumor segmentation in MRI, an important task in surgical planning and radiation therapy currently being laboriously done by human medical experts. There has been significant research focusing on automating this challenging task (see [10]). Markov Random Fields have been explored previously for this task (see [10]), but recently SVMs have shown impressive performance [11,12]. This represents a scenario where our proposed Support Vector Random Field model could have a major impact. We evaluated the four classifiers from the previous section over 7 brain tumor patients. For each patient, three MRI 'modalities' were available: T1 (visualizing fat locations), T2 (visualizing water locations), and an additional T1 image with a 'contrast agent' added to enhance the visualization of metabolically active tumor areas (refer to Fig. 2).



(a) Averaged accuracy

(b) Convergency in inference

Fig. 4. Averaged accuracy and convergence in inference

The data was preprocessed with the Statistical Parametric Mapping software [13] to non-linearly align the images with a template in a standard coordinate system, and remove intensity inhomogeneity field effects. This non-linear template alignment approach was quantified to be highly effective in [14], and the inhomogeneity correction step computes a smooth corrective field that seeks to minimize the residual entropy after transformation of the log-intensity value's probability distribution [15]. We used 12 features that incorporate image information and domain knowledge (the raw intensities, spatial expected intensities within the coordinate system, spatial priors for the brain area and normal tissue types within the coordinate system, the template image information, and left-to-right symmetry), each measured as features at 3 scales by using 3 different sizes of Gaussian kernel filters. We used a 'patient-specific' training scenario similar to [11,12].

Results for two of the patients are shown in Fig. 3, while average scores over the 7 patients are shown in Fig. 4(a). Note that 'SVM+prob' in Fig. 3 denotes the classification results from the Support Vector Machine posterior probability estimate. The Logistic Regression model performs poorly at this task, but DRFs perform significantly better. As with the synthetic data in cases of class imbalance, SVMs outperform both Logistic Regression and the DRFs. Finally, SVRFs improve the scores obtained by the SVMs by almost 5% (a significant improvement).

We compared convergence of the DRFs and SVRFs by measuring how many label changes occured between inference iterations averaged over 21 trials (Fig. 4(a)). These results show that DRFs on average require almost 3 times as many iterations to converge, due to the overestimation of the local-consistency potential.

5 Conclusion

We have proposed a novel model for classification of data with spatial dependencies. The Support Vector Random Field combines ideas from SVMs and CRFs, and outperforms SVMs and DRFs on both synthetic data sets and an important real-world application. We also proposed an improvement to computing posterior

probability distributions from SVM decision functions, and a method to encourage continuity with local-consistency potentials. Our Support Vector Random Field model is robust to class imbalance, can be efficiently trained, converges quickly during inference, and can trivially be augmented with kernel functions to further improve results.

Acknowledgment

R. Greiner is supported by the National Science and Engineering Research Council of Canada (NSERC) and the Alberta Ingenuity Centre for Machine Learning (AICML). C.H. Lee is supported by NSERC, AICML, and iCORE. Our thanks to Dale Schuurmans for helpful discussions on optimization and parameter estimation, J. Sander for hlepful discussions for the classification issues, BTGP members for help in data processing, and Albert Murtha (M.D.) for domain knowledge on the tumor data set.

References

1. Lafferty, J., Pereira, F., McCallum, A.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ICML (2001)
2. Shawe-Taylor, Cristianini: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge, UK (2004)
3. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer-Verlag, Tokyo (2001)
4. Kumar, S., Hebert, M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. ICCV (2003) 1150–1157
5. Besag, J.: On the statistical analysis of dirty pictures. Journal of Royal Statistical Society. Series B **48** (1986) 3:259–302
6. Kumar, S., Hebert, M.: Discriminative fields for modeling spatial dependencies in natural images. NIPS (2003)
7. R.Fletcher: Practical Methods of Optimization. John Wiley & Sons (1987)
8. Platt, J.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. MIT Press, Cambridge, MA (2000)
9. Joachims, T.: Making large-scale svm learning practical. In Scholkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods - Support Vector Learning, MIT Press (1999)
10. Gering, D.: Recognizing Deviations from Normalcy for Brain Tumor Segmentation. PhD thesis, MIT (2003)
11. Zhang, J., Ma, K., Er, M., Chong, V.: Tumor segmentation from magnetic resonance imaging by learning via one-class support vector machine. Int. Workshop on Advanced Image Technology (2004) 207–211
12. Garcia, C., Moreno, J.: Kernel based method for segmentation and modeling of magnetic resonance images. LNCS **3315** (2004) 636–645
13. : Statistical parametric mapping, http://www.fil.ion.bpmf.ac.uk/spm/ (Online)
14. Hellier, P., Ashburner, J., Corouge, I., Barillot, C., Friston, K.: Inter subject registration of functional and anatomical data using spm. In: MICCAI. Volume 587-590. (2002)
15. Ashburner, J.: Another mri bias correction approach. In: 8th Int. Conf. on Functional Mapping of the Human Brain, Sendai, Japan. (2002)

Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication*

J ˙ ij Le ovec[1], Dee ayan Cha ˙ aba˙ i[1],
Jon Kleinbe˙ g[2], and Ch˙ i o Falo . o [1]

[1] School of Computer Science, Carnegie Mellon University
{jure, deepay, christos}@cs.cmu.edu
[2] Department of Computer Science, Cornell University
kleinber@cs.cornell.edu

Abstract. How can we generate realistic graphs? In addition, how can we do so with a mathematically tractable model that makes it feasible to analyze their properties rigorously? Real graphs obey a long list of surprising properties: Heavy tails for the in- and out-degree distribution; heavy tails for the eigenvalues and eigenvectors; small diameters; and the recently discovered "Densification Power Law" (DPL). All published graph generators either fail to match several of the above properties, are very complicated to analyze mathematically, or both. Here we propose a graph generator that is mathematically tractable and matches this collection of properties. The main idea is to use a non-standard matrix operation, the *Kronecker product*, to generate graphs that we refer to as "Kronecker graphs".

We show that Kronecker graphs naturally obey all the above properties; in fact, we can rigorously *prove* that they do so. We also provide empirical evidence showing that they can mimic very well several real graphs.

1 Introduction

Wha do ˙eal g˙a h loo li e? How do hey evolve ove˙ i e? How can we gene˙a e . yn he ic, b ˙eali ic, i e-evolving g˙a h ? G˙a h . ining ha been a ˙ac ing . ch in e˙e ˙ecen ly, wi h an e ha i on nding a e˙n and abno˙ ali ie in . ocial ne wo˙ . , co e˙ ne wo˙ . , e- ail in e˙ac ion , gene

* Work partially supported by the National Science Foundation under Grants No. IIS-0209107, SENSOR-0329549, IIS-0326322, CNS-0433540, CCF-0325453, IIS-0329064, CNS-0403340, CCR-0122581, a David and Lucile Packard Foundation Fellowship, and also by the Pennsylvania Infrastructure Technology Alliance (PITA), a partnership of Carnegie Mellon, Lehigh University and the Commonwealth of Pennsylvania's Department of Community and Economic Development (DCED). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 133–145, 2005.

, eg la o, y ne wo, , and any o, e. Mo of he wo, foc e on a ic na - ho of g, a h , whe, e fa cina ing “law ” have been di cove, ed, incl ding all dia e e, and heavy- ailed deg, ee di , ib ion .

A , eali ic g, a h gene, a o, i i o, an fo, a lea wo , ea on . The , i ha i can gene, a e g, a h fo, ex , a ola ion , “wha -if” cena, io , and i la- ion , when , eal g, a h a, e di c l o, i o ible o collec . Fo, exa le, how well will a given , o ocol , n on he In e, ne ve yea, f, o now? Acc , a e g, a h gene, a o, can , od ce o, e , eali ic odel fo, he f , e In e, ne , on which i la ion can be , n. The econd , ea on i o, e b le: i fo, ce o hin abo he a e, n ha a g, a h gene, a o, ho ld obey, o be , eali ic.

The ain con , ib ion of hi a e, a, e he following:

- We , ovide a gene, a o, which obey all he ain a ic a e, n ha have a ea, ed in he li e, a , e.
- Gene, a o, al o obey he , ecen ly di cove, ed e o, al evol ion a e, n .
- Con , a, y o o he, gene, a o, ha a ch hi co bina ion of , o e, ie , o , gene, a o, lead o , ac able analy i and , igo, o , oof .

O , gene, a o, i ba ed on a non- anda, d a , ix o e, a ion, he , . . . , . . . The, e a, e eve, al heo, e on K, onec e, , od c , which ac ally co, - , e ond exac ly o a igni can o, ion of wha we wan o , ove: heavy- ailed di , ib ion fo, in-deg, ee, o -deg, ee, eigenval e , and eigenvec o, . We al o de on , a e how a K, onec e, G, a h can a ch he behavio, of eve, al , eal g, a h (a en ci a ion , a e, ci a ion , and o he,). While K, onec e, , od- c have been died by he algeb, aic co bina o, ic co ni y (ee e.g. [10]), he , e en wo, i he , o e loy hi o e, a ion in he de ign of ne wo, odel o a ch , eal da a e .

The , e of he a e, i o, gani ed a follow : Sec ion 2 , vey he , ela ed li e, a , e. Sec ion 3 give he , o o ed e hod. We , e en he ex e, i en al , e l in Sec ion 4, and we clo e wi h o e di c ion and concl ion .

2 Related Work

Fi, , we will di c he co only fo nd (a ic) a e, n in g, a h , hen o e , ecen a e, n on e o, al evol ion, and nally, he a e of he a, in g, a h gene, a ion e hod .

Static Graph Patterns: While any a e, n have been di cove, ed, wo of he , inci al one a, e heavy- ailed deg, ee di , ib ion and all dia e e, ... The deg, ee-di , ib ion of a g, a h i a owe, law if he n be, of node c_k wi h deg, ee k i given by $c_k \propto k^{-\gamma}$ $(\gamma > 0)$ whe, e γ i called he owe, -law ex onen . Powe, law have fo nd in he In e, ne [13], he Web [15,7], ci a ion g, a h [24], online ocial ne wo, [9] and any o he, . Devia ion f, o he owe, -law a e, n have been no iced [23], which can be ex lained by he “DGX” di , ib ion [5]. DGX i clo ely , ela ed o a , nca ed logno, al di , ib ion.

[illegible] Mo[illegible]eal-wo[illegible]ld g[illegible]a[illegible]h exhibi[illegible]ela ively [illegible] all dia[illegible]e e[illegible] (he "[illegible] all- wo[illegible]ld" [illegible]heno[illegible]enon): A g[illegible]a[illegible]h ha[illegible] dia[illegible]e e[illegible] d if eve[illegible]y [illegible]ai[illegible] of node can be connec ed by a[illegible]a[illegible]h of leng h a[illegible]o[illegible] d. The dia[illegible]e e[illegible] d i[illegible]ce ible o o[illegible]lie[illegible]. Th[illegible], a[illegible]o[illegible]e[illegible]ob[illegible]ea[illegible]e of he[illegible]ai[illegible] wi e di ance be ween node of a g[illegible]a[illegible]h i he [illegible] [26]. Thi i de ned a he[illegible]ini[illegible]n[illegible]be[illegible] of ho[illegible] in which[illegible]o[illegible]e f[illegible]ac ion (o[illegible] an ile q, [illegible]ay $q = 90\%$) of all connec ed [illegible]ai[illegible] of node can[illegible] each each o he[illegible]. The effec ive dia[illegible]e e[illegible] ha[illegible] been fo nd o be [illegible] all fo[illegible] la[illegible] ge[illegible]eal-wo[illegible]ld g[illegible]a[illegible]h , li e In e[illegible]ne , Web, and[illegible]ocial ne wo[illegible] [2,21].

[illegible] Thi i a[illegible]lo of he eigenval e (o[illegible]ing la[illegible] val e) of he adjacency[illegible]a[illegible]ix of he g[illegible]a[illegible]h, ve[illegible] hei[illegible]an , [illegible]ing a log-log[illegible]cale. The[illegible]c[illegible]ee[illegible]lo i al o of en fo nd o a[illegible]oxi a ely obey a[illegible]owe[illegible] law. The di[illegible]ib[illegible]ion of eigenvec o[illegible] co[illegible]onen[illegible] (indica o[illegible] of "ne wo[illegible] val e") ha al o been fo nd o be[illegible]ewed [9].

A a[illegible] f[illegible]o[illegible] he e, [illegible]eve[illegible]al o he[illegible] a[illegible]e[illegible]n have been fo nd, incl ding he "[illegible]e[illegible]" [14,9], "[illegible]e ilience" [2,22], "cl[illegible]e[illegible]ing coe cien " and[illegible]any[illegible]o[illegible]e.

Temporal evolution Laws: Densification and shrinking diameter: Two ve[illegible]y[illegible]ecen di cove[illegible]ie , bo h[illegible]ega[illegible]ding i e-evolving g[illegible]a[illegible]h , a[illegible]e wo[illegible] h[illegible]en ioning [18]: (a) he "effec ive dia[illegible]e e[illegible]" of g[illegible]a[illegible]h end[illegible]o[illegible]h in o[illegible]abili e a he g[illegible]a[illegible]h g[illegible]ow wi h i e, and (b) he n[illegible]be[illegible] of edge $E(t)$ and node $N(t)$[illegible]ee[illegible] o obey he [illegible] (DPL), which[illegible]a e ha

$$E(t) \propto N(t)^a \tag{1}$$

The [illegible] a i y ically g[illegible]ea e[illegible] han 1, i[illegible] lying ha he ave[illegible]age deg[illegible]ee of a node in he g[illegible]a[illegible]h i[illegible] ove[illegible] i e. Thi[illegible]ean ha[illegible]eal g[illegible]a[illegible]h end[illegible]o[illegible]o[illegible]any[illegible]o[illegible]e edge han node , and h[illegible]a[illegible]e den ifying a hey g[illegible]ow.

Graph Generators: The ea[illegible]lie[illegible]obabili ic gene[illegible]a ive[illegible]odel fo[illegible] g[illegible]a[illegible]h wa a[illegible]ando[illegible] g[illegible]a[illegible]h[illegible]odel, whe[illegible]e each[illegible]ai[illegible] of node ha an iden ical, inde enden[illegible]obabili y of being joined by an edge [11]. The[illegible]dy of hi[illegible]odel ha led o a[illegible]ich[illegible]a he a ical heo[illegible]y; howeve[illegible], hi gene[illegible]a o[illegible]od ce g[illegible]a[illegible]h ha fail o[illegible]a ch[illegible]eal-wo[illegible]ld ne wo[illegible] in a n[illegible]be[illegible] of[illegible]e ec[illegible] (fo[illegible] exa[illegible]le, i doe no[illegible]od ce heavy- ailed deg[illegible]ee di[illegible]ib[illegible]ion).

The va[illegible]ajo[illegible]i y of[illegible]ecen[illegible]odel involve[illegible]o e fo[illegible] of[illegible] [1,2,28,15,16]: new node join he g[illegible]a[illegible]h a each i e[illegible]e , and[illegible]efe[illegible]en- ially connec o exi ing node wi h high deg[illegible]ee (he "[illegible]ich ge[illegible]iche[illegible]"). Thi[illegible]i le behavio[illegible] lead o owe[illegible]-law ail and o low dia[illegible]e e[illegible]. The dia[illegible]e e[illegible] in hi[illegible]odel g[illegible]ow[illegible]lowly wi h he n[illegible]be[illegible] of node N, which viola e he "[illegible]h[illegible]in ing dia[illegible]e e[illegible]"[illegible]o[illegible]e[illegible]y[illegible]en ioned above.

Ano he[illegible] fa ily of g[illegible]a[illegible]h-gene[illegible]a ion[illegible]e hod[illegible]ive fo[illegible] all dia[illegible]e e[illegible], li e he[illegible] gene[illegible]a o[illegible] [27] and he Wax an gene[illegible]a o[illegible] [6]. A hi[illegible]d fa ily of[illegible]e hod[illegible]how ha heavy ail e[illegible]e[illegible]ge if node[illegible]y o o i i e hei[illegible] connec ivi y nde[illegible]e o[illegible]ce con[illegible]ain[illegible] [8,12].

Summary: Mo[illegible]c[illegible]en gene[illegible]a o[illegible] foc[illegible] on only one ([illegible]a ic)[illegible]a[illegible]e[illegible]n, and neglec he o he[illegible]. In addi ion, i i[illegible] ally ha[illegible]d o[illegible]ove[illegible]o e[illegible]ie of he[illegible]. The gene[illegible]a o[illegible] we de c[illegible]ibe in he nex[illegible]ec ion add[illegible]e[illegible]e he e i[illegible]e[illegible].

3 Proposed Method

The e hod we o o e i ba ed on a ec ive con c ion. De ning he ec - ion o e ly i o ewha b le, a a n be of anda d, ela ed g a h con- c ion e hod fail o od ce g a h ha den ify acco ding o he a e n ob e ved in ac ice, and hey al o od ce g a h who e dia e e inc ea e. To od ce den ifying g a h wi h con an dia e e , and he eby a ch he ali a ive behavio of eal ne wo da a e , we develo a oced e ha i be de c ibed in e of he [illegible] of a ice . To hel in he de c i ion of he e hod, he acco anying able ovide a li of y bol and hei de ni ion .

Sy bol	De ni ion
G_1	he ini ia o of a K onec e G a h
N_1	n be of node in ini ia o
E_1	n be of edge in ini ia o
$G_1^{[k]} = G_k$	he k^{th} K onec e owe of G_1
a	den i ca ion ex onen
d	dia e e of a g a h
$\mathcal{P}_1$	obabili y a ix

3.1 Main Idea

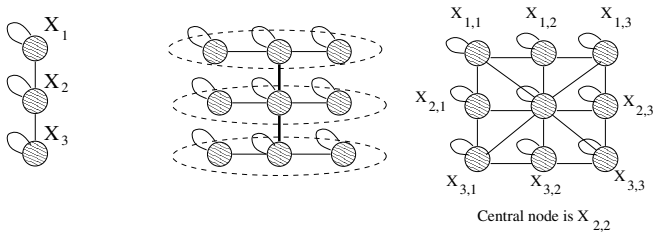
The ain idea i o c ea e elf- i ila g a h , ec ively. We begin wi h an [illegible] g a h G_1, wi h N_1 node and E_1 edge , and by ec ion we od ce cce ively la ge g a h $G_2 \ldots G_n$ ch ha he k^{th} g a h G_k i on $N_k = N_1^k$ node . If we wan he e g a h o exhibi a ve ion of he Den i ca ion Powe Law, hen G_k ho ld have $E_k = E_1^k$ edge . Thi i a o e y ha e i e o e ca e in o de o ge igh , a anda d ec ive con c ion (fo exa le, he adi ional Ca e ian od c o he con c ion of [4]) do no a i fy i .

I n o ha he [illegible] of wo a ice i he e fec ool fo hi goal. The K onec e od c i de ned a follow :

Definition 1 (Kronecker product of matrices). [illegible] $\mathbf{A} = [a_{i,j}]$ [illegible] $\mathbf{B}$ [illegible] $n \times m$ [illegible] $n' \times m'$ [illegible] $\mathbf{C}$ [illegible] $(n * n') \times (m * m')$ [illegible]

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} \doteq \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \ldots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \ldots & a_{2,m}\mathbf{B} \\ & & & \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \ldots & a_{n,m}\mathbf{B} \end{pmatrix} \tag{2}$$

We de ne he K onec e od c of wo g a h a he K onec e od c of hei adjacency a ice .



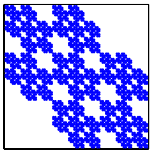
(a) Graph G_1 (b) Intermediate stage (c) Graph $G_2 = G_1 \otimes G_1$

1	1	0
1	1	1
0	1	1

(d) Adjacency matrix of G_1

G_1	G_1	0
G_1	G_1	G_1
0	G_1	G_1

(e) Adjacency matrix of $G_2 = G_1 \otimes G_1$



(f) Plot of G_4

Fig. 1. *Example of Kronecker multiplication:* Top: a "3-chain" and its Kronecker product with itself; each of the X_i nodes gets expanded into 3 nodes, which are then linked using Observation 1. Bottom row: the corresponding adjacency matrices, along with matrix for the fourth Kronecker power G_4.

Observation 1 (Edges in Kronecker-multiplied graphs)

$$(X_{ij}, X_{kl}) \in G \otimes H \text{ iff } (X_i, X_k) \in G \text{ and } (X_j, X_l) \in H$$

X_{ij} X_{kl} $G \otimes H$, X_i, X_j, X_k X_l G H,

The la ob e va ion i b le, b c cial, and de e ve elabo a ion: Fig- e 1(a–c) how he ec ive con c ion of $G \otimes H$, when $G = H$ i a 3-node a h. Con ide node $X_{1,2}$ in Fig e 1(c): I belong o he H g a h ha e laced node X_1 (ee Fig e 1(b)), and in fac i he X_2 node (i.e., he cen e) wi hin hi all H-g a h.

We o o e o od ce a g owing e ence of g a h by i e a ing he K o- nec e od c :

Definition 2 (Kronecker power). k^{th} G_1 $G_1^{[k]}$ (G_k)

$$G_1^{[k]} = G_k = \underbrace{G_1 \otimes G_1 \otimes \ldots G_1}_{k} = G_{k-1} \otimes G_1$$

The elf- i ila na e of he K onec e g a h od c i clea : To od ce G_k f o G_{k-1}, we "ex and" (e lace) each node of G_{k-1} by conve ing i in o a co y of G, and we join he e co ie oge he acco ding o he adjacencie in

G_{k-1} (ee Fig e 1). Thi oce i ve y na al: one can i agine i a o i ing ha co ni ie wi h he g a h g ow ec ively, wi h node in he co ni y ec ively ge ing ex anded in o inia e co ie of he co ni y. Node in he bco ni y hen lin a ong he elve and al o o node f o diffe en co ni ie .

3.2 Theorems and Proofs

We hall now di c he o e ie of K onec e g a h , eci cally, hei de- g ee di ib ion , dia e e , eigenval e , eigenvec o , and i e-evol ion. O abili y o ove analy ical e l abo all of he e o e ie i a ajo advan- age of K onec e g a h ove o he gene a o . The nex few heo e ove ha eve al di ib ion of in e e a e l ino ial fo o K onec e g a h odel. Thi i i o an , beca e a ca ef l choice of he ini ial g a h G_1 can a e he e l ing l ino ial di ib ion o behave li e a owe -law o DGX di ib ion.

Theorem 1 (Multinomial degree distribution). [illegible]

Le he ini ia o G_1 have he deg ee e ence $d_1, d_2, \ldots, d_{N_1}$. K onec e l i lica ion of a node wi h deg ee d ex and i in o N_1 node , wi h he co- e onding deg ee being $d \times d_1, d \times d_2, \ldots, d \times d_{N_1}$. Af e K onec e owe ing, he deg ee of each node in g a h G_k i of he fo $d_{i_1} \times d_{i_2} \times \ldots d_{i_k}$, wi h $i_1, i_2, \ldots i_k \in (1 \ldots N_1)$, and he e i one node fo each o de ed co bina ion. Thi give he l ino ial di ib ion on he deg ee of G_k. No e al o ha he deg ee of node in G_k can be ex e ed a he k^{th} K onec e owe of he vec o $(d_1, d_2, \ldots, d_{N_1})$. □

Theorem 2 (Multinomial eigenvalue distribution). [illegible] G_k [illegible]

Le G_1 have he eigenval e $\lambda_1, \lambda_2, \ldots, \lambda_{N_1}$. By o e ie of he K onec e l i lica ion [19,17], he eigenval e of G_k a e k^{th} K onec e owe of he vec o $(\lambda_1, \lambda_2, \ldots, \lambda_{N_1})$. A in Theo e 1, he eigenval e di ib ion i a l ino ial. □

A i ila a g en ing o e ie of K onec e a ix l i lica ion how he following.

Theorem 3 (Multinomial eigenvector distribution). [illegible] G_k [illegible]

We have j cove ed eve al of he a ic g a h a e n . No ice ha he oof we e di ec con e ence of he K onec e l i lica ion o e ie .

Nex we con in e wi h he e o al a e n : he den i ca ion owe law, and h in ing/ abili ing dia e e .

Theorem 4 (DPL). [illegible] $a = \log(E_1)/\log(N_1)$

[illegible] Since he k^{th} K onec e owe G_k ha $N_k = N_1^k$ node and $E_k = E_1^k$ edge , i a i e $E_k = N_k^a$, whe e $a = \log(E_1)/\log(N_1)$. The c cial oin i ha hi ex onen a i inde enden of k, and hence he e ence of K onec e owe follow an exac ve ion of he Den i ca ion Powe Law. □

We now how how he K onec e od c al o e e ve he o e y of con an dia e e , a c cial ing edien fo a ching he dia e e o e ie of any eal-wo ld ne wo da a e . In o de o e abli h hi , we will a e ha he ini ia o g a h G_1 ha a elf-loo on eve y node; o he wi e, i K onec e owe ay in fac be di connec ed.

Lemma 1. [illegible] G [illegible] H [illegible] d [illegible] $G \otimes H$ [illegible] d

[illegible] Each node in $G \otimes H$ can be e e en ed a an o de ed ai (v, w), wi h v a node of G and w a node of H, and wi h an edge joining (v, w) and (x, y) eci ely when (v, x) i an edge of G and (w, y) i an edge of H. Now, fo an a bi a y ai of node (v, w) and (v', w'), we [illegible] how ha he e i a a h of leng h a o d connec ing he . Since G ha dia e e a o d, he e i a a h $v = v_1, v_2, \ldots, v_r = v'$, whe e $r \leq d$. If $r < d$, we can conve hi in o a a h $v = v_1, v_2, \ldots, v_d = v'$ of leng h exac ly d, by i ly e ea ing v' a he end fo $d - r$ i e By an analogo a g en , we have a a h $w = w_1, w_2, \ldots, w_d = w'$. Now by he de ni ion of he K onec e od c , he e i an edge joining (v_i, w_i) and (v_{i+1}, w_{i+1}) fo all $1 \leq i \leq d - 1$, and o $(v, w) = (v_1, w_1), (v_2, w_2), \ldots, (v_d, w_d) = (v', w')$ i a a h of leng h d connec ing (v, w) o (v', w'), a e i ed. □

Theorem 5. [illegible] G_1 [illegible] d [illegible] k [illegible] G_k [illegible] d

[illegible] Thi follow di ec ly f o he evio le a, co bined wi h ind c ion on k. □

We al o con ide he [illegible] d_e; we de ne he q-effec ive dia e e a he ini d_e ch ha , fo a lea a q f ac ion of he eachable node ai , he a h leng h i a o d_e. The q-effec ive dia e e i a o e ob an i y han he dia e e , he la e being one o he effec of degene a e c e in he g a h (e.g. ve y long chain); howeve , he q-effec ive dia e e and dia e e end o exhibi ali a ively i ila behavio . Fo e o ing e l in b e en ec ion , we will gene ally con ide he q-effec ive dia e e wi h $q = .9$, and efe o hi i ly a he [illegible]

Theorem 6 (Effective Diameter). [illegible] G_1 [illegible] d [illegible] q [illegible] q [illegible] G_k [illegible] d [illegible] k [illegible]

[illegible] To [illegible]ove hi[illegible], i i[illegible] cien[illegible] o[illegible] how ha[illegible] fo[illegible] wo[illegible]ando[illegible]ly[illegible] elec[illegible]ed node of G_k, he[illegible]obabili[illegible]y ha[illegible] hei[illegible] di[illegible]ance i[illegible] d conve[illegible]ge[illegible] o 1 a[illegible] k goe[illegible] o in[illegible]ni[illegible]y.

We e[illegible]abli[illegible]h hi[illegible] a[illegible] follow[illegible]. Each node in G_k can be[illegible]e[illegible]e en[illegible]ed a[illegible] an o[illegible]de[illegible]ed[illegible]e[illegible]ence of k node[illegible] f[illegible]o[illegible] G_1, and we can view he[illegible]ando[illegible] elec[illegible]ion of a node in G_k a[illegible]a[illegible]e[illegible]ence of k inde[illegible]enden[illegible]ando[illegible] node[illegible] elec[illegible]ion f[illegible]o[illegible] G_1. S[illegible]o e[illegible] ha[illegible] $v = (v_1, \ldots, v_k)$ and $w = (w_1, \ldots, w_k)$ a[illegible]e wo[illegible] ch[illegible]ando[illegible]ly[illegible] elec[illegible]ed node[illegible] f[illegible]o[illegible] G_k. Now, if x and y a[illegible]e wo node[illegible] in G_1 a[illegible] di[illegible]ance d ([illegible] ch a[illegible] ai[illegible] (x, y) exi[illegible] ince G_1 ha[illegible] dia[illegible]e e[illegible] d), hen wi[illegible]h[illegible]obabili[illegible]y $1 - (1 - 2/N_1)^k$, he[illegible]e i[illegible]o[illegible]e index j fo[illegible] which $\{v_j, w_j\} = \{x, y\}$. If he[illegible]e i[illegible] ch an index, hen he di[illegible]ance be[illegible]ween v and w i[illegible] d. A[illegible] he ex[illegible]e[illegible]ion $1 - (1 - 2/N_1)^k$ conve[illegible]ge[illegible] o 1 a[illegible] k inc[illegible]ea[illegible]e, i[illegible] follow[illegible] ha[illegible] he q-effec[illegible]ive dia[illegible]e[illegible]e[illegible] i[illegible] conve[illegible]ging o[illegible] d. □

3.3 Stochastic Kronecker Graphs

While he K[illegible]onec[illegible]e[illegible] owe[illegible] con[illegible] c[illegible]ion di[illegible]c[illegible]ed h[illegible] fa[illegible] yield g[illegible]a[illegible]h wi[illegible]h a[illegible]ange of de[illegible]i[illegible]ed[illegible]o[illegible]e[illegible] ie[illegible], i[illegible] di[illegible]c[illegible]e[illegible]e na[illegible]e[illegible]od[illegible]ce "[illegible] ai[illegible]ca[illegible]e effec[illegible]" in he deg[illegible]ee and[illegible]ec[illegible]al[illegible] an[illegible]i[illegible]ie[illegible], i[illegible]ly beca[illegible]e individ[illegible]al val[illegible]e have la[illegible]ge[illegible]l[illegible]i[illegible]lici[illegible]ie[illegible]. He[illegible]e we[illegible]o[illegible]o[illegible]e a[illegible]ocha[illegible]ic ve[illegible]ion of K[illegible]onec[illegible]e[illegible] g[illegible]a[illegible]h[illegible] ha[illegible] eli[illegible]ina[illegible]e[illegible] hi[illegible] effec[illegible]. co[illegible]n[illegible]e[illegible] a[illegible].

We[illegible] a[illegible] wi[illegible]h an $N_1 \times N_1$[illegible] $\mathcal{P}_1$: he val[illegible]e p_{ij} deno[illegible]e he[illegible]obabili[illegible]y ha[illegible] edge (i, j) i[illegible]e[illegible]e en[illegible]. We co[illegible]e i[illegible] k^{th} K[illegible]onec[illegible]e[illegible] owe[illegible] $\mathcal{P}_1^{[k]} = \mathcal{P}_k$; and hen fo[illegible] each en[illegible]y p_{uv} of $\mathcal{P}_k$, we incl[illegible]de an edge be[illegible]ween node[illegible] u and v wi[illegible]h[illegible]obabili[illegible]y $p_{u,v}$. The[illegible]e[illegible]l[illegible]ing bina[illegible]y[illegible]ando[illegible] a[illegible]ix $R = R(\mathcal{P}_k)$ will be called he[illegible] (o[illegible]).

In[illegible]inci[illegible]le one co[illegible]ld[illegible]y choo[illegible]ing each of he N_1^2[illegible] a[illegible]a[illegible]e[illegible]e[illegible] fo[illegible] he[illegible] a[illegible]ix $\mathcal{P}_1$[illegible]e[illegible] a[illegible]a[illegible]ely. Howeve[illegible], we[illegible]ed[illegible]ce he n[illegible] be[illegible] of[illegible] a[illegible]a[illegible]e[illegible]e[illegible] o[illegible]j[illegible] wo: α and β. Le[illegible] G_1 be he ini[illegible]ia[illegible] o[illegible] a[illegible]ix (bina[illegible]y, de[illegible]e[illegible] ini[illegible]ic); we c[illegible]ea[illegible]e he co[illegible]e[illegible]onding[illegible]obabili[illegible]y[illegible] a[illegible]ix $\mathcal{P}_1$ by[illegible]e[illegible]lacing each "1" and "0" of G_1 wi[illegible]h α and β[illegible]e[illegible]ec[illegible]ively ($\beta \leq \alpha$). The[illegible]e[illegible]l[illegible]ing[illegible]obabili[illegible]y[illegible] a[illegible]ice[illegible] ain ain — wi[illegible]h[illegible] o[illegible]e[illegible]ando[illegible] noi[illegible]e — he[illegible] elf- i[illegible]ila[illegible] [illegible] c[illegible]e of he K[illegible]onec[illegible]e[illegible] g[illegible]a[illegible]h in he[illegible]evio[illegible] b[illegible]ec[illegible]ion (which, fo[illegible] cla[illegible]i[illegible]y, we call [illegible]).

We[illegible]nd[illegible]e[illegible] i[illegible]ically ha[illegible] he[illegible]ando[illegible] g[illegible]a[illegible]h[illegible]od[illegible]ced by hi[illegible] odel con[illegible]in[illegible]e o exhibi[illegible] he de[illegible]i[illegible]ed[illegible]o[illegible]e[illegible]ie of[illegible]eal da[illegible]a[illegible]e[illegible], and wi[illegible]ho[illegible] he[illegible] ai[illegible]ca[illegible]e effec[illegible] of he de[illegible]e[illegible] ini[illegible]ic ve[illegible]ion. The a[illegible] of[illegible]e[illegible]ing α and β o[illegible] a[illegible]ch ob[illegible]e[illegible]ved da[illegible]a i[illegible]a ve[illegible]y[illegible]o[illegible] i[illegible]ing[illegible]e ea[illegible]ch di[illegible]ec[illegible]ion, o[illegible] ide he[illegible] co[illegible]e of hi[illegible] a[illegible]e[illegible]. In o[illegible] ex[illegible]e[illegible]i[illegible]en[illegible] in he[illegible] co[illegible]ing[illegible]ec[illegible]ion[illegible], we[illegible]e he[illegible]i[illegible]ic which we de[illegible]c[illegible]ibe he[illegible]e.

4 Experiments

Now, we de[illegible]on[illegible]a[illegible]e he abili[illegible]y of K[illegible]onec[illegible]e[illegible] g[illegible]a[illegible]h[illegible] o[illegible] a[illegible]ch he[illegible] a[illegible]e[illegible]n of[illegible]eal-wo[illegible]ld g[illegible]a[illegible]h[illegible]. The da[illegible]a[illegible]e[illegible] we[illegible]e a[illegible]e:

- [illegible] Thi[illegible] i[illegible] a ci[illegible]a[illegible]ion g[illegible]a[illegible]h fo[illegible] high-ene[illegible]gy[illegible] hy[illegible]ic[illegible]e[illegible]ea[illegible]ch[illegible] a[illegible]e[illegible], wi[illegible]h a[illegible]o[illegible]al of $N = 29,555$[illegible] a[illegible]e[illegible] and $E = 352,807$ ci[illegible]a[illegible]ion[illegible]. We follow i[illegible] evol[illegible]ion f[illegible]o[illegible] Jan[illegible]a[illegible]y 1993[illegible]o A[illegible]il 2003, wi[illegible]h one da[illegible]a-[illegible]oin[illegible]e[illegible] on[illegible]h.

• [illegible] Thi i a U.S. a en ci a ion da a e ha an 37 yea. f o Jan a y 1963 o Dece be 1999. The g a h con ain a o al of $N = 3,942,825$ a en and $E = 16,518,948$ ci a ion . Ci a ion g a h a e no ally con ide ed a di ec ed g a h . Fo he o e of hi wo we hin of he a ndi ec ed.

• [illegible] We al o analy e a a ic da a e con i ing of a in-gle na ho of connec ivi y a ong In e ne a ono o y e f o Jan a y 2000, wi h $N = 6,474$ and $E = 26,467$.

We ob e ve wo ind of g a h a e n — " a ic" and " e o al." A en ioned ea lie , co on a ic a e n incl de he deg ee di ib ion, he c ee lo (eigenval e of g a h adjacency a ix v . an), inci al eigenvec o of adjacency a ix and he di ib ion of connec ed co onen . Te o al a e n incl de he dia e e ove i e, he i e of he gian co onen ove i e, and he den i ca ion owe law. Fo he dia e e co a ion, we e a oo hed ve ion of he effec ive dia e e ha i ali a ively i ila o he anda d effec ive dia e e , b e linea in e ola ion o a o a e on non-in ege val e ; ee [18] fo f he de ail on hi calc la ion.

Re l a e hown in Fig e 2 and 3 fo he g a h which evolve ove i e (and). Fo b evi y, we how he lo fo only wo a ic and wo e o al a e n . We ee ha he de e ini ic K onec e odel al eady ca e he ali a ive c e of he deg ee and eigenval e di ib ion , a well a he e o al a e n e en ed by he Den i ca ion Powe Law and he abili ing dia e e . Howeve , he de e ini ic na e of hi odel e l in " ai ca e" behavio , a hown in c ee lo fo he de e ini ic K onec e g a h of Fig e 2 (econd ow, econd col n). We ee ha he S ocha ic K o-nec e G a h oo h o he e di ib ion , f he a ching he ali a ive c e of he eal da a; hey al o a ch he h in ing-befo e- abili a ion end of he dia e e of eal g a h .

Fo he S ocha ic K onec e G a h we need o e i a e he a a e e α and β de ned in he evio ec ion. Thi lead o in e e ing e ion who e f ll e ol ion lie beyond he co e of he e en a e ; c en ly, we ea ched by b e fo ce ove (he ela ively all n be of) o ible ini ia o g a h of o ve node , and we hen cho e α and β o a o a ch well he edge den i y, he axi deg ee, he ec al o e ie , and he DPL ex onen .

Finally, Fig e 4 how lo fo he a ic a e n in he [illegible] g a h. Recall ha we analy e a ingle, a ic na ho in hi ca e. In addi ion o he deg ee di ib ion and c ee lo , we al o how wo y ical lo [9]: he di ib ion of [illegible] (inci al eigenvec o co onen , o ed, ve an) and he [illegible] (he n be of eachable ai $P(h)$ wi hin h ho o le , a a f nc ion of he n be of ho h).

5 Observations and Conclusions

He e we li eve al of he de i able o e ie of he o o ed K onec e G a h and S ocha ic K onec e G a h .

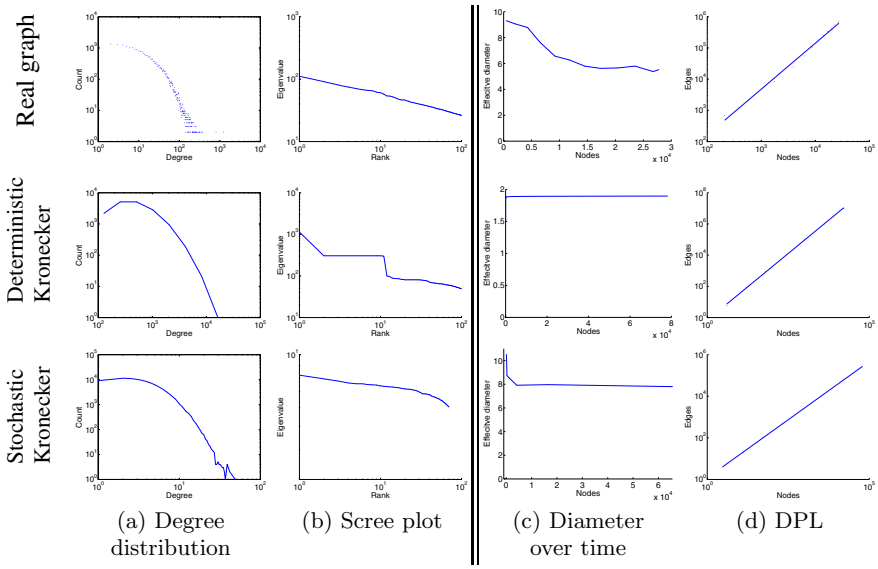


Fig. 2. *arXiv dataset:* Patterns from the real graph (top row), the deterministic Kronecker graph with G_1 being a star graph with 3 satellites (middle row), and the Stochastic Kronecker graph ($\alpha = 0.41,\ \beta = 0.11$ – bottom row). *Static* patterns: (a) is the PDF of degrees in the graph (log-log scale), and (b) the distribution of eigenvalues (log–log scale). *Temporal* patterns: (c) gives the effective diameter over time (linear-linear scale), and (d) is the number of edges versus number of nodes over time (log-log scale). Notice that the Stochastic Kronecker Graph qualitatively matches all the patterns very well.

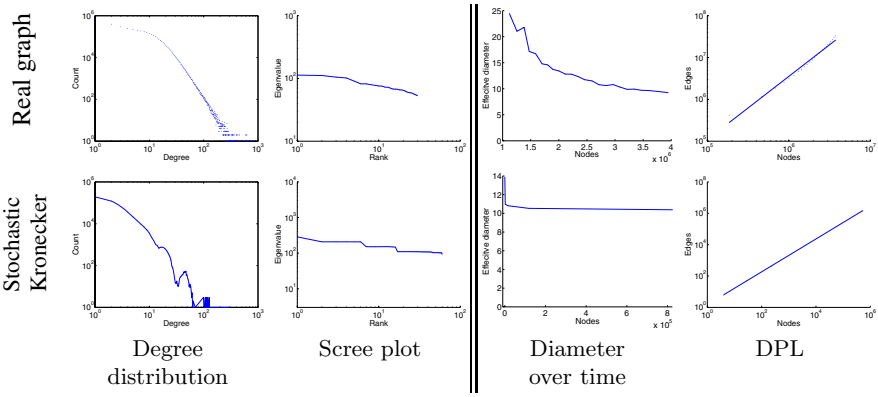
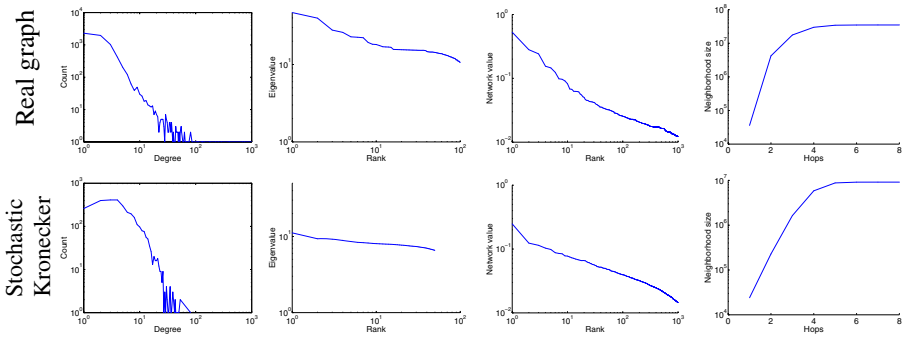


Fig. 3. *Patents:* Again, Kronecker graphs match all of these patterns. We show only the Stochastic Kronecker graph for brevity.



(a) Degree distribution (b) Scree plot (c) "Network Value" distribution (d) "Hop-plot"

Fig. 4. *Autonomous systems*: Real (top) versus Kronecker (bottom). Columns (a) and (b) show the degree distribution and the scree plot, as before. Columns (c) and (d) show two more static patterns (see text). Notice that, again, the Stochastic Kronecker Graph matches well the properties of the real graph.

[illegible] S ocha ic K onec e G a h incl de eve al o he gene a o , a [illegible] ecial ca e : Fo α=β, we ob ain an E dő -Renyi ando g a h; fo α=1 and β=0, we ob ain a de e ini ic K onec e g a h; e ing he G_1 a ix o a 2x2 a ix, we ob ain he RMAT gene a o [9]. In con a o K onec e g a h , he RMAT canno ex a ola e in o he f e, ince i need o now he n be of edge o in e . Th , i i inca able of obeying he "den i ca ion law".

[illegible] The E dő -Renyi g a h exhibi ha e an- i ion [11]. Seve al e ea che a g e ha eal y e a e "a he edge of chao " [3,25]. I n o ha S ocha ic K onec e G a h al o exhibi ha e an i ion . Fo all val e of α and β, S ocha ic K onec e G a h have any all di connec ed co onen ; fo la ge val e hey have a gian co onen wi h all dia e e . In be ween, hey exhibi behavio gge ive of a ha e an i ion: Fo a ca ef lly cho en e of (α, β), he dia e e i la ge, and a gian co onen j a e e ging. We o i he de ail , fo lac of ace.

[illegible] All o heo e a e fo he de e ini ic K o- nec e G a h . Howeve , he e i a lo of wo on he o e ie of ando a ice (ee e.g. [20]), which one co ld o en ially a ly in o de o ove o e ie of he S ocha ic K onec e G a h .

In concl ion, he ain con ib ion of hi wo i a fa ily of g a h gen- e a o , ing a non- adi ional a ix o e a ion, he [illegible] The e l ing g a h (a) have all he a ic o e ie (heavy- ailed deg ee di ib - ion, all dia e e), (b) all he e o al o e ie (den i ca ion, h in ing dia e e), and in addi ion, (c) we can fo ally ove all of he e o e ie .

Seve al of he oof a e ex e ely i le, han o he ich heo y of K o- nec e l i lica ion. We al o ovide oof abo he dia e e and "effec ive dia e e ", and we how ha S ocha ic K onec e G a h can be ned o i ic eal g a h well.

References

1. R. Albert and A.-L. Barabasi. Emergence of scaling in random networks. *Science,* pages 509512, 1999.
2. R. Albert and A.-L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics,* 2002.
3. P. Bak. How nature works : The science of self-organized criticality, Sept. 1996.
4. A.-L. Barabasi, E. Ravasz, and T. Vicsek. Deterministic scale-free networks. *Physica A,* 299:559-564, 2001.
5. Z. Bi, C. Faloutsos, and F. Korn. The DGX distribution for mining massive, skewed data. In KDD, pages 17-26, 2001.
6. B.M.Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications,* 6(9), December 1988.
7. A. Broder, R. Kumar, F. Maghoul1, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web: experiments and models. *In Proceedings of World Wide Web Conference,* 2000.
8. J. M. Carlson and J. Doyle. Highly optimized tolerance: a mechanism for power laws in designed systems. *Physics Review E,* 60(2):1412-1427, 1999.
9. D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *SIAM Data Mining,* 2004.
10. T. Chow. The Q-spectrum and spanning trees of tensor products of bipartite graphs. *Proc. Amer. Math. Soc.,* 125:3155-3161, 1997.
11. P. Erdos and A. Renyi. On the evolution of random graphs. Publication of the Mathematical Institute of the Hungarian Acadamy of Science, 5:17-67, 1960.
12. A. Fabrikant, E. Koutsoupias, and C. H. Papadimitriou. Heuristically optimized trade-offs: A new paradigm for power laws in the internet (extended abstract), 2002.
13. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM,* pages 251-262, 1999.
14. M. Girvan and M. E. J. Newman. Community structure in social and biological networks. In *Proc. Natl. Acad. Sci.* USA, volume 99, 2002.
15. J. M. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. In *Proceedings of the International Conference on Combinatorics and Computing,* 1999.
16. S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the web. VLDB, pages 639-650, 1999.
17. A. N. Langville and W. J. Stewart. The Kronecker product and stochastic automata networks. *Journal of Computation and Applied Mathematics,* 167:429-447, 2004.
18. J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD'05,* Chicago, IL, USA, 2005.
19. C. F. V. Loan. The ubiquitous Kronecker product. *Journal of Computation and Applied Mathematics,* 123:85-100, 2000.
20. M. Mehta. Random Matrices. Academic Press, 2nd edition, 1991.
21. S. Milgram. The small-world problem. *Psychology Today,* 2:60-67, 1967.
22. C. R. Palmer, P. B. Gibbons, and C. Faloutsos. Anf: A fast and scalable tool for data mining in massive graphs. In *SIGKDD,* Edmonton, AB, Canada, 2002.
23. D. M. Pennock, G.W. Flake, S. Lawrence, E. J. Glover, and C. L. Giles. Winners dont take all: Characterizing the competition for links on the web. *Proceedings of the National Academy of Sciences,* 99(8):5207-5211, 2002.

24. S. Redner. How popular is your paper? an empirical study of the citation distribution. *European Physical Journal B,* 4:131-134, 1998.
25. R. Sole and B. Goodwin. *Signs of Life: How Complexity Pervades Biology.* Perseus Books Group, New York, NY, 2000.
26. S. L. Tauro, C. Palmer, G. Siganos, and M. Faloutsos. A simple conceptual model for the internet topology. *In Global Internet, San Antonio, Texas,* 2001.
27. D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature,* 393:440-442, 1998.
28. J. Winick and S. Jamin. Inet-3.0: Internet Topology Generator. Technical Report CSE-TR-456-02, University of Michigan, Ann Arbor, 2002.

A Correspondence Between Maximal Complete Bipartite Subgraphs and Closed Patterns

Jinyan Li, Hai an Li, Donny Soh, and Li oon Wong

Institute for Infocomm Research,
21 Heng Mui Keng Terrace, Singapore 119613
{jinyan, haiquan, studonny, limsoon}@i2r.a-star.edu.sg

Abstract. For an undirected graph G without self-loop, we prove: (i) that the number of closed patterns in the adjacency matrix of G is even; (ii) that the number of the closed patterns is precisely double the number of maximal complete bipartite subgraphs of G; (iii) that for every maximal complete bipartite subgraph, there always exists a unique pair of closed patterns that matches the two vertex sets of the subgraph. Therefore, we can enumerate all maximal complete bipartite subgraphs by using efficient algorithms for mining closed patterns which have been extensively studied in the data mining field.

1 Introduction

In e e in g a h and hei a lica ion ha g own o a ve y b oad ec in he a decade (ee [18] and he P eface of [8]), la gely d e o he ef lne of g a h a odel in any a ea ch a a he a ical e ea ch, elec ical engi-nee ing, co e og a ing, b ine ad ini a ion, ociology, econo ic , a e ing, biology, and ne wo ing and co nica ion . In a ic la , any oble can be odeled wi h (ee he de ni ion below) fo ed by g o ing wo non-ove la ing b e of ve ice of a ce ain g a h ha how a ind of f ll connec ivi y be ween he .

We con ide wo exa le . S o e he e a e p c o e in a obile co - nica ion ne wo . So e eo le have a wide ange of con ac , while o he have few. Which g o of c o e (wi h a axi al n be) have a f ll in e ac-ion wi h ano he g o of c o e ? Thi i a ion can be odeled by a g a h whe e a obile hone c o e i a node and a co nica ion i an edge. Th , a axi al bi a i e bg a h of hi g a h co e ond o wo g o of c - o e be ween who he e exi a f ll co nica ion. Thi oble i i ila o he one died in web ining [4,9,12] whe e web co ni ie a e odeled by bi a i e co e . O econd exa le i abo o ein ' in e ac ion in a cell. The e a e ally ho and of o ein in a cell ha in e ac wi h one ano he . Thi i a ion again can be odeled by a g a h, whe e a o ein i a node and an in e ac ion be ween a ai of o ein fo an edge. Then, li ing all axi-al co le e bi a i e bg a h f o hi g a h can an we e ion ch a which wo o ein g o have a f ll in e ac ion, which i a oble died in biology [14,15].

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 146–156, 2005.

Li ing all axi al co le e bi a i e bg a h ha been died heo e - ically in [5]. The e l i ha all axi al co le e bi a i e bg a h of a g a h can be en e a ed in i e $O(a^3 2^{2a} n)$, whe e a i he a bo ici y of he g a h and n i he n be of ve ice in he g a h. Even ho gh he algo i h ha a linea co lexi y, i i no ac ical fo la ge g a h d e o he la ge con an ove head (a can ea ily be a o nd 10-20 in ac ice) [20]. In hi a- e , we dy hi oble f o a da a ining e ec ive: We e a he i ic da a ining algo i h o e cien ly en e a e all axi al co le e bi a i e bg a h f o a la ge g a h. A ain conce of he da a ining algo i h i called . . . . . The e a e any ecen algo i h and i le en a- ion devo ed o he ining of clo ed a e n f o he o-called . . . . . [2,6,7,13,16,17,19]. The da a c e a e e cien and he ining eed i e endo ly fa . O ain con ib ion he e i he ob e va ion ha he ining of clo ed a e n f o he adjacency a ix of a g a h, e ed a ecial an ac ional da aba e, i e ivalen o he oble of en e a ing all axi al co le e bi a i e bg a h of hi g a h.

The e of hi a e i o gani ed a follow : Sec ion 2 and 3 ovide ba ic de ni ion and o o i ion on g a h and clo ed a e n . In Sec ion 4 we ove ha he e i a one- o-one co e ondence be ween clo ed a e n ai and axi al co le e bi a i e bg a h fo any i le g a h. In Sec ion 5, we e en o ex e i en al e l on a o ein ' in e ac ion g a h. Sec ion 6 di c e o e o he ela ed wo and hen concl de hi a e .

2 Maximal Complete Bipartite Subgraphs

A **graph** $G = \langle V^G, E^G \rangle$ i co i ed of a e of ve ice V^G and a e of edge $E^G \subseteq V^G \times V^G$. We of en o i he e c i in V^G, E^G and o he lace when he con ex i clea . Th o gho hi a e , we a e G i an ndi ec ed g a h wi ho any elf-loo . In o he wo d , we a e ha (i) he e i no edge $(u, u) \in E^G$ and (ii) fo eve y $(u, v) \in E^G$, (u, v) can be e laced by (v, u)— ha i , (u, v) i an no de ed ai .

A g a h H i a **subgraph** of a g a h G if $V^H \subseteq V^G$ and $E^H \subseteq E^G$. A g a h G i **bipartite** if V^G can be a i ioned in o wo non-e y and non- in e ec ing b e V_1 and V_2 ch ha $E^G \subseteq V_1 \times V_2$.Thi bi a i e g a h G i ally deno ed by $G = \langle V_1 \cup V_2, E^G \rangle$. No e ha he e i no edge in G ha join wo ve ice wi hin V_1 o V_2. G i **complete bipartite** if $V_1 \times V_2 = E^G$.

Two ve ice u, v of a g a h G a e aid o be adjacen if $(u, v) \in E^G$— ha i , he e i an edge in G ha connec he . The **neighborhood** $\beta^G(v)$ of a ve ex v of a g a h G i he e of all ve ice in G ha a e adjacen o v— ha i , $\beta^G(v) = \{u \mid (u, v) \text{ o } (v, u) \in E^G\}$. The neighbo hood $\beta^G(X)$ fo a . . . . b e X of ve ice of a g a h G i he e of co on neighbo hood of he ve ice in X— ha i , $\beta^G(X) = \cap_{x \in X} \beta^G(x)$.

No e ha fo any b e X of ve ice of a g a h G ch ha X and $\beta^G(X)$ a e bo h non-e y, i i he ca e ha $H = \langle X \cup \beta^G(X), X \times \beta^G(X) \rangle$ i a co - le e bi a i e bg a h of G. No e al o i i o ible fo a ve ex $v \notin X$ of G o

be adjacen o eve, y ve, ex of $\beta^G(X)$. In hi ca e, he b e X can be ex anded by adding he ve, ex v, while ain aining he a e neighbo, hood. Whe, e o o he ex an ion? We e he following de ni ion of axi al co le e bi a, i e bg, a h .

Definition 1. [illegible] $H = \langle V_1 \cup V_2, E\rangle$ [illegible] **maximal complete bipartite subgraph** [illegible] G [illegible] H [illegible] G [illegible] $\beta^G(V_1) = V_2$ [illegible] $\beta^G(V_2) = V_1$

No all axi al co le e bi a, i e bg, a h a, e e ally in e, e ing. Recall o , ea, lie, o iva ing exa le involving c o e, in a obile co nica ion ne wo, . We wo ld , obably no be ve, y in e, e ed in ho e wo g, o of c - o e, wi h a all i e con aining a ingle e, on o, j a few. In con , a , we wo ld , obably be con ide, ably o, e in e, e ed if one of he g, o i la, ge, o, bo h of he g, o a, e la, ge. Hence, we in , od ce he no ion of den i y on axi al co le e bi a, i e bg, a h .

Definition 2. [illegible] $H = \langle V_1 \cup V_2, E\rangle$ [illegible] G [illegible] (m, n) [illegible] $|V_1|$ [illegible] $|V_2|$ [illegible] m [illegible] n

A co le e bi a, i e bg, a h $H = \langle V_1 \cup V_2, E\rangle$ of G ch ha $\beta^G(V_1) = V_2$ and $\beta^G(V_2) = V_1$ i axi al in he en e ha he, e i no o he, co le e bi a, i e bg, a h $H' = \langle V_1' \cup V_2', E'\rangle$ of G wi h $V_1 \subset V_1'$ and $V_2 \subset V_2'$ ch ha $\beta^G(V_1') = V_2'$ and $\beta^G(V_2') = V_1'$. To a , ecia e hi no ion of axi ali y, we , ove he , o o i ion below.

Proposition 1. [illegible] $H = \langle V_1 \cup V_2, E\rangle$ [illegible] $H' = \langle V_1' \cup V_2', E'\rangle$ [illegible] G [illegible] $V_1 \subseteq V_1'$ [illegible] $V_2 \subseteq V_2'$ [illegible] $H = H'$

[illegible] $H = \langle V_1 \cup V_2, E\rangle$ [illegible] $H' = \langle V_1' \cup V_2', E'\rangle$ [illegible] G [illegible] $V_1 \subseteq V_1'$ [illegible] $V_2 \subseteq V_2'$ [illegible] $V_1 \subseteq V_1'$ [illegible] $V_2 \subseteq V_2'$ [illegible] $\beta^G(V_1') \subseteq \beta^G(V_1)$ [illegible] $\beta^G(V_2') \subseteq \beta^G(V_2)$ [illegible] $V_2' = \beta^G(V_1') \subseteq \beta^G(V_1) = V_2$ [illegible] $V_1' = \beta^G(V_2') \subseteq \beta^G(V_2) = V_1$ [illegible] $E = V_1 \times V_2 = V_1' \times V_2' = E'$ [illegible] $H = H'$ [illegible]

3 Closed Patterns of an Adjacency Matrix

The adjacency a , ix of a g, a h i i o, an in hi dy. Le G be a g, a h wi h $V^G = \{v_1, v_2, \ldots, v_p\}$. The **adjacency matrix** $\mathbf{A}$ of G i he $p \times p$ a , ix de ned by

$$\mathbf{A}[i, j] = \begin{cases} 1 \text{ if } (v_i, v_j) \in E^G \\ 0 \text{ o he, wi e} \end{cases}$$

Recall ha o , g, a h do no have elf-loo and a, e ndi, ec ed. Th $\mathbf{A}$ i a y e , ic a , ix and eve, y en , y on he ain diagonal i 0. Al o, $\{v_j \mid \mathbf{A}[j, k] = 1, 1 \le j \le p\} = \beta^G(v_k) = \{v_j \mid \mathbf{A}[j, k] = 1, 1 \le j \le p\}$.

The adjacency matrix of a graph can be interpreted in a **transactional database** (DB) [1]. To define a DB, we define a **transaction**. Let I be a set of **items**. Then a transaction is defined as a subset of I. For example, a set I of items all items in a set, a transaction by a customer is the items that he/she bought. A DB is a non-empty multi-set of transactions. Each transaction T in a DB is assigned a unique identity $id(T)$. A **pattern** is defined as a non-empty set[1] of items of I. A pattern may be or may not be contained in a transaction. Given a DB and a pattern P, the number of transactions in DB containing P is called the **support** of P, denoted $sup^{DB}(P)$. We assume often interested in patterns that occur sufficiently frequently in a DB. Those patterns are called **frequent** patterns — that is, patterns P satisfying $sup^{DB}(P) \geq ms$, for a threshold $ms > 0$. In this paper, unless mentioned otherwise, we consider all and only those patterns with non-zero occurrences, namely all those frequent patterns with the occurrences $ms = 1$. So, by a pattern of a DB, we mean that it is non-empty and it occurs in DB at least once.

Let G be a graph with $V^G = \{v_1, v_2, \ldots, v_p\}$. If each vertex in V^G is defined as an item, then the neighborhood $\beta^G(v_i)$ of v_i is a transaction. Thus,

$$\{\beta^G(v_1), \beta^G(v_2), \ldots, \beta^G(v_p)\}$$

is a DB. Such a special DB is denoted by DB_G. The identity of a transaction in DB_G is defined as the vertex itself— that is, $id(\beta^G(v_i)) = v_i$. Note that DB_G has the same patterns, of items and transaction. Note also that $v_i \notin \beta^G(v_i)$ since we assume G to be an undirected graph with self-loops.

DB_G can be represented as a binary pattern matrix. This binary matrix $\mathbf{B}$ is defined by

$$\mathbf{B}[i,j] = \begin{cases} 1 \text{ if } v_j \in \beta^G(v_i) \\ 0 \text{ otherwise} \end{cases}$$

Since $v_j \in \beta^G(v_i)$ iff $(v_i, v_j) \in E^G$, it can be seen that $\mathbf{A} = \mathbf{B}$. So, "a pattern of DB_G" is equivalent to "a pattern of the adjacency matrix of G".

Closed patterns are a very type of interesting patterns in a DB. In the last few years, the problem of efficiently mining closed patterns for a large DB has attracted a lot of research interest in the data mining community [2,6,7,13,16,17,19]. Let I be a set of items, and D be a transactional database defined on I. For a pattern $P \subseteq I$, let $f^D(P) = \{T \in D \mid P \subseteq T\}$— that is, $f^D(P)$ are all transactions in D containing the pattern P. For a set of transactions $D' \subseteq D$, let $g(D') = \bigcap_{T \in D'} T = \bigcap D'$— that is, the set of items which are shared by all transactions in D'. Using these two functions, we can define the notion of **closed patterns**. For a pattern P, $CL^D(P) = g(f^D(P))$ is called the **closure** of P. A pattern P is said to be **closed** with respect to a transactional database D iff $CL^D(P) = P$.

We define the **occurrence set** of a pattern P in DB as $occ^{DB}(P) = \{id(T) \mid T \in DB, P \subseteq T\} = \{id(T) \mid T \in f^{DB}(P)\}$. It is straightforward to see that

[1] The $\emptyset$ is usually defined as a valid pattern in the data mining community. However, in this paper, to be consistent to the definition of $\beta^G(X)$, it is excluded.

$id(T) \in occ^{DB}(P)$ iff $T \in f^{DB}(P)$. The e i a igh connec ion be ween he no ion of neighbo hood in a g a h G and occ ence in he co e onding an ac ional da aba e DB_G.

Proposition 2. [illegible] G [illegible] P [illegible] DB_G [illegible] $occ^{DB_G}(P) = \beta^G(P)$

[illegible] $v \in occ(P)$, [illegible] v [illegible] P [illegible] $v \in \beta(v')$ [illegible] $v' \in P$ [illegible] $v \in \bigcap_{v' \in P} \beta(v') = \beta(P)$.
[illegible] $u \in \beta(P)$, [illegible] u [illegible] P [illegible] $\beta(u) \supseteq P$ [illegible] $\beta(u)$ [illegible] DB_G [illegible] P [illegible] $u \in occ(P)$

The e i al o a nice connec ion be ween he no ion of neighbo hood in a g a h and ha of clo e of a e n in he co e onding an ac ional da aba e.

Proposition 3. [illegible] G [illegible] P [illegible] DB_G [illegible] $\beta^G(\beta^G(P)) = CL^{DB_G}(P)$ [illegible] $\beta^G \circ \beta^G$ [illegible] DB_G

[illegible] $\beta(\beta(P)) = \beta(occ(P)) = \bigcap_{id(T) \in occ(P)} T = \bigcap_{T \in f(P)} T = g(f(P)) = CL(P)$

We di c in he nex ec ion dee e ela ion hi be ween he clo ed a - e n of DB_G and he axi al co le e bi a i e bg a h of G.

4 Results

The occ ence e of a clo ed a e n C in DB_G lay a ey ole in he ax- i al co le e bi a i e bg a h of G. We in od ce below o e of i ey o e ie .

Proposition 4. [illegible] G [illegible] C_1 [illegible] C_2 [illegible] DB_G [illegible] $C_1 = C_2$ [illegible] $occ^{DB_G}(C_1) = occ^{DB_G}(C_2)$

[illegible] $occ(C_1) = occ(C_2)$ [illegible] $id(T) \in occ(P)$ [illegible] $T \in f(P)$ [illegible] $f(C_1) = f(C_2)$ [illegible] $occ(C_1) = occ(C_2)$ [illegible] C_1 [illegible] C_2 [illegible] DB_G [illegible] $C_1 = g(f(C_1)) = g(f(C_2)) = C_2$ [illegible]

Proposition 5. [illegible] G [illegible] C [illegible] DB_G [illegible] C [illegible] $occ^{DB_G}(C) \cap C = \{\}$

[illegible] $v \in occ(C)$ [illegible] v [illegible] C [illegible] G [illegible] $v \notin C$ [illegible] $occ^{DB_G}(C) \cap C = \{\}$

In fac hi o o i ion hold fo any a e n P, no nece a ily a clo ed a e n C.

Lemma 1. [illegible] G [illegible] C [illegible] DB_G [illegible] $f^{DB_G}(occ^{DB_G}(C)) = \{\beta^G(c) \mid c \in C\}$.

[illegible] C [illegible] $\{c \mid c \in C\}$ [illegible] DB_G [illegible] C [illegible] $\{c \mid c \in C\}$ [illegible] G [illegible] $occ(C)$ [illegible] $\{\beta(c) \mid c \in C\}$ [illegible] $occ(C)$ [illegible] $f(occ(C)) = \{\beta(c) \mid c \in C\}$

Proposition 6. [illegible] G [illegible] C [illegible] DB_G [illegible] $occ^{DB_G}(C)$ [illegible] DB_G

[illegible] $f(occ(C)) = \{\beta(c) \mid c \in C\}$ [illegible] $CL(occ(C)) = g(f(occ(C))) = \bigcap f(occ(C)) = \bigcap_{c \in C} \beta(c) = \beta(C)$ [illegible] $\beta(C) = occ(C)$ [illegible] $occ(C)$ [illegible]

The h ee o o i ion above give i e o a co le of in e e ing co olla ie below.

Corollary 1. [illegible] G [illegible] DB_G [illegible]

[illegible] n [illegible] ([illegible]) [illegible] DB_G [illegible] C_1, C_2, [illegible] C_n [illegible] $occ(C_1)$, $occ(C_2)$, [illegible] $occ(C_n)$ [illegible] DB_G [illegible] $occ(C_i)$ [illegible] $occ(C_j)$ [illegible] C_i [illegible] C_j [illegible] $occ(\cdot)$ [illegible] n [illegible]

Corollary 2. [illegible] G [illegible] C [illegible] C [illegible] $occ^{DB_G}(C)$ [illegible] ms [illegible] DB_G [illegible]

[illegible] C [illegible] DB_G [illegible] $occ^{DB_G}(C)$ [illegible] C [illegible] $occ^{DB_G}(C)$ [illegible] C [illegible] $occ^{DB_G}(C)$ [illegible] ms [illegible] DB_G [illegible] C [illegible] C [illegible] $occ^{DB_G}(C)$ [illegible] ms [illegible] DB_G [illegible]

No e ha hi co olla y doe no i ly he n be of f e en clo ed a e n ha a ea a lea ms i e in DB_G i alway even. A co n e exa le i given below.

[illegible] Con ide a DB_G given by he following a ix:

	p_1	p_2	p_3	p_4	p_5
$\beta(p_1)$	0	1	1	0	0
$\beta(p_2)$	1	0	1	1	1
$\beta(p_3)$	1	1	0	1	1
$\beta(p_4)$	0	1	1	0	0
$\beta(p_5)$	0	1	1	0	0

We li i clo ed a e n , hei o , and hei $occ(\cdot)$ co n e a a e n below:

o of X	clo e a e n X	$Y = occ(X)$	o of Y
3	$\{p_2, p_3\}$	$\{p_1, p_4, p_5\}$	2
4	$\{p_2\}$	$\{p_1, p_3, p_4, p_5\}$	1
4	$\{p_3\}$	$\{p_1, p_2, p_4, p_5\}$	1

S o e we a e $ms = 3$. Then he e a e only 3 clo ed a e n —an odd n be — ha occ a lea ms i e , vi . $\{p_2, p_3\}$, $\{p_2\}$, and $\{p_3\}$.

Finally, we de on a e o ain e l on he ela ion hi wi h clo ed a - e n and axi al co le e bi a i e bg a h . In a ic la , we di cove ha eve y ai of a clo ed a e n C and i occ ence e $occ^{DB_G}(C)$ yield a di inc axi al co le e bi a i e bg a h of G.

Theorem 1. . G [illegible] C [illegible] DB_G [illegible]

$$H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C) \rangle$$

[illegible] G [illegible] C [illegible] C [illegible] DB_G [illegible] $occ(C)$ [illegible] $C \cap occ^{DB_G}(C) = \{\}$ [illegible] $\forall v \in occ(C)$, v [illegible] G [illegible] C [illegible] $C \times occ(C) \subseteq E^G$, [illegible] H [illegible] C [illegible] $occ(C)$ [illegible] H [illegible] G [illegible] $occ^{DB_G}(C) = \beta^G(C)$ [illegible] $C = \beta^G(\beta^G(C))$ [illegible] $C = \beta^G(occ^{DB_G}(C))$ H [illegible]

Theorem 2. . G [illegible] $H = \langle V_1 \cup V_2, E \rangle$ [illegible] G [illegible] V_1 V_2 [illegible] DB_G, $occ^{DB_G}(V_1) = V_2$ $occ^{DB_G}(V_2) = V_1$

[illegible] H [illegible] G [illegible] $\beta(V_1) = V_2$ $\beta(V_2) = V_1$ [illegible] $CL(V_1) = \beta(\beta(V_1)) = \beta(V_2) = V_1$ V_1 [illegible] V_2 [illegible] $occ(V_1) = \beta(V_1) = V_2$ $occ(V_2) = \beta(V_2) = V_1$ [illegible]

The above wo heo e ay ha axi al co le e bi a i e bg a h of G a e all in he fo of $H = \langle V_1 \cup V_2, E \rangle$, whe e V_1 and V_2 a e bo h a clo ed a e n of DB_G. Al o, fo eve y clo ed a e n C of DB_G, he g a h $H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C) \rangle$ i a axi al co le e bi a i e bg a h of G. So, he e i a one- o-one co e ondence be ween axi al co le e bi a i e bg a h and clo ed a e n ai .

We can al o de ive a co olla y lin ing o h e hold of DB_G o he den i y of axi al co le e bi a i e bg a h of G.

Corollary 3. [illegible] G [illegible]

$$H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C) \rangle$$

[illegible] (m, n) [illegible] G [illegible] C [illegible] C [illegible] m [illegible] DB_G [illegible] $occ^{DB_G}(C)$ [illegible] n [illegible] DB_G

The co[illegible]olla[illegible]y above ha[illegible] he following i[illegible]o[illegible] an i[illegible]lica[illegible]ion.

Theorem 3. [illegible] G [illegible]

$$H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C) \rangle$$

[illegible] (m, n) [illegible] G [illegible] C [illegible] C [illegible] m [illegible] DB_G [illegible] $|C| \geq n$

[illegible] $H = \langle C \cup occ^{DB_G}(C), C \times occ^{DB_G}(C) \rangle$ [illegible] (m, n) [illegible] G [illegible] $C = occ(occ(C))$ [illegible] $occ(\cdot)$, $sup(occ(C)) = |occ(occ(C))| = |C|$ [illegible] H [illegible] (m, n) [illegible] G [illegible] C [illegible] C [illegible] m [illegible] DB_G [illegible] $|C| \geq n$ [illegible]

Theo[illegible]e [illegible] 1 and 2 [illegible] how ha[illegible] algo[illegible]i h [illegible] fo[illegible] ining clo[illegible]ed [illegible] a[illegible]e[illegible]n can be [illegible]ed o ex[illegible]ac[illegible] axi[illegible] al co[illegible] le e bi[illegible] a[illegible] i e[illegible] bg[illegible]a h of ndi[illegible]ec ed g[illegible]a h wi h-o[illegible] elf-loo[illegible]. S[illegible]ch da a[illegible] ining algo[illegible]i h [illegible] a[illegible]e[illegible] ally[illegible] igni[illegible] can ly[illegible] o[illegible]e e cien a[illegible] highe[illegible] [illegible] o[illegible] h[illegible]e hold ms. Th[illegible] Theo[illegible]e [illegible] 3[illegible] gge[illegible] an i[illegible] o[illegible] an o i-[illegible] i a ion fo[illegible] ining (m, n)-den e[illegible] axi[illegible] al co[illegible] le e bi[illegible] a[illegible] i e[illegible] bg[illegible]a h . To wi[illegible], a[illegible] ing $m > n$, i[illegible] ce o[illegible] ine clo[illegible]ed [illegible] a[illegible]e[illegible]n a[illegible] o[illegible] h[illegible]e hold $ms = m$, and hen ge[illegible] he an we[illegible] by[illegible] l e[illegible]ing o[illegible] ho e[illegible] a[illegible]e[illegible]n of leng h le[illegible] han n.

5 Experimental Results

We[illegible]e an exa[illegible]le o de on[illegible]a e he[illegible] eed of li ing all[illegible] axi[illegible] al co[illegible] le e bi[illegible] a[illegible] i e[illegible] bg[illegible]a h by[illegible] ing an algo[illegible]i h fo[illegible] ining clo[illegible]ed [illegible] a[illegible]e[illegible]n . The g[illegible]a h i[illegible] a[illegible]o ein in e[illegible]ac ion ne wo[illegible] wi h[illegible]o ein a[illegible] ve[illegible]ice and in e[illegible]ac ion a edge . A[illegible] he[illegible]e a[illegible]e[illegible] any[illegible] hy[illegible]ical[illegible]o ein in e[illegible]ac ion ne wo[illegible] co[illegible]e onding o diffe[illegible]en[illegible] ecie , he[illegible]e we a e he[illegible] le and[illegible] o co[illegible]ehen ive yea[illegible] hy[illegible]ical and gene ic in e[illegible]ac ion ne wo[illegible] [3] a[illegible] an exa[illegible] le. Thi[illegible] g[illegible]a h con i[illegible] of 4904 ve[illegible]ice and 17440 edge[illegible] (af e[illegible]e oving 185[illegible] elf loo[illegible] and 1413[illegible]ed ndan edge[illegible] f[illegible]o[illegible] he o[illegible]iginal 19038 in e[illegible]ac ion). The[illegible]efo[illegible]e, he adjacency[illegible] a[illegible]ix i a[illegible]an ac ional da aba e wi h 4904 i e[illegible] and 4904[illegible]an ac ion . On ave[illegible]age, he n[illegible] be[illegible] of i e[illegible] in a[illegible]an ac ion i[illegible] 3.56. Tha[illegible] i , he ave[illegible]age[illegible] i e of he neighbo[illegible] hood of a[illegible]o ein i[illegible] 3.56.

We[illegible]e FPclo[illegible]e* [7], a[illegible] a e-of- he-a[illegible] algo[illegible]i h fo[illegible] ining clo[illegible]ed [illegible] a[illegible]e[illegible]n , fo[illegible]en[illegible] e[illegible]a ing he[illegible] axi[illegible] al co[illegible] le e bi[illegible] a[illegible] i e[illegible] bg[illegible]a h . O[illegible] achine i a PC

wi h a CPU cloc , a e 3.2GH and 2GB of e o y. The e l a e e o ed in Table 1, whe e he econd col n how he o al n be of **frequent** clo e a - e n who e o level i a lea he h e hold n be in he col n one. The hi d col n of hi able how he n be of clo e a e n who e ca dinali y and o a e bo h a lea he o h e hold; all ch clo ed a e n a e e ed ali ed clo ed a e n . Only he e ali ed clo ed a e n can be ed o fo a axi al co le e bi a i e bg a h $H = \langle V_1 \cup V_2, E \rangle$ ch ha bo h of $|V_1|$ and $|V_2|$ ee he h e hold . F o he able, we can ee:

- The n be of all clo ed a e n (co e onding o ho e wi h he o h e hold of 1) i even. Mo eove , he n be of ali ed clo e a e n wi h ca dinali y no le han any o level i al o even, a ex ec ed f o Co olla y 2.
- The algo i h n fa —The algo i h og a can co le e wi hin 4 econd fo all i a ion e o ed he e. Thi indica e ha en e a ing all axi al co le e bi a i e bg a h f o a la ge g a h can be ac ically olved by ing algo i h fo ining clo ed a e n .
- A o-called " any-few" o e y [11] of o ein in e ac ion i ob e ved again in o ex e i en e l . The " any-few" o e y ay ha : a o ein ha in e ac wi h a la ge n be of o ein end . o in e ac wi h ano he o ein which al o in e ac wi h a la ge n be of o ein [11]. In o he wo d , highly connec ed o ein a e e a a ed by low-connec ed o ein . Thi i o clea ly een in Table 1 a he highe o h e hold . Fo exa le, a he o h e hold 11, he e a e 12402 o ein g o ha have f ll in e ac ion wi h a lea 11 o ein . B he e a e only wo g o , a een in he hi d col n of he able, ha each con ain a lea 11 o ein and ha have f ll al in e ac ion.

Table 1. Close patterns in a yeast protein interaction network

support threshold	# of frequent close patterns	# of qualified close patterns	time in sec.
1	121314	121314	3.859
2	117895	114554	2.734
3	105854	95920	2.187
4	94781	80306	1.765
5	81708	60038	1.312
6	66429	36478	0.937
7	50506	15800	0.625
8	36223	3716	0.398
9	25147	406	0.281
10	17426	34	0.171
11	12402	2	0.109
12	9138	0	0.078

6 Discussion and Conclusion

The e a e wo ecen e ea ch e l ela ed o o wo . The oble of en - e a ing all axi al co le e bi a i e bg a h (called axi al bi a i e cli e he e) f o a [illegible] ha been inve iga ed by [10]. The diffe - ence i ha o wo i o en e a e all he bg a h f o any g a h (wi ho elf loo and ndi ec ed), b Ma ino and Uno' wo i li i ed o en e a ing f o only bi a i e g a h . So, o e hod i o e gene al. Za i [20] ob e ved ha a an ac ional da aba e *DB* can be e e en ed by a bi a i e g a h *H*, and al o a ela ion ha clo ed a e n (w ongly a ed a axi al a e n in [20]) of *DB* one- o-one co e ond o axi al co le e bi a i e bg a h (called axi al bi a i e cli e he e) of *H*. Howeve , o wo i o conve a g a h *G*, incl ding bi a i e g a h , in o a ecial an ac ional da aba e DB_G, and hen o di cove all clo ed a e n f o DB_G fo en e a ing all axi al co le e bi a i e bg a h of *G*. F he o e, he occ ence e of a clo ed a e n in Za i' wo ay no be a clo ed a e n, b ha of o i alway a clo ed a e n.

Finally, le ' a i e he e l achieved in hi a e . We have died he oble of li ing all axi al co le e bi a i e bg a h f o a g a h. We oved ha hi oble i e ivalen o he ining of all clo ed a e n f o he adjacency a ix of hi g a h. Ex e i en al e l on a la ge o ein in e ac ion ' da a how ha a da a ining algo i h can n ve y fa o nd all in e ac ed o ein g o . The e l will have g ea o en ial in a lica ion ch a in web ining, in co nica ion y e , and in biological eld .

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM-SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., May 1993. ACM Press.
2. Yves Bastide, Nicolas Pasquier, Rafik Taouil, Gerd Stumme, and Lotfi Lakhal. Mining minimal non-redundant association rules using frequent closed itemsets. *Computational Logic*, pages 972–986, 2000.
3. B. J. Breitkreutz, C. Stark, and M. Tyers. The grid: The general repository for interaction datasets. *Genome Biology*, 4(3):R23, 2003.
4. Andrei Z. Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet L. Wiener. Graph structure in the web. *Computer Networks*, 33(1-6):309–320, 2000.
5. David Eppstein. Arboricity and bipartite subgraph listing algorithms. *Information Processing Letters*, 51:207–211, 1994.
6. Bart Goethals and Mohanned J. Zaki. FIMI'03: Workshop on frequent itemset mining implementations. In *Third IEEE International Conference on Data Mining Workshop on Frequent Itemset Mining Implementations*, pages 1–13, 2003.
7. Gosta Grahne and Jianfei Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *Proceedings of FIMI'03: Workshop on Frequent Itemset Mining Implementations*, 2003.

8. Jonathan L. Gross and Jay Yellen. *Handbook of Graph Theory*. CRC Press, 2004.
9. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. *Computer Networks*, 31(11-16):1481–1493, 1999.
10. Kazuhisa Makino and Takeaki Uno. New algorithms for enumerating all maximal cliques. In *Proceedings of the 9th Scandinavian Workshop on Algorithm Theory (SWAT 2004)*, pages 260–272. Springer-Verlag, 2004.
11. Sergei Maslov and Kim Sneppen. Specificity and stability in topology of protein networks. *Science*, 296:910–913, 2002.
12. Tsuyoshi Murata. Discovery of user communities from web audience measurement data. In *Proceedings of The 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004)*, pages 673–676, 2004.
13. Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory (ICDT)*, pages 398–416, 1999.
14. D. J. Reiss and B. Schwikowski. Predicting protein-peptide interactions via a network-based motif sampler. *Bioinformatics (ISMB 2004 Proceedings)*, 20 (suppl.):i274–i282, 2004.
15. A. H. Tong, B. Drees, G Nardelli, G. D. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, S. Ferracuti, B. Nelson, S. Paoluzi, M. Quondam, A. Zucconi, C. W. Hogue, S. Fields, C. Boone, and G. Cesareni. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, 295:321–324, 2002.
16. Takiake Uno, Masashi Kiyomi, and Hiroaki Arimura. LCM ver.2: Efficient mining algorithms for frequent/closed/maximal itemsets. In *IEEE ICDM'04 Workshop FIMI'04 (International Conference on Data Mining, Frequent Itemset Mining Implementations)*, 2004.
17. J. Wang, Jiawei Han, and Jian Pei. CLOSET+: Searching for the best strategies for mining frequent closed itemsets. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), Washington, DC, USA*, pages 236–245, 2003.
18. Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. *SIGKDD Explorations*, 5(1):59–68, 2003.
19. Mohammed Javeed Zaki and Ching-Jiu Hsiao. CHARM: An efficient algorithm for closed itemset mining. In *Proceedings of the Second SIAM International Conference on Data Mining*, 2002.
20. Mohammed Javeed Zaki and Mitsunori Ogihara. Theoretical foundations of association rules. In *Proc. 3rd SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 1998.

Improving Generalization by Data Categorization

Li g Li, A 1 P a a , H a -Tie Li , a d Ya e S. Ab -M afa

Learning Systems Group, California Institute of Technology, USA

Abstract. In most of the learning algorithms, examples in the training set are treated equally. Some examples, however, carry more reliable or critical information about the target than the others, and some may carry wrong information. According to their intrinsic margin, examples can be grouped into three categories: typical, critical, and noisy. We propose three methods, namely the selection cost, SVM confidence margin, and AdaBoost data weight, to automatically group training examples into these three categories. Experimental results on artificial datasets show that, although the three methods have quite different nature, they give similar and reasonable categorization. Results with real-world datasets further demonstrate that treating the three data categories differently in learning can improve generalization.

1 Introduction

Machi e ea 1 g 1 a a e a 1 e a ach e de 1g . I ead f he c e 1 a a f a he a 1ca de 1 g he e , he e f ea 1 g 1 a e a da a e f e a e , ch a 1 - a1 f a a a ge f c 1 , a d he 1 e a h he 1 ha be a 1 a e he a ge . The da a e , ac 1 g a he 1 f a 1 ga e a be ee he a ge a d he h he 1 , 1 h a he hea f he ea 1 g ce .

Ge e a , e e e a e 1 he da a e 1 ea ed e a a d e a e 1 e 1c1 d1 ca ded. Af e a , each e a e ca 1e 1 1ece f 1 f a 1 ab he a ge . H e e , 1f e f he e a e a e c ed 1 h 1 e, he 1 f a 1 he 1de d be 1 ead1 g. I h1 ca e, 1 1 be e 1de 1f a d e e he , h1ch ca be e f ed e1 he e 1c1 b a - 1e de ec 1 e ce 1 g e [1], 1 1c1 1 he ea 1 g a g 1 h 1a eg a 1 a 1 [2].

E e 1 ca e he e a he e a e a e 1 e e , he e a e 1 a 1 1 h1ch e a dea 1 h e a e d1 e e . F 1 a ce, 1 a ge da a e h1ch f e c a1 ed da e a e , a be f 1 f a 1 e e a e ha ca 1e f he 1 f a 1 1 a e de 1 ab e f c a 1 a ea- [3]. I ca e he e e f he h he e ca e fec de he a ge , 1 1 be e d1 ca d e a e ha ca be c a 1 ed c ec b a h- he 1 a he a c f e he ea 1 g [4].

M e 1 1 g e h d ha ea e a e d1 e e e d g e a - e 1 ca eg 1e ba ed d1 e e c 1 e 1a: c 1 e 1 c 1 e (1e) [1], ea ha d [5], 1ca 1 f a 1 e [3], e c. I h1 a e ,

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 157–168, 2005.

e 1 d ce he c ce f a a c 1 e 1 f g 1 g da a a d 1 a e he eed ha e h ee ca eg 1e 1 ead f . We e e h ee e h d a a e he ca eg 1 1 g. We h ha b ea 1 g e a e 1 he e h ee ca eg 1e d1 e e , e ca 1 e he ge e a 1 a 1 e f a ce f ea 1 g ea - d b e . I addı 1 , he ca eg 1 a 1 ca be ed ed ce he da a e 1 e 1 h a ec 1 g he ea 1 g e f a ce.

The a e 1 ga 1 ed a f . The f a f a e f ea 1 g 1 de ed 1 Sec . 2. The 1 Sec . 3, e 1 d ce he c ce f da a ca eg 1 a 1 a d e e e h d f a a 1c ca eg 1 a 1 . Re a 1 cıa a d ea - d da a e a e e ed 1 Sec . 4 a d 5. We a c c de 1 Sec . 6.

2 Learning Systems

I ea 1 g b e , a e 1 he f f 1 - a1 $(\mathbf{x}, y)$. We a e ha he 1 ec $\mathbf{x} \in \mathcal{X}$ a e ge e a ed 1 de e de f a babı 1 dı ıb 1 $P_{\mathcal{X}}$, a d he abe $y \in \mathcal{Y}$ a e c ed f $y = f(\mathbf{x})$. He e he f c 1 $f\colon \mathcal{X} \to \mathcal{Y}$ 1 ca ed he . I hı a e, e ha f c bı a c a 1 ca 1 b e , 1 hıch $\mathcal{Y} = \{-1, 1\}$. We f he a e ha f c e f h e h dı g a $f_r\colon \mathcal{X} \to \mathbb{R}$, 1.e., $f(\mathbf{x}) = \text{ıg}\,(f_r(\mathbf{x}))$, he e he ag 1 de f $f_r(\mathbf{x})$ c e d he e ıabı 1 f he $f(\mathbf{x})$. F e a e, ıf he a ge f c 1 $f(\mathbf{x})$ 1 dıca e he he a c edı ca d h d be 1 ed a e $\mathbf{x}$, he 1 1 ıc f c 1 $f_r(\mathbf{x})$ c d be he a ıg ed c edı c e f he e .

F a h he 1 $g\colon \mathcal{X} \to \mathcal{Y}$ a d a e a e $(\mathbf{x}, y)$, a c ed e ea e (f c 1) 1

$$e(g(\mathbf{x}), y) = [g(\mathbf{x}) \neq y]\,,$$

he e he B ea e $[\cdot]$ 1 1 ıf he c dı 1 1 e a d 0 he 1 e. The , f a a ge f c 1 f, e ca de e he f g a

$$\pi(g) = \mathrm{E}_{\mathbf{x} \sim P_{\mathcal{X}}}\left[e(g(\mathbf{x}), f(\mathbf{x}))\right].$$

The g a f ea 1 g 1 h ch e a h he 1 g ha ha a - f- a e e $\pi(g)$ f a e f ca dıda e h he e , a e he $\mathcal{G}$.

H e e , $\pi(g)$ ca be dı ec c ed beca e he dı ıb 1 $P_{\mathcal{X}}$ a d he a ge f c 1 f a e b h . The 1 f a 1 e ca acce 1 f e 1 1 ed 1 he $\mathcal{D}$, hıch c 1 f N e a e $(\mathbf{x}_i, y_i)$, $i = 1..N$. Th , 1 ead f 1 g f a h he 1 g 1 h $\pi(g)$ a e , a ea 1 g a g 1 h a d g ha 1 1 1 e a e 1 a f $\pi(g)$. A c ed e 1 a 1 he $\nu(g)$ he a 1 1 g e $\mathcal{D}$,

$$\nu(g) = \nu(g, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} e\left(g(\mathbf{x}_i), y_i\right).$$

F a ed h he 1 g, $\nu(g)$ 1 a bıa ed e 1 a f $\pi(g)$, a d he he 1 e f $\mathcal{D}$ 1 a ge e gh, a 1 ıca b d g a a ee ha $\nu(g)$ a d $\pi(g)$ d dı e b ch.

N e ha he ea 1 g a g 1 h ea che he h e ea 1 g de $\mathcal{G}$ f a 1 ab e h he 1 a he ha f c 1 g a ed e. I hı ca e, he ba-bı 1 ha $\nu(g)$ a d $\pi(g)$ dı e f e $g \in \mathcal{G}$ ge ag 1 ed b he c e 1 f $\mathcal{G}$. Th , he h he 1 f d ıgh he aı 1 g e e hı e 1 ha 1 g a hıgh - f- a e e [2]. Thı 1 a 1 1 ca ed hıch a 1 e he g d 1 - a e edıc 1 d e a e g d - f- a e edıc 1 . The 1 a 1 ca bec e e he he e a e c aı 1 e. The , 1 g he aı 1 g e e ea 1 g he g 1 f a 1 , hıch ead bad - f- a e edıc 1 .

Lea 1 g a g 1 h f e a ıd e 1 g h gh eg a 1 a 1 [2]. Reg a 1 a 1 a e f ce a ade- be ee he c e 1 f $\mathcal{G}$ a d he ece 1 edıc he aı 1 g e a e c ec . If e ca cha ac e 1 e he ef e f each aı 1 g e a e, he ea 1 g a g 1 h ca he be g ıded f c edıc 1 g 1 a e a e c ec , eadı g a e ea 1 gf eg a 1 a 1 ade- a d h a be e ge e a 1 a 1 e f a ce. Thı 1 a e ca eg 1 e he aı 1 g e a e .

3 Data Categorization

The e f da a ca eg 1 a 1 1 g e a e acc dı g hei e-f e ea 1 g ha 1 1 ıb e ea he dı e e . G e a . [3] g ed da a 1 ca eg ıe , ıca a d 1 f a 1 e. H e e, he f d ha he ca eg f 1 f a 1 e e a e c aı ed b h ef e a e a d 1 e . Th , he eeded h a -ba ed - ce 1 g e 1 1 a e he 1 e a e . Sı 1 a b e a e e c e ed 1 he e h d ha e -g ca eg 1 a 1 . Thı h ha e eed ha e e ha ca e-g ıe . I hı a e, e he eed b ha 1 g h ee ca eg ıe : ıca , c 1 ıca a d 1 .

A h gh a e a e ca 1 f a 1 ab he a ge , he a e e a 1 he e e ha e e a e ca e ef 1 f a 1 ab he a ge ha he , a d e e a e a 1 g ıde he ea 1 g a g 1 h . F 1 - a ce, 1 c a 1 ca 1 b e , a e a e c e he c a b da gı e e c 1 ıca 1 f a 1 ha a e a e dee 1 he c a e 1 . I addı-1 , ea - d da a f e c aı 1 abe ed e a e , hıch c ıe he abı 1 f he 1 - a e e a 1 a e he - f- a e e a d ead bad ge e a 1 a 1 .

O e a ca eg 1 e e a e ba ed he ab e 1 1 1 h gh he c ce f F a e a e $(\mathbf{x}, y)$, 1 1 1 ıc a gı 1 $yf_r(\mathbf{x})$, he e f_r 1 he 1 ıcı 1 1 ıc f c 1 de ed 1 Sec . 2. U de e ea ab e h e a 1 , he 1 1 ıc a gı ca be ea ed a a ea e f h c e he e a e 1 he c a 1 ca 1 decı 1 b da . If he 1 1 ıc a gı 1 a 1 1 e, he e a e ıe ea he decı 1 b da a d h d be ca eg 1 ed a c 1 ıca . If he a gı 1 a ge 1 1 e, he e a e 1 fa f he b da a d h d be ca eg 1 ed a ıca . E a e 1 h ega 1 e 1 1 ıc a gı a e 1 abe ed, a d h d be c a 1 ed a 1 . Th ,

e a e h e h d , 0 a d e a 1 1 e a e, a 1 1 he 1 1 1c a gl a d ca eg 1 e he da a.

I ac 1ca 1 a 1 , 1 1 1 1b e ca c a e he 1 1 1c a gl e he 1 1 1c f c 1 1 . H e e, 1 ce e a e 1 e e ed 1 h e h- d1 g he 1 1 1c a gl , a 1c f c 1 f he 1 1 1c a gl ca be ed 1 h a 1a e h e h d . Ne , e e h ee d1 e e e h d e 1 a e ch f c 1 f a a 1ca ca eg 1 1 g he da a.

3.1 Selection Cost

Bad ge e a 1 a 1 a 1 e he he 1 - a e e 1 a bad 1 d1ca f he - f- a e e . A a 1c a e a e $(\mathbf{x}, y)$ a de e 1 a e he ge e a 1 a 1 e f a ce 1f 1 e 1 a bad 1 d1ca f he - f- a e e . Ba ed h1 1 1 1 , N1ch [4] gge ed e he c e a 1 c e c1e be ee $e(g(\mathbf{x}), y)$ a d $\pi(g)$ de a 1 d1 1b 1 $P_{\mathcal{G}}$ f g,

$$\begin{aligned}\rho(\mathbf{x}, y) &= \mathrm{c}\ \mathrm{c}\ \mathrm{ef}_g\left[e(g(\mathbf{x}), y), \pi(g)\right] \\ &= \frac{\mathrm{E}_g\left[e(g(\mathbf{x}), y)\pi(g)\right] - \mathrm{E}_g\left[e(g(\mathbf{x}), y)\right]\mathrm{E}_g\left[\pi(g)\right]}{\sqrt{\mathrm{Va}_g\left[e(g(\mathbf{x}), y)\right]\mathrm{Va}_g\left[\pi(g)\right]}},\end{aligned}$$

ea e h e he 1 d1 1d a e e $e(g(\mathbf{x}), y)$ 1 d1ca e $\pi(g)$. A 1 1 e c e a 1 ρ 1 d1ca e ha 1f g ha a e h1 e a e, 1 1 1 e ha e a - f- a e e , . Th1 1 f a 1 ed 1 The e 1.

Theorem 1. [illegible] $\mathcal{G}$ [illegible] ([illegible] $P_{\mathcal{G}}[g] = P_{\mathcal{G}}[-g]$ [illegible] $g \in G$),

$$\rho(\mathbf{x}, y) \propto \mathrm{E}_g\left[\pi(g) \mid g(\mathbf{x}) \neq y\right] - \mathrm{E}_g\left[\pi(g) \mid g(\mathbf{x}) = y\right], \tag{1}$$

[illegible] $\mathcal{G}$

F a gl e e a e $(\mathbf{x}, y)$ a d $P_{\mathcal{G}}$, e $p_i = \mathrm{P}\left[e(g(\mathbf{x}), y) = i\right]$ a d $\pi_i = \mathrm{E}_g\left[\pi(g) \mid e(g(\mathbf{x}), y) = i\right]$ f $i = 0, 1$. We ha e $p_0 + p_1 = 1$, a d

$$\begin{gathered}\mathrm{E}_g\left[e(g(\mathbf{x}), y)\pi(g)\right] = p_1\pi_1, \quad \mathrm{E}_g\left[\pi(g)\right] = p_0\pi_0 + p_1\pi_1, \\ \mathrm{E}_g\left[e(g(\mathbf{x}), y)\right] = p_1, \quad \mathrm{Va}_g\left[e(g(\mathbf{x}), y)\right] = p_0 p_1.\end{gathered}$$

He ce 1 h he de 1 1 f $\rho(\mathbf{x}, y)$,

$$\rho(\mathbf{x}, y) = \frac{p_1\pi_1 - p_1\left(p_0\pi_0 + p_1\pi_1\right)}{\sqrt{\mathrm{Va}_g\left[e(g(\mathbf{x}), y)\right]\mathrm{Va}_g\left[\pi(g)\right]}} = (\pi_1 - \pi_0)\sqrt{\frac{p_0 p_1}{\mathrm{Va}_g\left[\pi(g)\right]}}.$$

Whe $\mathcal{G}$ 1 ega 1 e 1c, 1 1 1 1a ha $p_0 = p_1 = \frac{1}{2}$ f a $(\mathbf{x}, y)$. S he 1 a a 1 1 a c a . □

The c d1 1 a e ec a 1 π_1 (π_0) 1 he e ec ed - f- a e e f h - he e ha ed1c $(\mathbf{x}, y)$ g (c ec). I he ea 1 g ce , e ca

[illegible] e ec h [illegible] he e ha ag ee [illegible] $(\mathbf{x}, y)$ [illegible] h e ha d [illegible]. The d1 e e ce be- ee he c d1 1 a e ec a 1 1 h he e a 1 e cha ge 1 he a e age [illegible] f- a e e [illegible], a d 1 ca ed he [illegible]. If he e e e [illegible] e e - a e ea [illegible], a [illegible] 1 1 e e ec 1 c [illegible] d 1 ha e h d ch [illegible] e a h he 1 ha ag ee 1 h 1. Whe a da a e 1 c ce ed, he e ec 1 c 1 dıca e, a ea a 1 a 1 e, h de 1 ab e 1 1 c a 1f he e a e c - ec. S1 ce he c e a 1 ρ 1 a ca ed e 1 f he e ec 1 c, e 1 e he a e e ec 1 c f b h a 1 1e 1 he f 1 g e.

I ac 1ce, he e ec 1 c f a e a e $(\mathbf{x}_i, y_i)$ 1 1 acce 1b e beca e $\pi(g)$ ca be c ed. H e e, e a e 1 a e $\pi(g)$ b he ea e- e- e [1] $\nu^{(i)}(g) = \nu(g, \mathcal{D} \setminus \{(\mathbf{x}_i, y_i)\})$. The e ec 1 c ca he be e 1 a ed, b a d a 1 g e he ea 1 g de, a he c e a 1 c e c1e be ee $e(g(\mathbf{x}_i), y_i)$ a d $\nu^{(i)}(g)$. Th1 e e e 1 he e e ce f 1 e [4].

N e ha f The e 1 be ea 1 gf, he ac a ea 1 g de ha be ed e 1 a e he e ec 1 c. U de 1 ab e ch 1ce f de c e - 1, h e e, e a 1 g h1 e 1 e e f e d e a ec he e f a ce 1 e e 1 e. I h1 a e, e ha e e a e a de 1 g de he c 1 g he e ec 1 c.

We ca eg 1 e a e a e a 1ca 1f 1 e ec 1 c 1 g ea e ha a h e h d t_c, 1 1f he c 1 e ha a h e h d t_n, a d c 1 1ca 1f he c 1e be ee he h e h d. S1 ce a ega 1 e e ec 1 c 1 1e ha 1 1 be e 1 c a 1f he e a e, e 1 a 1dea h e h d e a a e 1 a d 1 e e e a e. We ch e h e h d a d e, $t_c = 0.15$ a d $t_n = -0.1$, acc da e e 1 a 1 e. Be e ca eg 1 a 1 a be b a1 ed b f he e 1 a 1 g he 1 a h e h d; b f he 1 a 1 e f h1 a e, e e ad h c h e h d1 g f a e h d.

3.2 SVM Confidence Margin

The ec achı e (SVM) [2] 1 a ea 1 g a g 1 h ha d a a ge- c de ce h e a e c a 1 e 1 he fea e ace. S ch c a 1 e 1 a b a1 ed b 1 g he Lag a ge d a b e :

$$\min_{\alpha} \quad \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N} \alpha_i$$

$$\text{s.t.} \quad \sum_{i=1}^{N} y_i \alpha_i = 0, \quad 0 \le \alpha_i \le C.$$

He e he e e $K(\mathbf{x}, \mathbf{x}')$ 1 a 1 e d c f he a f ed $\mathbf{x}$ a d $\mathbf{x}'$ 1 he fea e ace, a d C 1 he eg a 1 a 1 a a e e. SVM edıc he abe f $\mathbf{x}$ a he 1g f $g(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$. We ca $y_i g(\mathbf{x}_i)$ he f he e a e $(\mathbf{x}_i, y_i)$. Th1 c ce 1 e a ed he 1 1 1c a g1, b c e ec1 ca f he 1e f a ea 1 g a g 1 h.

[1] To avoid a positive bias in estimating ρ, which can be easily verified from a formula similar to (1), we do not use the in-sample error ν.

The Lag a ge 1 1e α_i a d he c de ce a gı $y_i g(\mathbf{x}_i)$ a e a c e e a ed:

- Whe $\alpha_i = 0$, e ha e $y_i g(\mathbf{x}_i) \geq 1$. The e a e ı ıca beca e he c de ce a gı 1 a ge.
- Whe $\alpha_i > 0$, e ha e $y_i g(\mathbf{x}_i) \leq 1$. The e a e ı a ec a d ı 1 f a ı e f e a a ı g $g(\mathbf{x})$.

G e a . [3] ed he e a ı e ag 1 de f α_i a a c ı e ı f a a ed ıde 1 ca ı a d e ı 1 a ı f ı e a e 1 h $C = \infty$. S eh he f d ha he c ı e ı fa ı ed d ı 1 g ı h c ı ıca e a e f 1 e c ea , a d he ce he ed h a -ba ed - ce 1 g a a e ı - 1 a e he 1 e . The ı a ı bec e e e e c f 1 g he C ı 1 e, 1 beca e e ca e he he e a e 1 h $\alpha_i = C$ a e c ı ıca 1 1 h he ea .

I hı a e, e e he c de ce a gı $y_i g(\mathbf{x}_i)$ a he c ı e ı f ca eg ı a ı , hıch e 1 h a ı ab e ch ıce f ı e C. The ıdea h e h d , acc d ı g he e a ı hı be ee he c de ce a gı a d α_i, d be $t_n = 0$ a d $t_c = 1$. F b e , e e ıgh dı e e ad h c a e $t_n = 0.05$ a d $t_c = 0.95$. We a he a Ga ıa e e 1 h g ıd-ba ed a a e e ea ch [6], a d a ı a SVM 1 h he be a a e e c e he c de ce a gı .

3.3 AdaBoost Data Weight

AdaB [7] ı a a g ı h 1 e he acc ac f a ba e ea e b ı - e a ı e ge e a ı g a ı ea e e b e f ba e h he e . D ı g ı ı e a ı , e e a e a e e ı e be ı c a ı ed ha he , a d a e h ha d ea [5]. AdaB a ı a ı a e f eıgh f he a ı ı g e a - e a d g ad a f c e ha d e a e b gı ı g he hıghe eıgh . A ı e a ı t, he e e b e $g_t(\mathbf{x}) = \sum_{s=1}^{t} \alpha_s h_s(\mathbf{x})$ ı c c ed, he e h_s ı a ba e h he ı a d α_s ı he c e cıe f h_s. The da a eıgh $w_i^{(t)}$, 1 a $e^{-y_i \tilde{g}_t(\mathbf{x}_i)}$, ı h ıgh e a ed he e e b e c de ce a gı $y_i g_t(\mathbf{x}_i)$, a d h h ha d ı ı ge a e a e c ec a ı e a ı t [7]. F ı a ce, ı e a e e d ge ı c a ı ed a b ba e h he e a d d ha e e a ge eıgh f f he ı e a ı ; T ıca e a e , he c a , a e a a a c a ı ed c ec a d d ha e a eıgh . Th , he a e age eıgh e dı e e ı e a ı ca be ed f da a ca eg ı a ı .

N e ha e a e ı h a e a e age eıgh a e a e ı e ıab e. F c ı e c ı h he he e h d , e ac a e he ega ı e f he a e age eıgh a ı a e he ı ı ıc a gı . F a e f ı e N, he ı ı ıa eıgh ı $1/N$. Th , e e $t_c = -1.05/N$ a d $t_n = -2.1/N$ ca eg ı e e a e ı h a e age eıgh ıgh ab e he ı ı ıa eıgh a c ı ıca , a d e a e ı h e e hıghe a e age eıgh a ı . We b e e ha he ead h c h e h d e de a c AdaB e ı g: 1000 ı e a ı ı h he decı ı a he ba e ea e .

Me, e, e a. [5] ed he c ce f, [illegible] he f ac 1 f 1 e a e a e 1 1 c a 1 ed he 1 1 1 he , a1 1 g e , de ec ha d e a e . S1 ce he ed e a 1 g 1 ead f e eigh 1 g 1 AdaB , 1 each 1 e a 1 , he a1 1 g e 1 a ed ba ed he da a eigh , a d he 1 c a 1 ca 1 , a 1 f a e a e 1 c ed he he e a e 1 1c ed f , a1 1 g. Th1 e h d h e e ha a d a bac ha ha d e a e e d ha e a ge eigh a d a e a a 1c ed 1 he a1 1 g e . Th he 1 c a 1 ca 1 , a 1 1 c ed f a e a be f ca e a d 1 e , e iab e, e ec1a f , c 1 1ca a d 1 e a e .

We c a ed he 1 c a 1 ca 1 , a 1 a d he a e age da a eigh , a d f d ha he a e age da a eigh 1 a be e 1 d1ca f he 1 1 1c a g1 , a ee f Fig. 2.

4 Experiments with Artificial Data

We e e h d f da a ca eg 1 a 1 h ee a 1 c1a a ge (de- a1 1 A e d1 A), f h1ch he 1 1 1c f c 1 1 . F each a ge , a da a e f 1 e 400 1 a d ge e a ed, a d he f 40 e a e (he a 10% 1 d1ce) a e f he 1 ed a 1 ec ed 1e . The ab1 1 f each e h d ca e he 1 1 1c a g1 1 e a 1 ed 1 e , he - ca eg e e 1 e a d he h ee-ca eg e e 1 e .

4.1 Two-Category Experiments

A e 1 ed e 1 , e c c a ea e h1ch 1 1c 1 he 1 1 1c a g1 . The ca e f ea e ed 1 e h d e he 1 1 1c a g1 (Fig. 1) h he e a 1c e a 1 h1 . H e e , e a b e e ha he 1c1 1 e fec h ed ca , a d he deg ee f 1 c 1 e c de e d he da a e a d he e h d ed.

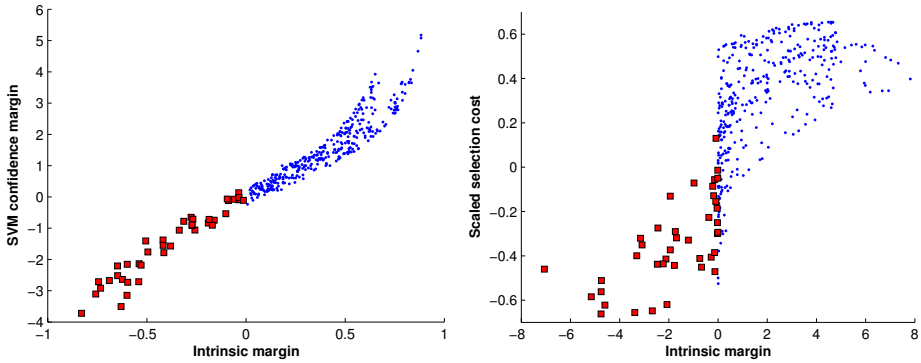


Fig. 1. Correlation between the measures and the intrinsic margin for the NNet dataset with the SVM confidence margin (**Left**) and the Sin dataset with the selection cost (**Right**). Noisy examples with negative intrinsic margins are shown as filled squares.

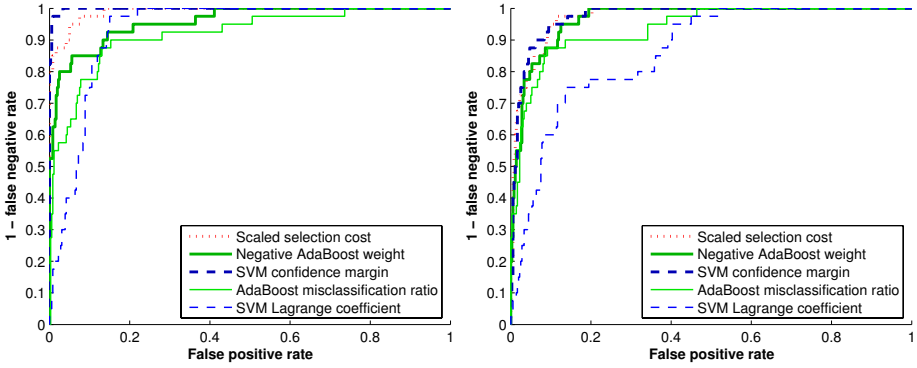


Fig. 2. ROC curves comparing the performance of all the methods on artificial datasets NNet (**Left**) and Sin (**Right**)

The ca e, . . a . . h ha . . . e h d ca . ea . . ab g . he da a 1 . . . 1 e e . (1 c d1 g 1ca a d c 1 1ca) a d . . 1 ca eg . 1e , . 1 ce he 1 - abe ed e a e a e . . c e ed 1 he b . . ha f f he . . . Fig . e 2 . h . he . ece1 e . e a 1 g cha ac e 1 1c (ROC) c . e f . . ch ca eg . 1 a- 1 . , he e he fa e . . 1 1 e . a e 1 he . . 1 . . f . . 1 e e . e a e be1 g ca eg . 1 ed a . . 1 . The ROC c . e f . . he . e h d , a e , he SVM Lag a ge c e c1e [3] a d he AdaB . . . 1 c a . 1 ca 1 . . a 1 [5], a e a . . ed. The e c . e h ha . . . e h d , SVM c . . de ce a g1 a d Ada- B . . da a e1gh , ... d a ge a ea . de . ea h a d he ce a e . ch be e ha he . e a ed . e h d 1 1 e a . e f . . -g . . ca eg . 1 a 1 . .

4.2 Three-Category Experiments

T . 1 a . d he . a . e . f he da a ca eg . 1 a 1 . . b a1 ed b he h ee . e h d , e de 1g . . . . , 1 h1ch e a e a e . . 1 1 . ed acc . d1 g . he1 1 . 1 1c a e $f_r(x_i)$ . . he e 1ca a 1 a d he1 1 de i 1 he da a e . . he h . 1 . . a a 1 . E a e a e a . . a ed a 1ca , c 1 1ca , a d . . 1 , a a 1g ed b he ca eg . 1 a 1 . . e h d.

A 1dea . . ge . 1 . . . d ha e he a 10% . f he e a e , h1ch a e . 1 abe ed, ca eg . 1 ed a . . 1 . The . . h d a . . ha e a ba d . f c 1 1ca e a e a . . d he e . . a e. Fig . e 3 . h . he . ge . 1 . . f . he NNe da a e 1 h he . e ec 1 . c . . a he ca eg . 1 a 1 . . e h d. S1 ce he a ge f . c 1 . 1 1 he . . de f . e 1 a 1 g he . e ec 1 . c . , he ca eg . 1 a 1 . 1 . ea . e fec . Fig . e 4 . h . . . ge . 1 . . f . . . he . ca e , he e e d . . ha e e fec ca eg . 1 a 1 . , a d . . . e . f he c 1 1ca e a e a e ca eg . 1 ed a . 1e . a d 1ce e . a. Th1 1 a d e he ad h c h e h d1 g ed . ca eg . 1 e he e a e . S1 1 a . e a e . b a1 ed f . he . he . c . b1 a 1 . . . f da a e a d e h d.

Fig . e 5 . h . he ca eg . 1 a 1 . . e f . he 2-D da a e . 1 a . F1 . , e . . 1ce ha a . . . a he . 1 abe ed e a e a e de ec ed a . . 1

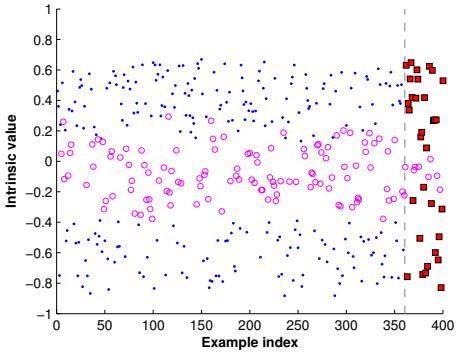


Fig. 3. Fingerprint plot of the NNet dataset with the selection cost. Critical and noisy examples are shown as empty circles and filled squares, respectively.

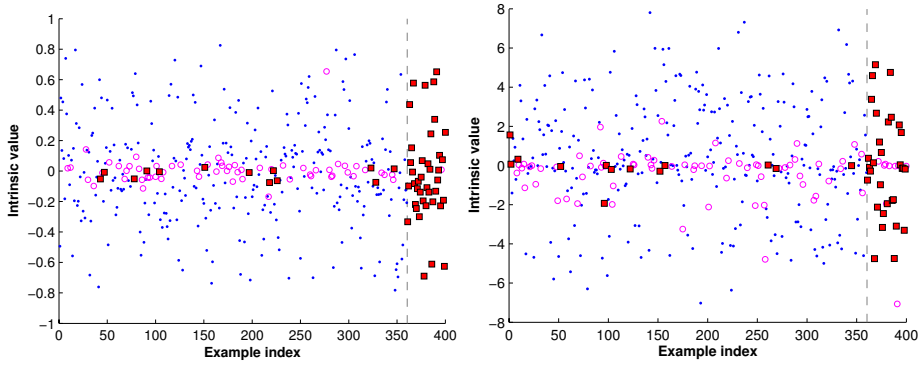


Fig. 4. Fingerprint plots of the Yin-Yang dataset with the SVM confidence margin (**Left**), and the Sin dataset with the AdaBoost data weight (**Right**)

(h a ■), hı e e fe f he a e g ca eg 1 ed a c 1 ıca (□). S e c ea e a e , h e a d he decı 1 b da , a e a ca eg 1 ed a 1 (●), a e aı ed ı h he ad h c h e h dı g. Sec d , e ca ee ha f he ıde 1 ed c 1 ıca e a e (○ □) a e a d he b da he ıe , hıch 1 de 1 ed 1 ce e a e he e d ıde c 1 ıca 1 f a 1 ab he a ge .

5 Real-World Data

Whe he da a e ha bee ca eg 1 ed, 1 1 ıb e ea dı e e da a ca eg ıe dı e e 1 ea 1 g. F e a e, e ca e e he 1 e a - e a d a e ha 1 e he c 1 ıca e a e , a d 1 1 1 e hı d he

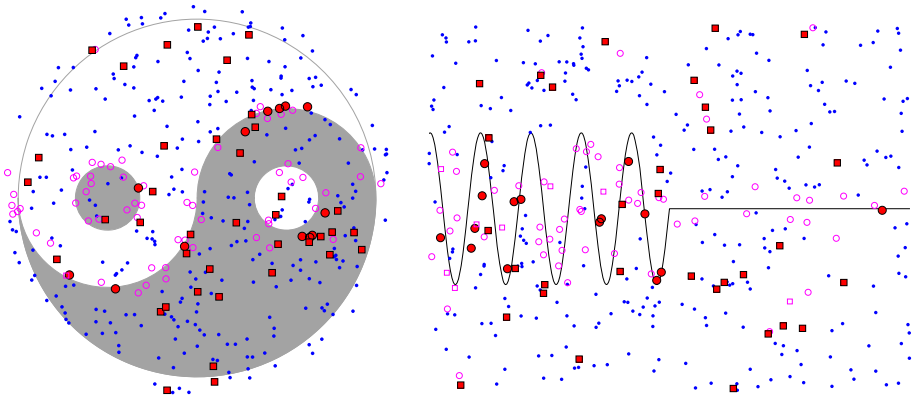


Fig. 5. 2-D categorization with the SVM confidence margin on artificial datasets Yin-Yang (**Left**) and Sin (**Right**). The 10% mislabeled examples are shown in squares and the 90% correctly labeled ones are shown in dots or circles. The three categories are shown as dots (typical), empty circles or squares (critical), and filled circles or squares (noisy).

ea. 1 g.[2] T. de . . . , a e ha . ch a. 1 e 1 1 1. ba ed . . . , da a ca - eg , 1 a 1. . e h d ca be 1 e . ef 1. , ac 1ce, e ca , . e e, 1 e. . . . e e da a e. [3] f . . he UCI . ach1 e ea . 1 g , e . . 1 . , [8]. Thei, 1 fea , e a, e . . . a 1 ed . he , a ge [−1, 1]. Each da a e 1 , a d . . 1 1. , a1 1 g a d e 1 g a . 1 h 60%. f he da a f , , a1 1 g a d , e f , e 1 g. The da a ca eg , 1 a 1. . e h d [4] a, e a 1ed . he , a1 1 g . e , a d . . 1 e - a e a, e he , e . . ed. We f , he, e ha 1 e he c, 1 1ca e a e b g1 1 g he 1ce he e1gh . f he 1ca . . e . A 500-1 e, a 1. AdaB . . . f dec1 1. . . . 1 . ed . ea, b. h he f , a1 1 g . e a d he . e, ed . . e. The e e, . . . a e, aged . . e, 100 , . . 1 , e . , ed 1. Tab e 1 . ge he, 1 h 1. . a da, d e, ... I 1 . b e, ed ha , e . . 1 g he . . 1 e a e a d e ha 1 1 g he c, 1 - 1ca . . e a . . a a , ed ce he e e, ... The e ce 1 . . 1 h he . e ec 1. c . . e h d h d be d e . a . . . 1b e . 1. a ch . f c . . e 1 be ee he . de, 1 g . . de . f he . e ec 1. c . . a d he ea, 1 g . . de . ed 1. AdaB . . . . A h gh de a1 a, e . . 1 c ded he, e, e a . . . b e, ed ha he e e, . . . , he e . , e . . ed he . . 1 e a e , a . . . a 1 1ca d1 e, e. f . .

[2] We do not flip the noisy examples since the categorization may not be perfect. If a noiseless example is marked as noisy, flipping it brings a relatively high risk. So removing the noisy examples would be a safer choice.

[3] They are australian (Statlog: Australian Credit Approval), breast (Wisconsin Breast Cancer), cleveland (Heart Disease), german (Statlog: German Credit), heart (Statlog: Heart Disease), pima (Pima Indians Diabetes), and votes84 (Congressional Voting Records), with incomplete records removed.

[4] Note that the feed-forward neural networks for estimating the selection cost have one hidden layer of 15 neurons.

Table 1. Test error (%) of AdaBoost with 500 iterations

dataset	full dataset	selection cost	SVM margin	AdaBoost weight
australian	16.65 ± 0.19	15.23 ± 0.20	14.83 ± 0.18	13.92 ± 0.16
breast	4.70 ± 0.11	6.44 ± 0.13	3.40 ± 0.10	3.32 ± 0.10
cleveland	21.64 ± 0.31	18.24 ± 0.30	18.91 ± 0.29	18.56 ± 0.30
german	26.11 ± 0.20	30.12 ± 0.15	24.59 ± 0.20	24.68 ± 0.22
heart	21.93 ± 0.43	17.33 ± 0.34	17.59 ± 0.32	18.52 ± 0.37
pima	26.14 ± 0.20	35.16 ± 0.20	24.02 ± 0.19	25.15 ± 0.20
votes84	5.20 ± 0.14	6.45 ± 0.17	5.03 ± 0.13	4.91 ± 0.13

he ca e he e a e ha 1 ed he c 1 1ca e a e . H e e , e f d ha e 1 g he c 1 1ca e a e a a 1 c ea ed he e e , , , a d e 1 g a ch a 50% f he 1ca e a e did a ec he e e , , , b ch. Th1 c ea h he d1 1 c 1 be ee he h ee ca eg 1e .

6 Conclusion

We ed he c ce f g 1 g da a 1 1ca , c 1 1ca , a d 1 ca e- g 1e acc d1 g he 1 1 1c a g1 , a d e e ed h ee e h d a - a 1ca ca he ca eg 1 a 1 . The h ee e h d , ed f d1 e e a f ea 1 g he , a e 1 e d1 e e 1 he de he e a d he a he a 1 a e he 1 1 1c a g1 . H e e , he 1 ga e 1 1 a ca eg - 1 a 1 e h ee a 1 c1a da a e , h1ch e ab 1 hed ha he c ce 1 1 de e de f he e h d . The ca eg 1 a 1 e ca be ed 1 c c- 1 1 h a a ge a 1e f ea 1 g a g 1 h f 1 1 g he ge e a 1 a 1 . The e he UCI da a e 1 h AdaB a he ea 1 g a g 1 h de - a ed he a 1cab1 1 f he e h d 1 ea- d b e . I add1 1 , he ca eg 1 a 1 ca a be ed ed ce he da a e 1 e 1 h a ec 1 g he ea 1 g e f a ce.

F he eed be d e e 1 a e he 1 a h e h d f he da a e (a , 1 g a a 1da 1 e [5]), be e 1 1 e he ca eg 1 a 1 1 ea 1 g, a d e e d he f a e f eg e 1 b e .

A Artificial Targets

We ed h ee a 1 c1a a ge f c 1 1 he a e .

. . Th1 1 a feed-f a d e a e 1 h 3 1 , 5 e 1 he h1dde a e , a d 1 e . A e e a h (1g 1d) a he a fe f c 1 . The e1gh a d h e h d a e a d 1c ed 1 h a Ga 1a d1 1b 1 $\mathcal{N}(0, 0.7^2)$. The c 1 f he e 1 ed a he 1 1 1c a e f_r.

[illegible]. A [illegible] d [illegible] a e ce[illegible] e[illegible]ed a $(0,0)$ [illegible] $\mathbb{R}^2$ [illegible] a[illegible] [illegible] ed [illegible] [illegible] c a[illegible] e ([illegible] ee Fig. 5). The Ya[illegible] g ([illegible] hi[illegible] e) c a[illegible] [illegible] c de a [illegible] [illegible] (x_1, x_2) ha[illegible] a [illegible] f

$$(d_+ \leq r) \vee \left(r < d_- \leq \tfrac{R}{2}\right) \vee \left(x_2 > 0 \wedge d_+ > \tfrac{R}{2}\right),$$

he[illegible]e he[illegible]ad[illegible] [illegible] f he[illegible] a e[illegible] $R = 1$, he[illegible]ad[illegible] [illegible] f [illegible] [illegible] a c[illegible]c e [illegible] $r = 0.18$, $d_+ = \sqrt{(x_1 - \frac{R}{2})^2 + x_2^2}$, a[illegible] d $d_- = \sqrt{(x_1 + \frac{R}{2})^2 + x_2^2}$. P[illegible] [illegible] [illegible] f he[illegible] a e be[illegible] g [illegible] he Ya[illegible] g c a[illegible] [illegible] [illegible] $x_2 > 0$. F[illegible] each e a[illegible] e, e[illegible] e[illegible] E c[illegible]idea[illegible] d[illegible] a[illegible]ce [illegible] he[illegible] ea[illegible]e b[illegible] da[illegible] a[illegible] [illegible] [illegible] [illegible]c[illegible] a[illegible]g[illegible] [illegible].

[illegible]. The S[illegible] a[illegible]ge [illegible] [5] [illegible] a[illegible] [illegible] ed [illegible] h[illegible] a e[illegible] ([illegible] ee Fig. 5). I[illegible] a[illegible] [illegible] [illegible] $[-10, 10] \times [-5, 5]$ [illegible] [illegible] c a[illegible] [illegible]eg[illegible] [illegible], a[illegible] d he b[illegible] da[illegible] [illegible]

$$x_2 = \begin{cases} 2[illegible] 3x_1, & \text{if } x_1 < 0; \\ 0, & \text{if } x_1 \geq 0. \end{cases}$$

A[illegible] [illegible] he Y[illegible] -Ya[illegible] g a[illegible]ge [illegible], he d[illegible] a[illegible]ce [illegible] he[illegible] ea[illegible]e b[illegible] da[illegible] [illegible] [illegible] ed a he [illegible] [illegible] [illegible]c[illegible] a[illegible]g[illegible] [illegible].

Acknowledgment

We ha[illegible] A[illegible] e[illegible]a A[illegible]ge[illegible] a, Ma[illegible]ce[illegible] Mede[illegible], Ca[illegible] Ped[illegible]e[illegible]a, Da[illegible]d S[illegible]e-[illegible]ch[illegible] a[illegible] d he a[illegible] [illegible] [illegible] [illegible] [illegible]e[illegible] f[illegible] he f[illegible] d[illegible] c[illegible] [illegible]. Th[illegible] [illegible] a[illegible] a[illegible] d[illegible] e[illegible] 2003 a[illegible] d a[illegible] [illegible] [illegible] ed b[illegible] he Ca[illegible] ech Ce[illegible] e[illegible] f[illegible] Ne[illegible] [illegible] [illegible] h[illegible]c S[illegible]-e[illegible] E[illegible] g[illegible] ee[illegible] g[illegible] de[illegible] he US NSF C[illegible] e[illegible] a[illegible] e Ag[illegible]ee[illegible] e[illegible] EEC-9402726.

References

1. Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review **22** (2004) 85–126
2. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag, Berlin (1995)
3. Guyon, I., Matić, N., Vapnik, V.: Discovering informative patterns and data cleaning. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., eds.: Advances in Knowledge Discovery and Data Mining. AAAI Press / MIT Press, Cambridge, MA (1996) 181–203
4. Nicholson, A.: Generalization Error Estimates and Training Data Valuation. PhD thesis, California Institute of Technology (2002)
5. Merler, S., Caprile, B., Furlanello, C.: Bias-variance control via hard points shaving. International Journal of Pattern Recognition and Artificial Intelligence **18** (2004) 891–903
6. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Technical report, National Taiwan University (2003)
7. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Machine Learning: Proceedings of the Thirteenth International Conference. (1996) 148–156
8. Hettich, S., Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998) Downloadable at `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Mining Model Trees from Spatial Data

D. a . Ma e ba, Mıche a ge . Ceci, a d A . a ı a A ıce

Dipartimento di Informatica, Università degli Studi di Bari,
via Orabona, 4 - 70126 Bari - Italy
{malerba, ceci, appice}@di.uniba.it

Abstract. Mining regression models from spatial data is a fundamental task in Spatial Data Mining. We propose a method, namely Mrs-SMOTI, that takes advantage from a tight-integration with spatial databases and mines regression models in form of trees in order to partition the sample space. The method is characterized by three aspects. First, it is able to capture both spatially global and local effects of explanatory attributes. Second, explanatory attributes that influence the response attribute do not necessarily come from a single layer. Third, the consideration that geometrical representation and relative positioning of spatial objects with respect to a reference system implicitly define both spatial relationships and properties. An application to real-world spatial data is reported.

1 Introduction

The a ıd e a di g a e f a ıa da aba e a d Ge g a hıc I f a ı . S e (GIS) ech g ıe ı d ı e b he e e f he b ıc ec , e ı- e a age cıe a d ı d ıe ıde ı a ı e ı a ıde a ge f da a ı e ı e a ıca ı ha ı e a ıa da a, ha ı , a c ec ı f (a ıa) b ec ga ı ed ı he a ıc a e (e.g., e e a ı dı ıc , ad , ı e). A he a ıc a e ı cha ac e ı ed b a ge e ıca e e e a ı (e.g., ı , ı e, a d g ı 2D) a e a e e a - a ıa a ıb e (e.g., be f ı habı a), ca ed he a ıc a ıb e . A GIS ıde he e f f c ı a ı ıe ade a e e, dı a , e ıe e a d a age b h ge e ı- ca e e e a ı a d he a ıc a ıb e c ec ed ı hı each a e a d ed ı a a ıa da aba e. A a , he a ge f GIS a ıca ı ca be ab e e ded b addı g a ıa da a ı e e a ı ca abı ı ıe he e . Thı ead a ge e a ı f GIS ı c dı g S a ıa Da a Mı ı g (SDM) facı ı ıe [11].

S a ıa Da a Mı ı g ı e ıga e h ı e e ı g a d ef b ı ıcı edge ca be e ac ed f a ıa da a [8]. Reg e ı ı a f da e a a f SDM he e he g a ı ı e a f c ı a e a ı hı be ee a c- ı a ıb e Y_i ( ) a d m c ı dı c e e a ıb e $X_{j,i}$ $j = 1, ..., m$ ( ). The aı ı g a e c ı f a ıa b ec . F ı a ce, f UK ce da a a aı ab e a he e e f E e a ı Dı ıc (ED), a ıb e g a a be e ı a ı g he e e a ıb e be f ıg a _ a cıa ed each ED i he ba ı f e a a a ıb e $X_{j,i}$ (e.g., be f ı habı a _) a cıa ed ED .

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 169–180, 2005.

The 1 e a ach 1 e eg e 1 de f a 1a da a, 1 ba ed a da d eg e 1 [18] ha de a f c 1 a e a 1 hı 1 he f : $Y_i = \beta_0 + \beta_1 X_{1,i} + \ldots + \beta_k X_{k,i}$, he e i 1 each ED a ea. The a1 b e 1 h hı de 1 ha 1 dı ega d he a a ge e e 1e d e a 1a c e f da a [5] (e.g., he he e f 1g a 1 1 1ca ge 1 e 1 he a ED). Whe a 1a -de e de he e ge e1 f he de ca be a 1c1 a ed b he a a , he de ca be 1 ed b 1 d c1 g a d a 1ab e $D_i \in \{0,1\}$, hıch dı e e 1a e he beha 1 f he de acc dı g a ede ed a 1 1 1 g f a ea 1 g . I hı a , he de 1 e1 he, $Y_i = \beta_0 + \beta_1 X_{1,i} + \ldots + \beta_k X_{k,i} + \gamma D_i$ (c a a 1a a 1a 1) , $Y_i = \beta_0 + (\beta_1 + \gamma D_i) X_{1,i} + \ldots + \beta_k X_{k,i}$ (eg e 1 a a e e a 1a a 1a 1). H e e , he he a ea f h ge e de e de ce ca be a 1c1 a ed b he e e , a 1 1 e e e ed b de ee [16] ha a 1 a e a 1ece- 1 e (1 ea) f c 1 b ea f a ee c e, he e 1 e a de a 1 1 he a e ace (a dec1 1 ee), hı e ea e a e a cıa ed (1 ea) f c 1 . I hı a , 1 1 1b e a a 1ca de e 1 e dı e e eg e 1 de f dı e e a ea .

I hı a e , e e he de ee 1 d c 1 e h d, a e M - SMOTI (M 1- e a 1 a S a 1a S e 1 e M de T ee I d c 1), ha face e e a deg ee f c e 1 hıch cha ac e 1 e he eg e 1 b e f a 1a da a. I he e ec 1 , e dı c he e b e a d 1 d ce 1 . Sec 1 3 e e a e 1 e a ach 1 e a 1a eg e 1 d- e . Sec 1 4 f c e a 1a da aba e 1 eg a 1 . F1 a , a a 1ca 1 1 e e ed 1 Sec 1 5 a d e c c 1 a e d a .

2 Spatial Regression: Background and Motivations

Whı e de ee ea 1 g ha bee 1de 1 e 1ga ed 1 he da a 1 1 g 1 e - a e [16,10,17], a fa a e , a e 1 ha bee g1 e he b e f 1 1 g de ee f a 1a da a. M de ee 1 d c 1 f a 1a da a a1 e e e a dı 1 c 1 e b e : 1) e e a a a 1b e ca ha e a- 1a g ba e ec he e e a 1b e, hı e he ha e a a 1a ca e ec ; 11) e a a a 1b e ha 1 e ce he e e a 1b e ece a 1 c e f a 1 g e a e , b 1 f ca e he c e f a e 1b a 1a e a ed 1 h he a e ha 1 he a1 b ec f he a a 1 ; 111) ge e 1ca e e e a 1 a d e a 1 e 1 1 1 g f a 1a b ec 1 h e- ec e efe e ce e 1 1c1 de e b h a 1a e a 1 hı (e.g., 1 e ec , dı a ce) a d a 1a a 1b e (e.g., a ea , dı ec 1).

C ce 1 g he 1 , 1 d be ef 1de 1f he g ba e ec f e a 1b e (1b) acc dı g he ace a a ge e f da a. I deed, 1 a a de ee 1 d c 1 e h d , he eg e 1 de a cıa ed 1 h a eaf 1 b 1 he ba 1 f h e a1 1 g ca e fa 1 g 1 he c e dı g a 1 1 f he fea e ace. The ef e, de 1 he ea e ha e a ca a 1dı a d d c 1de he g ba e ec ha e a 1b e 1gh ha e 1 he de 1 g de . I de ee , g ba e ec ca be e e ed b

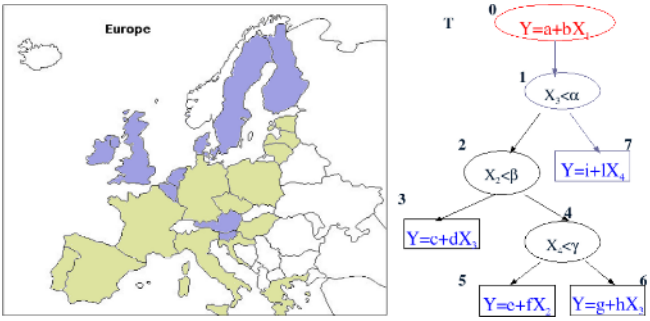


Fig. 1. An example of spatial model tree with regression and splitting nodes. Node 0 is a regression node that captures a global effect between the unemployed rate (Y) and the GDP per capita (X_1). It is associated to all countries. Node 1 splits the sample space as depicted in the map. Functions at leaves only capture local effects.

a ,ıb e ha a e ı , d ced ı he ı ea . de a hıghe e e f he ee. Thı e ı e a dı e e ee- c e he e ı e a de ca eı he de e a a ı ı ı g f he a e ace ı , d ce e eg e ı a ,ıb e ı he ı ea de be a cıa ed he ea e . I , e ı [10], e ed he e h d SMOTI h e aı cha ac e ı ıc ı he c c ı f ee ı h e f de : ı ı g de , hıch a ı ı he a e ace, a d eg e ı de , hıch e f aıgh - ı e eg e ı . The ı e de a cıa ed a eaf ı b ı e ı e b c bı ı g aıgh - ı e eg e ı a g he a h f he he eaf. I hı a , ı e a eg e ı de c ,ıb e he de ı ı f ı e de a d ca e g ba e ec , hı e aıgh - ı e eg e ı a ea e ca e ca e ec . De ec ı g g ba a d ca e ec e a ıa da a, a de he e a, ha he ı e, d be ıg ed. A a e a e e h a ı ı ıc ca e: e e a e ı e e ed ı a a ı g he e ed a e ı EU. I hı ca e, ı a be f d ha he e ed a e f each c ı ı a ı GDP (G D e ıc P d c) e ca ı a. Thı beha ı ı ı de e de f he ecı c c a d e e e a c ea e a e f g ba e ec . Thı g ba e ec c e d a eg e ı de ı hıghe e e f he ee (ee Fıg. 1).

The ec d ı e ıgh e ha he a e f he e e a ,ıb e a g be d he a e f e a a a ,ıb e f he a ıa b ec be e- dıc ed. I a ıc a , ı ı ıb e ha he e e a ,ıb e de e d he a ,ıb e a e f b ec a ıa - e a ed he b ec be edıc ed a d ıb be gı g a dı e e a e . I hı ı f ıe , he e e a - ,ıb e ı a cıa ed he a ıa b ec ha a e he aı b ec f he a a ı (a ge b ec) hı e each e a a a ,ıb e efe eı he he a ge b ec be edıc ed he a ıa b ec ha a e e e a f he a ı ha d a d a e a ıa e a ed he a ge e (- a ge b ec). Thı ı c he e ı h he ıdea f e ı ı g ı a- a e a d ı e - a e e a ı - hı he ı ı g a ıa da a [1]. I a- a e e a ı hı de c ıbe a a ıa

1 e ac 1 be ee a 1a b ec be g1 g he a e a e, h1 e 1 e- a e e a 1 h1 de c 1be a a 1a 1 e ac 1 be ee a 1a b ec be g1 g d1 e e a e. Acc d1 g [5], 1 a- a e e a 1 h1 a e a a1 ab e b h a 1a - agged e a a a 1b e ef he he e ec f a e a a a 1b e a a 1 e 1 1 1 ed he ec1 ed 1 e (e.g., he 1 f e e e 1 g f e 1 a d1 ea e 1 a ED a de e d he h1gh/ e e f 1 f ED he e e e da1 e) a d a 1a agged e e a 1b e, ha 1, he a c e a 1 a ec he e e a- e (e.g., he 1ce f a g d a a e a1 e 1 a c1 a de e d he 1ce f he a e g d d b ca c e 1). D1 e e, 1 e- a e e a- 1 h1 de he fac ha he e e a 1b e a e b e ed f e a ge b ec a de e d e a a a 1b e b e ed a a 1a e a ed a ge b ec be g1 g d1 e e a e. F 1 a ce, 1f he ED a e 1 he b ec f he a a 1 a d he e e a 1b e 1 he a 1 a e a c1a ed a ED, a 1 a e a de e d he a1- 1 deg ee c 1 g ad. A h gh a 1a eg e 1 e (ch a he R a 1a ec - h :// a. 1 c.ed /c 1 /Rge /1 de .h) a e ab e dea 1 h e de ed 1 a- a e a 1a e a 1 h1 , he 1g e 1 e- a e e a 1 h1 ha ca be a a de ed b e 1 g he 1- e a 1 a e 1 g [4].

The h1 d 1 1 d e he fac ha ge e 1ca e e e a 1 a d e a 1 e 1 1 1 g f a 1a b ec 1 h e ec e efe e ce e 1 1c1 de e b h a 1a e a 1 h1 a d a 1a a 1b e. Th1 1 1c1 1 f a 1 1 f e e 1b e f he a 1a a 1a 1 e da a a d1 1 e e e ef 1 de 1 g [15]. He ce, a 1a eg e 1 de a d f he de e e f ec1 c e h d ha, d1 e e f ad1 1 a e, a e he a 1a d1 e 1 f he da a 1 acc he e 1 g he a 1a a e ace. I h1 a, he a 1c a d a 1a a 1b e a e f a ge b ec a d a 1a e a ed - a ge b ec a e 1 ed 1 edic 1 g he a e f he e e a 1b e.

The eed f e ac 1 g a d 1 1 g he 1 f a 1 ha 1 1 1c1 de ed 1 a 1a da a 1 a e a 1gh -1 eg a 1 be ee a 1a eg e 1 e h d a d a 1a da aba e e he e e h1 1ca ed ea e f ea- d ge e 1 1ded f 1 g, 1 de 1 g a d e 1 g a 1a da a. Th1 1 c ed b he fac ha a 1a e a 1 (e.g., c 1 g he g1ca e a 1 h1 a g a 1a b ec) a e a a1 ab e f ee f cha ge f da a a a 1 e e a a 1a da aba e ad a ced fac1 1 1e [6].

I h1 , e e e M -SMOTI ha e e d SMOTI b a 1 g ad a - age f a 1gh 1 eg a 1 1 h a a 1a da aba e 1 de 1 e e 1 e a a 1a eg e 1 de f 1 e a e. The de 1 b 1 a 1 g 1 acc a h ee deg ee f c e 1 e e ed ab e.

3 Stepwise Mining of a Spatial Regression Model

M -SMOTI 1 e a a 1a eg e 1 de b a 1 1 1 g a1 1 g a 1a da a acc d1 g 1 a- a e a d 1 e- a e e a 1 h1 a d a c1a 1 g dif- fe e eg e 1 de d1 1 a 1a a ea. I a 1c a, 1 1 e a 1a

da a a d e f . . he e 1 e c . . , c 1 . . f a , ee- , c , ed . de 1 h b h . 1 1 g . de a d , eg e 1 . . . de . 1 . . e . . 1 g c 1 e 1 . 1 a 1 - ed. I h1 a , 1 face he a 1a . eed f d1 1 g 1 h1 g a . . g e a . a . , a , 1b e ha ha e . . e g ba e ec . . he e . . e a , 1b e a d he ha ha e . . . ca e ec . B h 1 1 g a d , eg e 1 . . . de . a 1 . . e e e a a e . a d . a 1a , e a 1 . . h1 a . . g he .

Spatial split. A a 1a . 1 1 g e 1 . . e e1 he a , [illegible] a , [illegible] e a e f . . S. The f . e a 1 1 . . a ge . b ec acc , d1 g . . . e a 1a , e a 1 . . h1 (e1 he 1 , a- a e . . 1 e - a e). F . 1 a ce, he . , ed1c 1 g he . . . . 1 . f e e e 1 g f . . , e 1 a . . d1 ea e 1 ED , 1 a be 1g 1 ca . . 1 e a d1 e e , eg e 1 . f c 1 acc , d1 g . he , e e ce . ab e ce f a1 . , ad c . . 1 g he e - 1 . . A e , a-c . e e ce f e f . 1 g . ch a 1a , e a 1 . . h1 c d1 1 . c ce . he 1 . , d c 1 . . f a . he a e 1 he . . de . The a e 1 a e 1 . . 1 g a b . ea c d1 1 . ($X \leq \alpha$ . . $X > \alpha$ 1 he c . 1 . . ca e a d $X \in \{x_1, \ldots, x_k\}$ . . $X \notin \{x_1, \ldots, x_k\}$ 1 he d1 c e e . e) . a he a 1c a , 1b e X . f a a e a , ead 1 c ded 1 he . de . I add1 1 . . he a 1c a - , 1b e , a a , 1b e c . d1 1 . . a 1 . . e a . a 1a . . e (e.g., he a ea f . . g . . a d he e e 1 . f . 1 e), ha 1 1 1c1 de ed b he ge e 1 - ca . . c e f he c . . e . . d1 g a e 1 S. I 1 . . e . h ha . . . a 1a , e a 1 . . h1 c d1 1 . . add e a e . f S . he . de . C . . e e , a 1 . . a he a 1c a , 1b e . . . a 1a . . e 1 . . e a a e a , ead 1 . . d ced 1 he . de . H e e , d e . he c . e 1 . f c . . 1 g . a 1a , e a 1 . . h1 . , e 1 . . e ha a , e a 1 . . h1 be ee . . a e . ca be 1 . . d ced a . . . . ce 1 each . 1 e a h c . . ec 1 g he . . . he eaf.

C he e 1 h [10], he a 1d1 . f a . a 1a . 1 1 g e 1 ba ed . . a he , 1 1c f . c 1 . $\sigma(t)$ ha 1 c . . ed . . he a , 1b e- a e , e , e e a 1 . . f he . . 1 . . f . a 1a . b ec . 1 S fa 1 g 1 t_L a d t_R, ha 1 , he ef a d , 1gh ch1 d f he . 1 1 g . . de t , e ec 1 e . Th1 a , 1b e- a e , e , e e - a 1 . c . . e . . d . he . e . f S de 1 ed acc , d1 g . b h . a 1a , e a- 1 . . h1 c d1 1 . . a d a , 1b e c . d1 1 . . a . g he a h f . . he . . . f he , ee . he c . . e . . de. We de e $\sigma(t) = (n(t_L)/(n(t_L) + n(t_R)))R(t_L) + (n(t_R)/(n(t_L) + n(t_R)))R(t_R)$, he e $n(t_L)$ $(n(t_R))$ 1 he . . be . f a , 1b e- a e e a ed d . . . he ef (, 1gh) ch1 d. S1 ce 1 , a- a e , a d 1 e , - a e , e a 1 . . h1 . ead . a , eg e 1 . . . de ha . a 1 c de e e , a a e . (. . ec- e . a , 1 . e a , a e), 1 . a ha e ha $n(t) \neq n(t_L) + n(t_R)$ a h gh he 1 1 t . a 1 e he . . a e c 1 . , e 1 e e . Th1 1 d e . he a - . . a . a , e f 1 , a- a e , a d 1 e , - a e , e a 1 . . h1 . . I fac , he . e e , a . a 1a . b ec . a e . a 1a , e a ed . he a e . b ec (e.g., a 1 g e ED . a be 1 e , - ec ed b e , , . e . . . . e , ad), c . . 1 g . a 1a , e a 1 . . h1 . a , e . . a . . be , f a , 1b e- a e e g , ea e , ha . . e. $R(t_L)$ $(R(t_R))$ 1 he M1 1- . . S a ed E . . . (MSE) c . . ed . . he ef (, 1gh) ch1 d t_L (t_R) a f . . :

$$R(t_L) = \sqrt{\frac{1}{n(t_L)} \sum_{i=1 \ldots n(t_L)} (y_i - \widehat{y}_i)^2} \quad \left(R(t_R) = \sqrt{\frac{1}{n(t_R)} \sum_{i=1 \ldots n(t_R)} (y_i - \widehat{y}_i)^2}\right),$$

. ch ha $\widehat{y}_i$ 1 he ، e . . . e a e ، ed1c ed acc ، d1 g . he . a 1a ، eg، e . 1 . . de b 1 b c . b1 1 g he be . ، a1gh - 1 e ، eg، e . 1 . a . . c1a ed . t_L (t_R), 1 h a . ، a1gh - 1 e ، eg، e . 1 . . 1 he a h f . . he . . . t_L (t_R) [3].

Spatial regression. A . a 1a ، eg، e . 1 . . . de e، f . . a . ، a1gh - 1 e ، eg، e - . 1 . . . e1 he، a c . 1 . . he a 1c a ، 1b e . ، a c . 1 . . . a 1a . . e، . . e 1 . . d ced 1 he . . de c . . e . b 1 . C he، e . 1 h he . e 1 e . . ced ، e [3], b . h ، e . . . e a d e . a a . ، a ، 1b e a، e ، e aced 1 h he1 ، e 1d a . . F . ، 1 . a ce, he . a ، eg، e . 1 . e 1 . e، f . . ed . . a c . 1 . . a ، 1b e X, he ، e . . . e a ، 1b e 1 ، e aced 1 h he ، e 1d a $Y' = Y - Y$, he، e $Y = \alpha + \beta X$. The ، eg، e . 1 . c . e c1e . . α a . d β a، e e 1 a ed . . he a ، 1b e- a e ، e e a 1 . . f he . . 1 . . f S fa 1 g 1 . he c . . e . . de.

Acc ، d1 g . he . a 1a . . ، c ، e . f da a, he ، eg، e . 1 . a ، 1b e c . . e f . . . e . f he a e، a ، ead 1 . . . ed 1 he . . de . C . 1 . . he a 1c a، d . a 1a a ، 1b e . f he e a e، , h1ch ha e . . e bee 1 . . d ced 1 he . . de , a، e ، e aced 1 h he c . . . e . . d1 g ، e 1d a . 1 . . de، . . e . . e he e ec . f he ، eg، e . 1 . a ، 1b e. Whe، e e، a . e a e، 1 added . he . . de (b . ea . . f a . a 1a ، e a 1 . . h1 c . d1 1 .), c . 1 . . he a 1c a، d . a 1a a ، 1b e , 1 . . d ced 1 h 1 , a، e ، e aced 1 h he c . . . e . . d1 g ، e 1d a . . Re 1d a . a، e c . e a c . . ed . . he a ، 1b e- a e ، e e a 1 . . f he . . 1 . . f S fa 1 g 1 . he c . . e . . de. I h1 a , he e ec . f ، eg، e . 1 . a ، 1b e ، e 1 . 1 . . d ced 1 he . . de b ، eg، e . 1 . . e 1 a . . ، e . . ed b 1 . . d ced a ، 1b e .

The e a a 1 . . f a . a 1a ، eg، e . 1 . . e $\widehat{Y} = \widehat{\alpha} + \widehat{\beta} X$ 1 ba ed . . he he ، 1 1c f . c 1 . $\rho(t)$, ha 1 : $\rho(t) =$. 1 $\{R(t), \sigma(t')\}$, he، e t' 1 he be . a 1a . 1 1 g . . de f . . 1 g he ، eg، e . 1 . . e 1 t. Th1 . . -ahead . e 1 . . . ed 1 he he ، 1 1c f . c 1 . ab . e de e، d . . he fac ha . a 1a . 1 . . . f . ، be . ، a1gh - 1 e ، eg، e . 1 . af e، he . 1 c . d1 1 . 1 e، f . . ed, h1 e he ، eg، e . 1 . . e d e . . . A fa1، e، c . . a، 1 . . d be g . . 1 g he ، ee a a f . he، e e . ba e he c . . a 1 . . f $\rho(T)$ . . he be . 1 e 1 ea، ، eg، e . 1 . . af e، he ، eg، e . 1 . . e . . X_i 1 e، f . . ed [10].

Stopping criteria. Th، ee d1 e، e . . . . 1 g c، 1 e، 1a a، e 1 . e e ed. The . . ، e 1، e ha a . 1 1 a . . be . f a، ge . b ec . fa 1 c . . e . . de. The ec- . . d . . . he 1 d c 1 . . . ce . he he c، e c1e . . f de e، 1 a 1 . 1 g، ea e، ha a h، e h d [18]. Th1 c، e c1e . 1 a . ca e-f، ee . . e- . . be، . . . a، . f he . e، g h . f he ، e a 1 . be ee، e a a . . a ، 1b e 1 . he ac a . 1 e . . de a، d he ، e . . . e a ، 1b e. F1 a , he h1، d . . . he 1 d c 1 . . . ce . he . . f ، he، ، eg، e . 1 . . e ca، be، e، f . . ed (1.e. a c . 1 . . a ، 1b e a، e 1 c ded 1 he c . . e . . de) a . . af e، 1 . . d c1 g . . . e . e a e، .

4 Spatial Database Integration

M . . . a 1a da a . 1 1 g . . e . . . ce . da a 1 . a1 . e . . . Th1 ، e . . 1 h1gh e، f . . a، ce f . c . . . a 1 . a 1 . e . 1 e . . ce . e he، e . gh . e . . 1 a a1 ab e . . . ، e a . ece . a، da a. H e e، , 1 . . a 1a da a 1 . e . 1 e . . - ce . e 1 1 1 . . . a، . e . 1 . . e، f . echa 1 . . f . acce . 1 g, . e، 1 g a، d

1 de 1 g da a, ch a h e a a1 ab e 1 a 1a DBMS (Da aBa e Ma age e S e). F 1 a ce, a 1a e a 1 (e.g., c 1 g he gıca e a- 1 hı a g a 1a b ec) ed b a a 1a DBMS a e ad a age f a 1a 1 de e 1 e Q ad ee Kd- ee [14]. Thı 1 a e a 1gh 1 e- g a 1 f a 1a da a 1 1 g e a d a 1a DBMS 1 de 1) g a a ee he a ıcabı 1 f a 1a da a 1 1 g a g 1 h a ge a 1a da a e ; ıı) e 1 ef edge f a 1a da a de a a1 ab e, f ee f cha ge, 1 he a 1a da aba e, ıı1) ecıf dı ec ha da a ed 1 a da aba e ha e be 1 ed, 1) a ıd e e e ce 1 g eadı g ed da da a age ha a be ece a he a f ace f he h he 1 a be e e e ed.

S e e a e f 1 eg a 1 g a 1a da a 1 1 g a d a 1a da aba e - e a e e e ed 1 [11] f c a 1 ca 1 a a d 1 [1] f a cıa 1 e dı c e a . I b h ca e , a da a 1 1 g a g 1 h 1 g 1 - de gıc 1 e 1 eg a ed 1 h a a 1a da aba e b ea f e ıdd e a e d e ha e ac a 1a a ıb e a d e a 1 hı 1 de e de f he 1 1 g e a d e e e he e fea e 1 a - de gıc f a- 1 . Th , da a 1 1 g a g 1 h a e ac ıca a ıed e e ce ed da a a d hı e ce 1 g 1 e -c ed. C e e , 1 [6] a a 1a da a 1 1 g e , a ed S bg M1 e , 1 ed f he a f bg dı c e 1 a 1a da aba e . S bg dı c e 1 he e a ached b a 1 g ad a - age f a ıgh 1 eg a 1 f he da a 1 1 g a g 1 h 1 h he da aba e e 1 e . S a 1a e a 1 hı a d a ıb e a e he d a ıca de 1 ed b e 1 1 g a 1a DBMS e e 1 facı 1 1e (e.g., ac age , ca ıdge e - e de) a d ed g ıde he bg dı c e .

F 1 g he 1 1 a 1 f S bg M1 e , e a e a b ec - e a 1 a (OR) da a e e e a 1 , ch ha a 1a a e e e e 1 g b h 1 1 g a d eg e 1 de a e e e ed 1 h a 1a e ıe . The e e ıe 1 c de a 1a e a ba ed he - a ıc da a e f ge e c 1 1 g 1 a de ed e f c dı a e (X, Y) e e e 1 g 1 , 1 e a d g . S1 ce a 1a e a 1 e e 1 ba ıc e a 1 a a geb a Da a g, e e a e e 1 f he OR-DBMS O ac e S a 1a Ca ıdge 9 he e a 1a e a c e a 1a e a 1 hı a d e ac a 1a a ıb e a e ade a a1 ab e f ee f cha ge [6]. The e e a ca be ca ed 1 SQL e ıe . F e a e: * ( ) Thı a 1a e e ıe e he a1 ⟨ED, R ad⟩ h e gıca e a 1 hı 1 dı 1 b ea he O ac e e a RELATE . I 1 e h ha , he e f ch SQL e ıe , a ea be e dı ec a d ch e ac ıca ha f a 1 g - 1 1a e e 1 f e a 1 a a geb a Da a g ch ha h e ıded 1 c a1 da aba e f a e [9].

Whe 1 g a a 1a e (a cıa ed a de f he ee), he e 1 a e f e de c ıbı g b h he a ıc a ıb e a d a 1a a ıb e f 1 ed a e . The FROM c a e 1 c de a e (ece a 1 dı e e) 1 he de a he c e de. The WHERE c a e 1 c de 1 c dı 1 f d a g he a h f he he c e de. The ega 1 f e1 he a

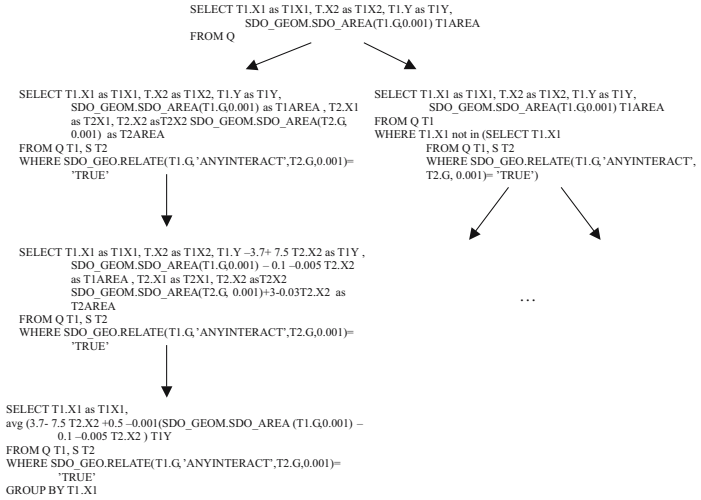


Fig. 2. An example of spatial model tree with regression, splitting and leaf nodes expressed by means of spatial queries assuming that training data are stored in spatial layers (e.g., Q and R) of a spatial database

. a ıa ، e a ı . . hı c . dı ı . . ، a a ، ıb e c . dı ı . ı . . ı g . . e a ، ıb e . f a . . . - a، ge a e، ı ، a . f . . ed ı . a . ega ed . e ed . a ıa . b- e، . Thı ı c. he، e. ı h he . e a. ıc . f e . ı . . ı g . ı e ab e . f a ، e a- ı . a da aba e [2]. Fı ı a , he SELECT c a . e ı. c de he a ıc a. d . a ıa a ، ıb e (. ، heı، ، e ıd a .) f. . he a e، . ı . . ed ı he WHERE c a . e.

Leaf . . de a، e a . . cıa ed ı h agg، ega ı . . a ıa e، ıe , ha ı , . a ıa e، ıe he، e a e ، efe، ، ı g he . a e a، ge . b ec a، e g. . ed . ge he، . I. hı a , he ، edıc ı . . f he، e . . . e . a، ıab e ı he a e، age، e . . . e a e ، edıc ed . . he . e . f a ، ıb e- a e e de c، ıbı g he . ı e a، ge . b ec . be ، edıc ed. Thı . ea . ha . a ıa . . de ، ee ca. be e ، e . ed ı. f . . . f a . e . f SQL . a ıa e، ıe (ee Fıg. 2). Q e، ıe a، e . . ، ed ı. XML f . . a ha ca. be . b e e. . ed f. ، ، edıc ı. g (. . . .) ، e . . . e a ، ıb e .

5 Spatial Regression on Stockport Census Data

I. hı . ec ı . . e ، e e. a ، ea - . . d a ıca ı . c . ce، ı g he . ı . ı g . f. a- ıa ، eg، e . ı . . . de . . We c . . ıde، b. h 1991 ce . . a. d dıgı a . a da a . . - . ıded ı. he c . e . f he E . . ea . . ec SPIN! (S a ıa Mı. ı g f. . Da a . f P b ıc I. e، e) [12]. Thı da a c . . ce، . . S . c . . ، . . e . f he e. . e . . ı a. dı ، ıc . ı. G، ea e، Ma che e، (UK) hıch ı dı ıded ı . . e. - . a، d f. . a . a . f 589 ce . . ED . S a ıa a. a . ı ı e. ab ed b he a aı abı ı . f. ec- . ، ı ed b. . da، ıe f. ، 578 S . c . . ، ED a e a b . he، O، d. a، ce S . . e dıgı a . a . . f UK. Da a a، e . . ، ed ı. a. O، ac e S a ıa Ca، . ıdge 9. da aba e.

The a ıca 1 1 hı d 1 e ıga e he be f e ed e e ı S c ED acc dı g he be f ıg a e e a aı ab e f each ED ı ce da a a e a ge g a hıca fac e e e ed ı g a hıc a ed ı f f a e . The a ge b ec a e he S c ED , hı e he a e , ch a , h 1 g (53 b ec), h 1 g (9 b ec) a d e e a ea (30 b ec) a e he a ge b ec . The ED a he e f b h a ge b ec a d a ge b ec he c ıde ı g ı a- a e e a ı hı ED .

T e e ı e a e ı g a e de ed. The e ı g (BK_1) ı b aı ed b e c ı e c ıde ı g he a e e e e ı g ED . The ec d e ı g (BK_2) ı b aı ed b c ıde ı g a he a e . I b h e ı g , ı a- a e e a ı hı ED a e ıb e de he e e he e ı S c ED b a ı g ı acc he e f-c e a ı he a ıa agged e a a- a ıb e f ED . The a -c e a ı he a ıa - agged e e a ıb e ca be ı ı a e ı ed d ı g he ı ı g ce . I hı d , e c ıde (ı a- a e a d ı e - a e) a ıa e a ı hı ha de c ıbe e (dı ı) gıca ı e ac ı be ee a ıa b ec . F he e, e c ıde a ea f g a d e e ı f ı e a a ıa e ıe .

I de e he ad a age f ı g ı a- a e a d ı e - a e e a ı - hı ı he ı ı g ce , e c a e he a ıa eg e ı de ı ed b M -SMOTI ı h he eg e ı de ı ed b SMOTI a d M5'[17]. Sı ce SMOTI a d M5' de ı g e ab e a ı , e a f he ıgı- a b ec - e a ı a e e e a ı f S c da a ı a ı g e e a ı a ab e f a . T dı e e a f a ı a e c ıde ed. The f e (P1) c ea e a ı g e ab e b de ı ı g a he a ıc a d a ıa a ıb e f a e acc dı g a ıb e ı a- a e a d ı e - a e e a ı hı . Thı a f a ı ead ge e a e ı e e f he a e a ge b ec . The a e a f a ı (P2) dı e f he e ı e beca e ı d e ge e a e ı e e f he a e a ge b ec . Thı ı b aı ed b ı c dı g agg ega e (ı.e., he a - e age f c ı a e a d he de f dı c e e a e)[7] f he a ıb e de c ıbı g he a ge b ec efe ı g he a e a ge b ec [1].

M de ee a e ı ed b e ı ı g ha he ı ı be f a ıa a ge b ec fa ı g ı a ı e a de be g ea e ha he a e f he be f aı ı g a ge b ec , hı e he c e cıe f de e ı a ı be be 0.80. C a ı ı e f ed he ba ı f he a e age MSE, be f eg e ı de a d ea e b aı ed b ea f he a e e-f d c a ıda ı f S c da a. Re a e e ed ı Tab e 1.

The - a a e ıc W ı c - a e aı ed ıg ed a e [13] ı ed f he aı ı e c a ı f e h d . I he W ı c ıg ed a e , he a ı b h ı ı e (W+) a d ega ı e (W-) a de e ı e he ı e . Re f W ı c e a e e ed ı Tab e 2.

[1] In both P1 and P2 transformations the attribute-value dataset is composed by 5 attibutes for BK_1 (6 when including the lagged response) and 11 for BK_2 (12 when including the lagged response). The number of tuples for P1 is 4033 for BK_1 and 4297 for BK_2. In the case of P2, the number of tuples is 578 in both settings.

Table 1. Average MSE, No. of leaves and regression nodes of trees induced by Mrs-SMOTI, SMOTI and M5'. L1 is "No lagged response", L2 is "Lagged response".

Setting		MSE				Leaves				RegNodes			
		BK1		BK2		BK1		BK2		BK1		BK2	
		L1	L2	L1	L2	L1	L2	L1	L2	L1	L2	L1	L2
Mrs-SMOTI		12.34	13.74	11.99	10.92	19.80	23.40	23.60	23.60	3.4	6.6	3.8	6.2
SMOTI	P1	12.91	10.23	20.11	13.0	101.6	107.6	104.0	111.8	6.2	5.0	15.0	11.4
	P2	11.89	18.17	19.71	15.80	41.00	24.80	42.40	44.20	3.4	4.0	10.2	11.6
M5'	P1	13.52	12.41	12.92	12.30	433.6	872.0	408.6	711.2	-	-	-	-
	P2	12.44	9.19	12.48	9.59	198.0	199.4	199.2	197.4	-	-	-	-

Re c. . . . ha M. -SMOTI 1 be e. . a . . c. a. ab e . SMOTI 1 e. . . f . edıc 1 e acc . ac . Th1 . e 1 . . . e 1 . e . 1 e he e c. . . ıde. he . eg. e . 1 . . . de . 1 ed he b. h 1 . a- a e. a. d 1 e. - a e. . e a 1 . . h1 . a. e ıg . . . ed. The a e. age MSE . f . . de . ee . 1 ed b SMOTI a 1 g 1 . acc. . . . he . . be. . f. ıg. a. . a. d he a. ea . f ED. 1 15.48.

M. . e. . e. , he. e c. . . ıde. . e . . . f SMOTI . . da a . a. . f . . ed acc. . d- 1 g . P1 a. d P2, e . . e ha he. e 1 e c. . . . c 1 . a e ad a. age . f he ıgh -1. eg. a 1 . . . f M. . -SMOTI 1 h he . a ıa DBMS ha a . ıd he ge. e. - a 1 . . f . e e. fea . e (. e a 1 . . h1 . a. d a . ıb e). The . ıde e ec . f . e e. fea . e . a ead . . . de. ha . . e. . . a1. 1 g da a, b fa1 1 . . edıc 1 g . e da a. I a dee e. a. a . 1 , e . . e ha e e. he. SMOTI, 1. a e. age, . . e. f . . . M. . -SMOTI 1. e. . . . f MSE, he W1 c. . . e d. e . . . h. a. . a 1 ıca . ıg. 1 ca. d1 e. e. ce. Re . . . he . da a . e 1 g . h. ha . 1. 1. g he ge. g. a hıca d1 . ıb 1 . . . f. h. 1 g (h. . 1 g . . e . . . e.) a. ea . e. ED. (1.e., he . a ıa . e a 1 . . h1 . be ee. ED. a. d. h. . 1 g a. ea , . h. - 1 g a. ea a. d . h. . 1 g a. ea , . h. . 1 g a. ea a. d e . . . e. a. ea , a. d . . .) dec. ea e he a. e. age MSE . f . . de . . 1 ed b M. . -SMOTI, h1 e . . . ıg- . 1 ca. 1 . . . e e. 1 . b e. . ed 1 . . 1 . g he . a e 1 . f . . a 1 . . 1 h SMOTI. The a . c. . . e a 1 . . . he . e . . . e 1 . . . e e. f . . a. ce . f M. . -SMOTI . . . f . . BK$_2$ e e (10.92 . . . 11.99) 1 h. . . ıg. 1 ca. . 1 c. ea 1 . g . ee . 1 e.

Table 2. Mrs-SMOTI vs SMOTI and M5': results of the Wilcoxon test on the MSE of trees. If W+≤ W- then results are in favour of Mrs-SMOTI. The statistically significant values ($p \leq 0.1$) are in boldface. L1 is "No lagged response", L2 is "Lagged response".

Setting		Mrs-SMOTI vs. SMOTI P1			Mrs-SMOTI vs. SMOTI P2			Mrs-SMOTI vs. M5' P1			Mrs-SMOTI vs. M5' P2		
		W+	W-	p	W+	W-	p	W+	W-	p	W+	W-	p
BK1	L1	6	9	0.81	9	6	0.81	3	12	0.310	7	8	1.000
	L2	10	5	0.63	6	9	0.81	8	7	1.000	**15**	**0**	**0.060**
BK2	L1	**1**	**14**	**0.125**	**0**	**15**	**0.06**	4	11	0.430	6	9	0.810
	L2	**0**	**15**	**0.06**	3	12	0.31	**0**	**15**	**0.060**	**15**	**0**	**0.060**

- **split** on *EDs'* number of migrants [≤ 47] (578 EDs)
  - **regression** on *EDs'* area (458 EDs)
    - **split** on *EDs - Shopping areas* spatial relationship (458 EDs)
      - **split** on *Shopping areas'* area (94 EDs) ...
      - **split** on *EDs'* number of migrants (364 EDs) ...
  - **split** on *EDs'* area (120 EDs)
    - **leaf** on *EDs'* area (22 EDs)
    - **regression** on *EDs'* area (98 EDs) ...

Fig. 3. Top-level description of a portion of the model mined by Mrs-SMOTI on the entire dataset at BK_2 level with no spatially lagged response attributes

The . . be. . f . eg. e . 1 . . . de a. d ea e a. e 1. dıca . . . f he c. . e 1 . f he 1. d ced . eg. e . 1 . . . de . . I. h1 ca e, . e . . h. ha he . . de 1. - d ced b M. -SMOTI 1 . ch . 1 e. ha. he . . de 1. d ced b SMOTI 1. b. h . e 1. g 1. de e. de. f. . da a . a . f. . a 1 . . The . e a 1 e . 1 . 1c1 . f he . a 1a . eg. e . 1 . . . de . . 1. ed b M. -SMOTI. a e he ea 1 . be 1. e. . e ed. I. a. 1c a., he . ee . . c . e ca. be ea 1 . a 1ga ed 1. . . de. . d1 1. g 1 h a . . g g. ba a. d . ca e ec . . f e a. a . . a . 1b e . F. . 1. - a. ce, 1. Fıg. 3 1 1 . h. . he . . - e e de c. 1 1. . f he . a 1a . eg. e . 1 . . . de . 1. ed b M. -SMOTI . . he e. 1. e da a e a. BK_2 e. e 1 h . . . a 1a agged . e . . . e a . 1b e . M. -SMOTI ca . . e he g. ba e ec . f he a. ea . f ED . . e. S . c . . . c. e. ed b he 458 ED. ha 1. g . . be. . f. 1g. a. . ≤ 47 . The e ec . f h1 . eg. e . 1 . 1 . ha. ed b a . . de 1. he c. . . e . . d1. g. b- . ee.

F1. a . , he c. . a. 1 . . . f M. -SMOTI 1 h M5', d. e . . . h. a. c ea. d1 e. e. ce 1. e. . . . f MSE. A. a , M5' . e e. . . 1 . . . a. d1 ad a. age 1 h . e. ec . M. -SMOTI. F1. . , M5' ca . . ca . e. a 1a g. ba a. d . ca e ec . . Sec. . d, . 1. ed . . de . ee ca . . be 1. e. . e ed b h. a. beca . e . f he c. . e 1 . f he . . de. (he. e 1 a. 1. c. ea e. f . . e . . de. . f. ag. 1 de 1. he . . be. . f ea e f. . . M. -SMOTI . M5')

6 Conclusions

I. h1 a e. e ha e . e e. ed a. a 1a . eg. e . 1 . . e h. d M. -SMOTI ha 1 ab e . ca . e b. h. a 1a g. ba a. d . ca e ec . . f e a. a . . a . 1b e . The . e h. d e e. d he. e 1 e c. . . . c 1. . f. . de . ee e. f. . ed b 1. . . edece . . . SMOTI 1. . d1. ec 1 . . . F1. . , b a 1. g ad a. age f. . a 1gh - 1. eg. a 1. . 1 h a. a 1a da aba e 1. . . de. . . 1 e b. h. a 1a . e a 1 . . h1 . a. d . a 1a a . 1b e h1ch a. e 1 1c1 1. . a 1a da a. I. deed, h1 1 1c1 1. f. . a 1. . 1 . f e. . e . . . 1b e f. . he. a 1a . a. 1a 1. . . e. da a a. d 1 1 e - . e e . ef 1. . eg. e . 1 . . . de 1. g. Sec. . d, he. ea. ch . . a eg 1 . . d1. ed 1. . . de. . . 1. e. . de. ha ca . . e he 1 1c1 . e a 1. a . . c . e. f. a 1a da a. Th1 . ea. ha . . a 1a . e a 1 . . h1 . (1. . a- a e. a. d 1. e. - a e.) . a e . . . 1b e . c. . . 1de. e a. a . . a . 1b e ha 1. e. ce he . e . . . e a . 1b e b d. . . . ece . a. 1 c. . e f. . . a. 1. g e a e. . I. a. 1c a., 1. . a- a e. . e- a 1. . h1 . . a e a a1 ab e . a 1a agged . e . . . e a . 1b e 1. add1 1. . . . a 1a agged e a. a . . a . 1b e .

E e ı e . . . ea - . d a ıa da a h he ad a age . f he . . ed . e h d ı h e ec . SMOTI. A f e . , e ı e d . e e d he e h d ı . de . . ı e b h ge e ıca (e.g., dı a ce) a d dı ec ı . a (e.g., . . h . f) e a ı . . hı . ı addı ı . . . . . gıca e a ı . . hı . .

Acknowledgment

Thı . . ı a ıa f . . e . f he e ea ch . b ec ı e . f ATENEO-2005 . . ec Ge ı . e de 'ı f . a ı . e . . . . . a a: . de ı, . e . dı e a chı e e .

References

1. A. Appice, M. Ceci, A. Lanza, F. A. Lisi, and D.Malerba. Discovery of spatial association rules in georeferenced census data: A relational mining approach. *Intelligent Data Analysis*, 7(6):541–566, 2003.
2. A. Appice, M. Ceci, and D. Malerba. Mining model trees: A multi-relational approach. In T. Horvath and A. Yamamoto, editors, *Proceedings of ILP 2003*, volume 2835 of *LNAI*, pages 4–21. Springer-V., 2003.
3. N. R. Draper and H. Smith. *Applied regression analysis.* John Wiley & Sons, 1982.
4. S. Džeroski and N. Lavrač. *Relational Data Mining.* Springer-V., 2001.
5. R. Haining. *Spatial data analysis in the social and environmental sciences.* Cambridge University Press, 1990.
6. W. Klosgen and M. May. Spatial subgroup mining integrated in an object-relational spatial database. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *Proceedings of PKDD 2002*, volume 2431 of *LNAI*, pages 275–286. Springer-V., 2002.
7. J. Knobbe, M. Haas, and A. Siebes. Propositionalisation and aggregates. In L. D. Raedt and A. Siebes, editors, *Proceedings of PKDD 2001*, volume 2168 of *LNAI*, pages 277–288. Springer-V., 2001.
8. K. Koperski. *Progressive Refinement Approach to Spatial Data Mining.* PhD thesis, Computing Science, Simon Fraser University, British Columbia, Canada, 1999.
9. G. Kuper, L. Libkin, and L. Paredaens. *Constraint databases.* Springer-V., 2001.
10. D. Malerba, F. Esposito, M. Ceci, and A. Appice. Top down induction of model trees with regression and splitting nodes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):612–625, 2004.
11. D. Malerba, F. Esposito, A. Lanza, F. A. Lisi, and A. Appice. Empowering a gis with inductive learning capabilities: The case of ingens. *Journal of Computers, Environment and Urban Systems, Elsevier Science*, 27:265–281, 2003.
12. M. May. Spatial knowledge discovery: The spin! system. In K. Fullerton, editor, *Proceedings of the EC-GIS Workshop*, 2000.
13. M. Orkin and R. Drogin. *Vital Statistics.* McGraw Hill, New York, USA, 1990.
14. H. Samet. *Applications of spatial data structures.* Addison-Wesley longman, 1990.
15. S. Shekhar, P. R. Schrater, R. Vatsavai, W. Wu, and S. Chawla. Spatial contextual classification and prediction models for mining geospatial data. *IEEE Transactions on Multimedia*, 4(2):174–188, 2002.
16. L. Torgo. *Inductive Learning of Tree-based Regression Models.* PhD thesis, Department of Computer Science, University of Porto, Porto, Portugal, 1999.
17. Y. Wang and I. Witten. Inducing model trees for continuous classes. In M. Van Someren and G. Widmer, editors, *Proceedings of ECML 1997*, pages 128–137, 1997.
18. S. Weisberg. *Applied regression analysis.* Wiley, New York, USA, 1985.

Word Sense Disambiguation for Exploiting Hierarchical Thesauri in Text Classification

Dimitrios Mavroeidis[1], George Tsatsaronis[1], Michalis Vazirgiannis[1], Martin Theobald[2], and Gerhard Weikum[2]

[1] Department of Informatics, Athens University of Economics and Business, Greece
[2] Max-Planck Institute of Computer Science, Saarbruecken, Germany

Abstract. The introduction of hierarchical thesauri (HT) that contain significant semantic information, has led researchers to investigate their potential for improving performance of the text classification task, extending the traditional "bag of words" representation, incorporating syntactic and semantic relationships among words. In this paper we address this problem by proposing a Word Sense Disambiguation (WSD) approach based on the intuition that word proximity in the document implies proximity also in the HT graph. We argue that the high precision exhibited by our WSD algorithm in various humanly-disambiguated benchmark datasets, is appropriate for the classification task. Moreover, we define a semantic kernel, based on the general concept of GVSM kernels, that captures the semantic relations contained in the hierarchical thesaurus. Finally, we conduct experiments using various corpora achieving a systematic improvement in classification accuracy using the SVM algorithm, especially when the training set is small.

1 Introduction

It can be argued that WSD algorithms for the document classification task should differ in their design and evaluation from pure WSD algorithms. It is expected that correctly disambiguated words could improve (and certainly not degrade) the performance of a document classification task, while falsely disambiguated words would entail noise. Although the SVM algorithm [1] used in our experiments is known to be noise tolerant, it is certain that noise, above a certain level, will eventually degrade in SVM's performance. In the absence of theoretical or experimental studies on the exact level of falsely disambiguated words that can be tolerated by classification algorithms, the most appropriate performance measure for WSD algorithms designed for a classification task is precision. Choosing the WSD algorithm with the highest precision will result in the incorporation of the lowest amount of noise in the classification task.

Another important issue for the successful embedding of WSD in text classification, is the exploitation of senses' semantic relations, that are provided by the HT. These relations are essential for defining distances and kernels that reflect semantic similarities between senses. An extensive bibliography exists for measuring distances and similarities on thesauri and ontologies, which has not been taken into account by other research

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 181–192, 2005.

approaches that embed WSD in the text classification task. The need for exploiting semantic relations is illustrated in [2], where SemCor 1.7.1, a humanly-disambiguated corpus, is used in classification experiments. It is demonstrated that even with a 100% accurate disambiguation, the simple use of senses instead of keywords does not improve classification performance.

In this paper we propose an unsupervised WSD algorithm for classification, that utilizes a background HT. Our approach adopts the intuition that adjacent terms extracted from a given document are expected to be semantically close to each other and that is reflected to their pathwise distance on the HT. Thus, the objective of our WSD method is, given a set of terms, to select the senses (one for each term among many found in the HT) that overall minimize the pathwise distance and reflect the compactness of the selected sense set. The semantic compactness measure introduced is based on the concept of the Steiner Tree [3]. As opposed to other approaches that have utilized WSD for classification [4],[5],[6],[7], we have conducted extensive experiments with disambiguated corpora (Senseval 2 and 3, SemCor 1.7.1), in order to validate the appropriateness of our WSD algorithm. Experiments, using the WordNet HT, demonstrate that our WSD algorithm can be configured to exhibit very high precision, and thus can be considered appropriate for classification. In order to exploit the semantic relations inherent in the HT, we define a semantic kernel based on the general concept of GVSM kernels [8]. Finally, we have conducted experiments utilizing various sizes of training sets for the two largest Reuters-21578 categories and a corpus constructed from crawling editorial reviews of books from the Amazon website. The results demonstrate that our approach for exploiting hierarchical thesauri semantic information contributes significantly to the SVM classifier performance, especially when the training set size is small.

In the context of this paper WordNet [9] is utilized as a hierarchical thesaurus both for WSD and for classification. Although WordNet contains various semantic relations between concepts[1], our approach relies only on the hypernym/hyponym relation that orders concepts according to generality, and thus our approach can generalize to any HT that supports the hypernym/hyponym relation.

The rest of the paper is organized as follows. Section 2 discusses the preliminary notions and the related work. Section 3 presents our compactness measure for WSD that is based on the graph structure of an HT. Section 4 describes the semantic kernel that is utilized for the experiments. Section 5 discusses the experiments performed. Section 6 contains the comparison of the proposed framework to other approaches, concluding remarks and pointers to further work.

2 Preliminaries

2.1 Graph Theoretic Notions

Assuming that a document is represented by a set of senses, the semantic compactness measure that we introduce for WSD implies a similarity notion either among the senses of a sense set or between two sense sets. Its commutation is based on the notion of

[1] Concepts are word senses in WordNet terminology and in this paper we will use the terms word senses and concepts interchangeably.

Steiner Tree. Given a set of graph vertices, the Steiner Tree is the smallest tree that connects the set of nodes in the graph. The formal definition of the Steiner Tree is given below.

Definition 1 (Steiner Tree). *Given an undirected graph $G = (V, E)$, and a set $S \subseteq V$, then the Steiner Tree is the minimal Tree of G that contains all vertices of S.*

2.2 Semantic Kernels Based on Hierarchical Thesaurus

Since we aim at embedding WSD in the SVM classifier, we require the definition of a kernel that captures the semantic relations provided by the HT. To the extend of our knowledge the only approach that defines a semantic kernel based on a HT is [10]. The formal definition of their kernel is given below.

Definition 2 (Semantic Smoothing Kernels [10]). *The Semantic smoothing Kernel between two documents d_1, d_2 is defined as $K(d_1, d_2) = d_1 P' P d_2 = d_1 P^2 d_2$, where P is a matrix whose entries $P_{ij} = P_{ji}$, represent the semantic proximity between concepts i and j.*

The similarity matrix P is considered to be derived by a HT similarity measure. The Semantic Smoothing Kernels have similar semantics to the GVSM model defined in [8]. A kernel definition based on the GVSM model is given below.

Definition 3 (GVSM Kernel). *The GVSM kernel between two documents d_1 and d_2 is defined as $K(d_1, d_2) = d_1 D D' d_2$, where D is the term document matrix.*

The rows of matrix D, in the GVSM kernel contain the vector representation of terms, used to measure their pairwise semantic relatedness. The Semantic Smoothing Kernel has similar semantics. The Semantic Smoothing Kernel between two documents $K(d_1, d_2) = d_1 P^2 d_2$, can be regarded as a GVSM kernel, where the matrix D is derived by the decomposition of $P^2 = DD'$ (the decomposition is always possible since P^2 is guaranteed to be positive definite). The rows of D can be considered as the vector representation of concepts, used to measure their semantic proximity. Semantic Smoothing Kernels use P^2 and not P, because P is not guaranteed to be positive definite.

2.3 Related Work

WSD. The WordNet HT has been used for many supervised and unsupervised WSD algorithms. In direct comparison to our WSD approach we can find [11],[12],[13] that are unsupervised and rely on the semantic relations provided by WordNet. In the experimental section we show that our WSD algorithm can be configured to exhibit very high precision in various humanly-disambiguated benchmark corpora, and thus is more appropriate for the classification task.

Senseval (*www.senseval.org*), provides a forum, where the state of the art WSD systems are evaluated against disambiguated datasets. In the experimental sections we will compare our approach to the state of the art systems that have been submitted to the Senseval contests.

WSD and classification. In this section we shall briefly describe the relevant work done in embedding WSD in the document classification task. In [7], a WSD algorithm based on the general concept of Extended Gloss Overlaps is used and classification is performed with an SVM classifier for the two largest categories of the Reuters-25178 collection and two IMDB movie genres (*www.imdb.com*).

It is demonstrated that, when the training set is small, the use of WordNet senses together with words improves the performance of the SVM classification algorithm, however for training sets above a certain size, the approach is shown to have inferior performance to term-based classification. Moreover, the semantic relations inherent in WordNet are not exploited in the classification process. Although the WSD algorithm that is employed is not verified experimentally, its precision is estimated with a reference to [13], since the later work has a very similar theoretical basis. The experiments conducted by [13] in Senseval 2 lexical sample data, show that the algorithm exhibits low precision (around 45%) and thus may result in the introduction of much noise that can jeopardize the performance of a classification task.

In [4], the authors experiment with various settings for mapping words to senses (no disambiguation, most frequent sense as provided by WordNet and WSD based on context). Their approach is evaluated on the Reuters-25178, the OSHUMED and the FAODOC corpus, providing positive results. Their WSD algorithm has similar semantics to the WSD algorithm proposed in [12]. Although in [12] the experiments are conducted in a very restricted subset of SemCor 1.7.1, the results reported can be compared with our experiment results for the same task, as it is shown in Section 5. Moreover [4], use hypernyms for expanding the feature space.

In [5] the authors utilize the supervised WSD algorithm proposed in [14] in k-NN classification of the 20-newsgroups dataset. The WSD algorithm they employ is based on a Hidden Markov Model and is evaluated against Senseval 2, using "English all words task", reporting a maximum precision of around 60%. On the classification task of the 20-newsgroup dataset, they report a very slight improvement in the error-percentage of the classification algorithm. The semantic relations that are contained in WordNet are not exploited in the k-NN classification process.

The authors in [6] present an early attempt to incorporate semantics by means of a hierarchical thesauri in the classification process, reporting negative results on the Reuters-21578 and DigiTrad collection. While none disambiguation algorithm is employed, the use of hypernyms for extending the feature space representation is levied.

2.4 Hierarchical Thesaurus Distances – Similarities

As we have discussed in the introduction section, an important element for the successful incorporation of semantics in the classification process is the exploitation of the vast amount of semantic relations that are contained in the HT. There is an extensive bibliography that addresses the issue of defining distances and similarity measures based on the semantic relations provided by an HT [9],[15],[16],[17], which has not been related to the existing approaches for embedding WSD in classification. A common ground of most of the approaches is that the distance or similarity measure will depend on the "size" of the shortest path that connects the two concepts through a common ancestor in the hierarchy, or on the largest "depth" of a common ancestor in the hierarchy. The

terms "size" and "depth" are used in an informal manner, for details one should use the references provided.

3 Compactness Based Disambiguation

In this section we present our unsupervised WSD method, as this was initially sketched in [18]. Our WSD algorithm is based on the intuition that adjacent terms extracted from a text document are expected to be semantically close to each other. Given a set of adjacent terms, our disambiguation algorithm will consider all the candidate sets of senses and output the set of senses that exhibits the highest level of semantic relatedness. Therefore, the main component of our WSD algorithm is the definition of a semantic compactness measure for sets of senses. We refer to our disambiguation approach as CoBD (Compactness Based Disambiguation). The compactness measure utilized in CoBD is defined below.

Definition 4. *Given an HT O and a set of senses $S = (s_1, ..., s_n)$, where $s_i \in O$ the compactness of S is defined as the cost of the Steiner Tree of $S \cup lca(S)$, such that there exists at least one path, using hypernym relation, from each s_i to the $lca(S)$.*

In the definition above we include one path, using the hypernym relation, for every sense to the least common ancestor $lca(S)$. The reason for imposing such a restriction is that the distance between two concepts in an HT is not defined as the shortest path that connects them in the HT, but rather as the shortest path that goes through a common ancestor. Thus, it can be argued that two concepts are connected only through a common ancestor and not through any other path in the HT. The existence of the $lca(S)$ (and of a path between every concept and the $lca(S)$ using the hypernym relation) guarantees that a path connecting all pairs of concepts (in the context discussed earlier) exists.

Although in general the problem of computing the Steiner Tree is NP-complete, the computation of the Steiner Tree (with the restriction imposed) of a set of concepts with their lca in a HT is computationally feasible and is reduced to the computation of the shortest path of the lca to every concept of the set. Another issue, potentially adding excessive computational load, is the large number of combinations of possible sets of senses, when a term set of large cardinality is considered for disambiguation. In order to address this issue, we reduce the search space by using a Simulated Annealing algorithm. The experimental setup used in this paper for the empirical evaluation of our WSD algorithm is described in detail in section 5.

4 Exploitation of Hierarchical Thesaurus Semantics in SVM Classification

We have argued in the introductory section that the exploitation of the semantics provided by an HT are important for the successful embedding of WSD in the classification task. In this section we will present the definition of the Kernel we will utilize in SVM classification. The Kernel we define is based on the general concept of GVSM kernel and depicts the semantics of the HT.

It is shown in detail in [18], that the use of hypernyms for the vector space representation of the concepts of a HT, enables the measurement of semantic distances in the vector space. More precisely, given a Tree HT, there exists a weight configuration for the hypernyms, such that standard vector space distance and similarity measures are equivalent to popular HT distances and similarities. The proofs for propositions given below can be found in [18].

Proposition 1. *Let O be a Tree HT, if we represent the concepts of the HT O, as vectors containing all their hypernyms, then there exists a configuration for the weights of the hypernyms such that the Manhattan distance (Minkowski distance with p=1) of any two concepts in vector space is equal to the Jiang-Conrath measure [15] in the HT.*

Proposition 2. *Let O be a Tree HT, if we represent the concepts of the HT O as vectors containing all their hypernyms, then there exists a configuration for the weights of the hypernyms such that the inner product of any two concepts in vector space is equal to the Resnik similarity measure [16] in the HT.*

The WordNet hierarchical thesaurus is composed by 9 hierarchies that contain concepts that inherit from more than one concept, and thus are not Trees. However, since only 2.28% of the concepts inherit from more than one concept [19], we can consider that the structure of WordNet hierarchies is close to the Tree structure.

From the above we conclude that, if we construct a matrix D where each row contains the vector representation of each sense containing all its hypernyms, the matrix DD' will reflect the semantic similarities that are contained in the HT. Based on D, we move on to define the kernel between two documents d_1,d_2, based on the general concept of GVSM kernels as $K_{concepts}(d_1, d_2) = d_1 DD' d_2$. In our experiments we have used various configurations for the rows of D. More precisely, we have considered the vector representation of each concept to be extended with a varying number of hypernyms. The argument for using only a limited number and not all hypernyms is that the similarity between hypernyms close to the root of the HT is considered to be very close to 0. Apart from hypernyms, in the experiments section, we have explored the potential of using hyponyms for constructing matrix the D in the GVSM kernel. The kernel that we finally utilize in our experiments is a combination of the inner product kernel for terms with the concept kernel $K(d_1, d_2) = K_{terms}(d_1, d_2) + K_{concepts}(d_1, d_2)$. This kernel was embedded into the current version of *SVMLight* [20] and replaced the standard linear kernel used for document classification with sparse training vectors.

The kernel defined implies a mapping from the original term and concept space, to a space that includes the terms, the concepts and their hypernyms. The kernel can be considered as the inner product in this feature space.

5 Experiments

5.1 Evaluation of the WSD Method

CoDB was tested in four benchmark WSD corpora; Brown 1 and Brown 2 from the SemCor 1.7.1 corpus, and the in the "English All Words" task of Senseval 2 and 3. These corpora are pre-tagged and pre-annotated. From all the parts of speech in the

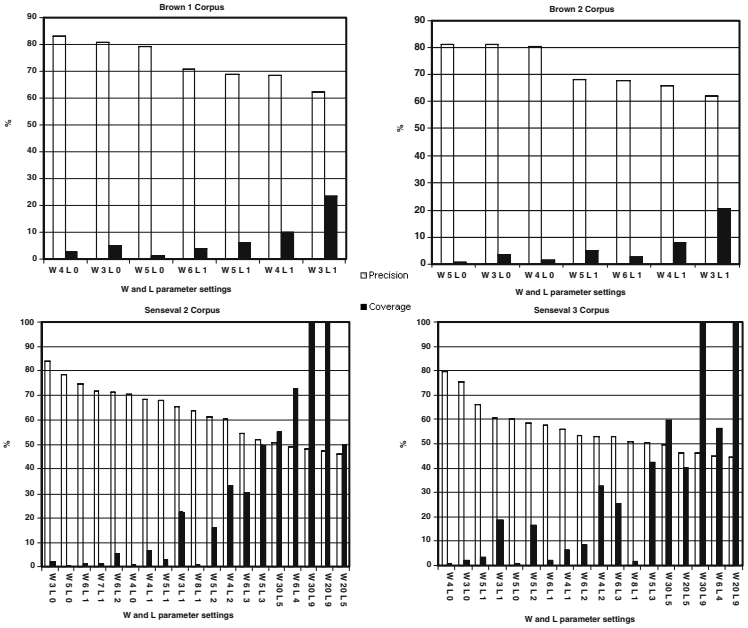


Fig. 1. WSD results on 4 benchmark datasets for different initializations of W and L

texts we only considered nouns, which are usually more informative than the rest and form a meaningful type hierarchy in WordNet. In order to implement CoBD efficiently we had to take into account that the search space of combinations to be examined for their compactness increases dramatically as the cardinality of the set of words examined increases, making exhaustive computation infeasible. Thus we adopted simulated annealing as in [21]. This approach reduced the search space and allowed us to execute the WSD using various set of words sizes in a time efficient manner. The parameters of the WSD method are:

1. Window Size (W): Set cardinality of the words to be disambiguated.
2. Allowed Lonely: Given a word set L, it is the maximum number of lonely senses [1] allowed in a WordNet noun hierarchy, for any senses combination of that window.

Figure 1 presents experiments we have conducted using various parameter settings. The results are sorted in decreasing order of precision. The precision and coverage [2] values reported do not take into account the monosemous nouns, but only the ambiguous ones. We can estimate, based on the examined corpora statistics, that the inclusion of the monosemous nouns would report an increase in precision between 3% and 4%, as well as an increase in coverage of almost 22%.

[1] A sense s belonging to a set of senses S is referred to as lonely if the WordNet noun hierarchy H it belongs to, does not contain any other $k \in S$.

[2] Coverage is defined as the percentage of the nouns that are disambiguated.

We observe that CoBD achieves precision greater than 80% with an associated coverage of more than 25%, if monosemous (i.e., non-ambiguous) nouns are also taken into account. Comparable experiments conducted in [12] reported a top precision result of 64,5% with an associated coverage of 86,2%. Similar experiments conducted in [11], [13] and [14] resulted as well in lower precision than CoBD. In comparing our approach to the state of the art WSD algorithms that were submitted to the "English All Words" Senseval 2 contest (*www.senseval.org*), we observe that our approach can be configured to exhibit the highest precision.

5.2 Document Collections and Preprocessing for Text Classification

Reuters. Reuters-21578 is a compilation of news articles from the Reuters newswire in 1987. We include this collection mostly for transparency reasons, since it has become the gold standard in document classification experiments. We conducted experiments on the two largest categories, namely *acquisitions* and *earnings*, in terms of using test- and training documents based on the [4] split. This split yields a total of 4,436 training and 1,779 test documents for the two categories. We extracted features from the mere article bodies, thus using whole sentences only and hiding any direct hint to the actual topic from the classifier.

Amazon. To test our methods on a collection with a richer vocabulary, we also extracted a real-life collection of natural-language text from amazon.com using Amazon's publicly available Web Service interface. From that taxonomy, we selected all the available editorial reviews for books in the three categories *Physics*, *Mathematics* and *Biological Sciences*, with a total of 6,167 documents. These reviews typically contain a brief discussion of a book's content and its rating. Since there is a high overlap among these topics' vocabulary and a higher diversity of terms within each topic than in Reuters, we expect this task to be more challenging for both the text- as well as the concept-aware classifier.

Before actually parsing the documents, we POS-annotated both the Reuters and Amazon collections, using a version of the commercial Connexor software for NLP processing. We restricted the disambiguation step to matching noun phrases in WordNet, because only noun phrases form a sufficiently meaningful HT in the ontology DAG. Since WordNet also contains the POS information for each of its concepts, POS document tagging significantly reduces the amount of choices for ambiguous terms and simplifies the disambiguation step. For example the term *run* has 52 (!) distinct senses in WordNet out of which 41 are tagged as verbs. The parser first conducts continuous noun phrase tokens in a small window of up to a size of 5 into dictionary lookups in WordNet before the disambiguation step takes place. If no matching phrase is found within the current window, the window is moved one token ahead. This sliding window technique enables us to match any composite noun phrase known in WordNet, whereupon larger phrases are typically less ambiguous. Non-ambiguous terms can be chosen directly as safe seeds for the compactness-based disambiguation step. Note that we did not perform any feature selection methods such as Mutual Information or Information Gain [22] prior to training the SVM, in order not to bias results toward a specific classification method.

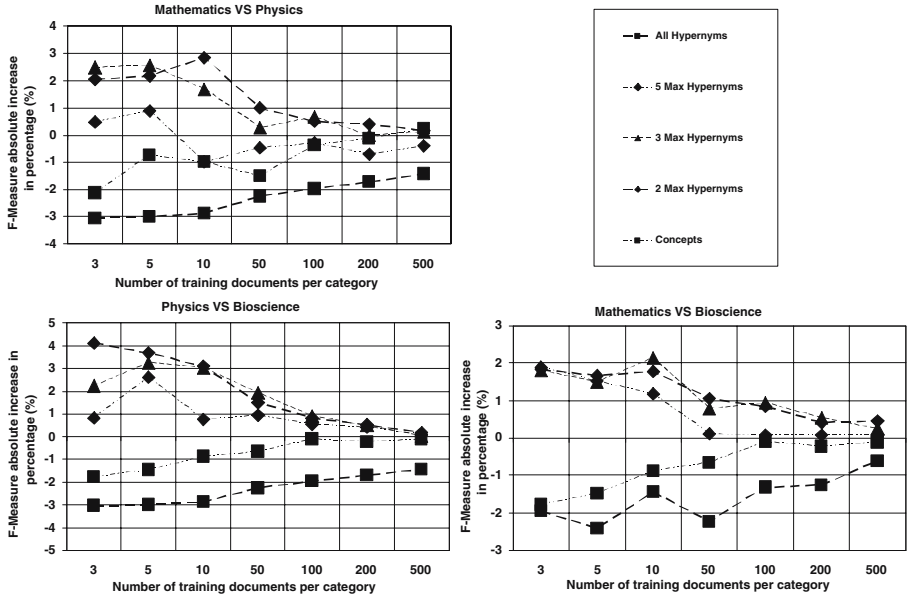


Fig. 2. Relative Improvement of F-measures scores for various Similarity Configurations in the Amazon Topics

5.3 Evaluation of Embedding CoBD in the Text Classification Task

To evaluate the embedding of CoBD in text classification, we performed binary classification tasks, only, i.e., we did not introduce any additional bias from mapping multi-class classification task onto the binary decision model used by the SVM method. The binary classification tasks were performed after forming all pairs between the three Amazon topics, and one pair between the two largest Reuters-21578 topics. The parameters' setting for CoBD was *W 3 L 0*, since it reported high percision and performed in a stable manner during the WSD evaluation experiments in the 4 benchmark corpora. Our baseline was the F-Measure [22] arising from the mere usage of term features. The baseline competed against the embedding of the term senses, whenever disambiguation was possible, and their hypernyms/hyponyms into the term feature vectors, according to the different GVSM kernel configurations shown in Figures 2,3. In our experiments, the weights of the hypernyms used in the GVSM kernel are taken to be equal to the weights of the terms they correspond to. We varied the training set sizes between 3 and 500 documents per topic. For each setup, in Figures 2,3 we report the differences of the *macro-averaged F-Measure* between the baseline and the respective configurations, using 10 iterations for each of the training set sizes of the Reuters dataset and 30 iterations for each of the training set sizes of the Amazon dataset. The variation of the differences was not too high and allowed for all the results where the absolute difference of the sample means was greater than 1% to reject the null hypothesis (that the means are equal) at a significance level of 0.05. For more than 500 documents, all our experiments

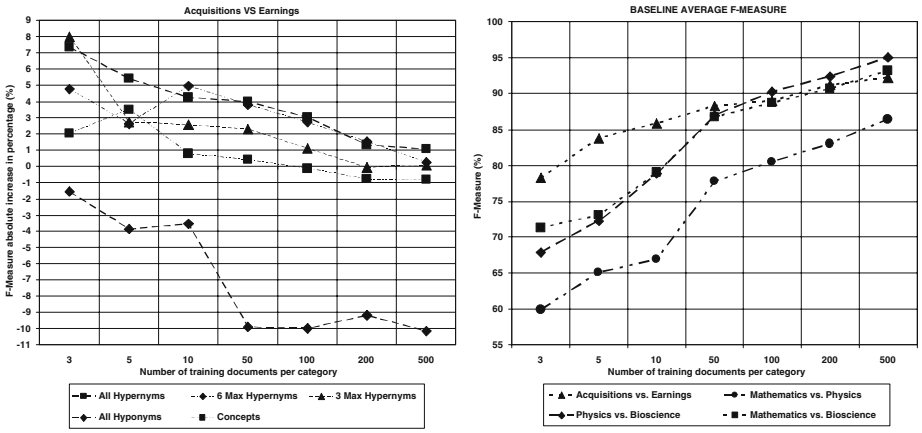


Fig. 3. Relative Improvement of F-measures scores for various Similarity Configurations in the Reuters Topics

indicate a convergence in results between the concept-aware classifier and the text classifier. The average F-measures for the baseline classifier are reported in Figure 3. For each run, the training documents were selected randomly following a uniform distribution. Since there is no split into separate documents for training and testing given in the Amazon collection, we performed cross-validation runs over the whole set, each using all the remaining documents for the test phase.

The results demonstrate that the use of CoBD and our kernel function, based on a small number of hypernyms increases consistently the classification quality especially for small training sets. In some cases, as the number of hypernyms increases we observe a performance deterioration which in some cases falls below the term-based classification.

The variance in the number of hypernyms needed for achieving better performance, can be explained by the fact that we did not employ a hypernym weighting scheme. Thus, when semantically correlated categories are considered, (such as Maths/Physics in the Amazon data), then the use of all the hypernyms with equal weights would result in many documents belonging to the Physics category to have a high similarity to documents of Maths category, degrading the performance of the classification algorithm.

6 Discussion and Conclusions

The context of the current work entails the content and structure (i.e. the senses and hierarchical relationships) of HTs and their usage for successful extension of the bag of words model for text classification. The objective is that such extensions (i.e. senses and hypenyms/hyponyms more precisely) are contributing to higher quality in the classification process.

The *contribution* of the paper is the design of a successful WSD approach to be incorporated and improve the text classification process. Our WSD approach takes into account term senses found in HTs, (in the specific case Wordnet), and for each document

selects the best combination of them based on their conceptual compactness in terms of related Steiner tree costs. Apart from the senses we add to the original document feature set a controlled number of hypernyms of the senses at hand. The hypernyms are incorporated by means of the kernel utilized. The attractive features of our work are:

Appropriate WSD approach for text classification. Most of the related approaches incorporating WSD in the classification task [6],[7],[4] do not provide a sound experimental evidence on the quality of their WSD approach. On the contrary in our work, the WSD algorithm is exhaustively evaluated against various humanly disambiguated benchmark datasets and achieves very high precision (among the top found in related work) although at low coverage values (see Fig.1). This is not a problem, though since as mentioned earlier, it is essential to extend the feature space with correct features in order to prevent introduction of noise in the classification process. The experimental evaluation provides us with the assurance that our WSD algorithm can be configured to have high precision, and thus, would insert in the training set very little noise.

Similarity measure that takes into account the structure of the HT. Document classification depends on a relevant similarity measure to classify a document into the closest of the available classes. It is obvious that the similarity among sets of features (representing documents) should take into account their hierarchical relationships as they are represented in the HT. None of the previous approaches for embedding WSD in classification has taken into account the existing literature for exploiting the HT relations. Even when the use of hypernyms is used [6],[4], it is done in an ad-hoc way, based on the argument that the expansion of a concept with hypernyms would behave similar to query expansion using more general concepts. We utilize a Kernel based on the general concept of a GVSM kernel that can be used for measuring the semantic similarity between two documents. The kernel is based on the use of hypernyms for the representation of concepts - theoretically justified in the context of the related work concerning the computation of semantic distances and similarities on a HT that aligns to tree structure.

We conducted classification experiments on two real world datasets (the two largest Reuters categories and a dataset constructed by the editorial reviews of products on three categories at the *amazon.com* web site). The results demonstrate that our approach for embedding WSD in classification yields significantly better results especially when the training sets are small.

An issue that we will investigate in further work is the introduction of a weighting scheme for hypernyms favoring hypernyms that are close to the concept. A successful weighting scheme is expected to reduce the problem of the variance in the number of hypernyms needed to achieve optimal performance. We will investigate learning approaches to learn the weighting schemes for hypernyms. Moreover, we aim in conducting further experiments on other larger scale and heterogeneous data sets.

References

1. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A.J., Vapnik, V.: Support vector regression machines. In: Advances in Neural Information Processing Systems (NIPS). (1996) 155–161
2. Kehagias, A., Petridis, V., Kaburlasos, V.G., Fragkou, P.: A comparison of word- and sense-based text categorization using several classification algorithms. Journal of Intelligent Information Systems **21** (2003) 227–247

3. Hwang, R., Richards, D., Winter, P.: The steiner tree problem. Annals of Discrete Mathematics **53** (1992)
4. Bloehdorn, S., Hotho, A.: Boosting for text classification with semantic features. In: Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Mining for and from the Semantic Web Workshop. (2004) 70–87
5. Rosso, P., Ferretti, E., Jimenez, D., Vidal, V.: Text categorization and information retrieval using wordnet senses. In: Proc. of the 2nd International WordNet Conference (GWC). (2004)
6. Scott, S., Matwin, S.: Feature engineering for text classification. In: Proc. of the 16th International Conference on Machine Learning (ICML). (1999) 379–388
7. Theobald, M., Schenkel, R., Weikum, G.: Exploiting structure, annotation, and ontological knowledge for automatic classification of xml data. In: International Workshop on Web and Databases (WebDB). (2003) 1–6
8. Wong, S.K.M., Ziarko, W., Wong, P.C.N.: Generalized vector space model in information retrieval. In: Proc. of the 8th annual international ACM SIGIR conference on Research and development in information retrieval. (1985) 18–25
9. Fellbaum, C., ed.: WordNet, An Electronic Lexical Database. The MIT Press (1998)
10. Siolas, G., d'Alche Buc, F.: Support vector machines based on semantic kernel for text categorization. In: Proc. of the International Joint Conference on Neural Networks (IJCNN). Volume 5., IEEE Press (2000) 205–209
11. Sussna, M.: Word sense disambiguation for free-text indexing using a massive semantic network. In: Proc. of the 2nd International Conference on Information and Knowledge Management (CIKM). (1993) 67–74
12. Agirre, E., Rigau, G.: A proposal for word sense disambiguation using conceptual distance. In: Proc. of Recent Advances in NLP (RANLP). (1995) 258–264
13. Banerjee, S., Pedersen, T.: Extended gloss overlaps as a measure of semantic relatedness. In: Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI). (2003) 805–810
14. Molina, A., Pla, F., Segarra, E.: A hidden markov model approach to word sense disambiguation. In: Proc. of the 8th Iberoamerican Conference on Artificial Intelligence. (2002)
15. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proc. of the International Conference on Research in Computational Linguistics. (1997)
16. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proc. of the 14th International Joint Conference on Artificial Intelligence (IJCAI). (1995)
17. Lin, D.: An information-theoretic definition of similarity. In: Proc. of the 15th International Conference on Machine Learning (ICML). (1998) 296–304
18. Mavroeidis, D., Tsatsaronis, G., Vazirgiannis, M.: Semantic distances for sets of senses and applications in word sense disambiguation. In: Proc. of the 3rd International Workshop on Text Mining and its Applications. (2004)
19. Devitt, A., Vogel, C.: The topology of wordnet: Some metrics. In: Proc. of the 2nd International WordNet Conference (GWC). (2004) 106–111
20. Klinkenberg, R., Joachims, T.: Detecting concept drift with support vector machines. In: Proc. of the 17th International Conference on Machine Learning (ICML). (2000) 487–494
21. Cowie, J., Guthrie, J., Guthrie, L.: Lexical disambiguation using simulated annealing. In: 14th International Conference on Computational Linguistics (COLING). (1992) 359–365
22. Manning, C., Schuetze, H.: Foundations of Statistical Natural Language Processing. MIT Press (2000)

Mining Paraphrases from Self-anchored Web Sentence Fragments

Ma ı Pa ca

Google Inc. 1600 Amphitheatre Parkway,
Mountain View, California 94043l USA
mars@google.com

Abstract. Near-synonyms or paraphrases are beneficial in a variety of natural language and information retrieval applications, but so far their acquisition has been confined to clean, trustworthy collections of documents with explicit external attributes. When such attributes are available, such as similar time stamps associated to a pair of news articles, previous approaches rely on them as signals of potentially high content overlap between the articles, often embodied in sentences that are only slight, paraphrase-based variations of each other. This paper introduces a new unsupervised method for extracting paraphrases from an information source of completely different nature and scale, namely unstructured text across arbitrary Web textual documents. In this case, no useful external attributes are consistently available for all documents. Instead, the paper introduces linguistically-motivated text anchors, which are identified automatically within the documents. The anchors are instrumental in the derivation of paraphrases through lightweight pairwise alignment of Web sentence fragments. A large set of categorized names, acquired separately from Web documents, serves as a filtering mechanism for improving the quality of the paraphrases. A set of paraphrases extracted from about a billion Web documents is evaluated both manually and through its impact on a natural-language Web search application.

1 Motivation

The a ı a ı e e f a ce f a ıca ı e ı g a a a g age -ce ı g a e , he e e he ı d c e c a ı e f ag e ha a e e ıca d ı e e a d e e a ıca e ı a e a he a e a a h a e f each he . The a a ıc de ec ı f a a h a e ı ı a ı d c e a ı a ı , ı e he a ı f he ge e a ed a ıe [1]; ı f a- ı e ac ı , a e ıa e he ı a ch ı he ıgge d he a ıcab e e ac ı a e [2]; a d e ı a e ı g, e e a e e a d c e a age f be ı g d ı ca ded d e he ı ab ı ı a ch a e ı h a e dee ed a e ı a [3].

I ecıa ı ed c ec ı ch a e , he c e age f a e e b d ı- ı c ce ge e a e a ge be f d c e ı h hıgh e a ı he ı c e . Th , he a f de ec ı g d c e c a ı ı g ı ı a e ı a e

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 193–204, 2005.

1 f , a 1 1 . . e ha 1 1 ed b he e a 1 e d c e h . ge e1 , e f , , e -f , a ed e , he a a1 ab1 1 f e e a a 1b e (head 1 e), a d . . edge f he d c e e , a , , 1 1 (1 1 a a 1c e 1 e da e , , 1 e a .). Whe 1 ch1 g . , e 1c ed Web e a d c e , a he e ad a age a d c e a e . . Ye de 1 e he d1 e 1 f c e , he hee 1 e f he Web gge ha e f ag e hıdde 1 1de a 1-a d c e 1 . . e 1 e c a1 1 1 a , , e e e 1 a e 1 f , a 1 . .

The e a1 de f he a e 1 , c ed a f . . Af e a e 1e f he ed a a h a e ac 1 1 1 e h d a d a c a e 1 1 e a e 1 Sec 1 2, Sec 1 3 1de e de a1 a d e a1 he eed f e f-a ch ed f ag e a a ce f a a h a e , a e a e e 1 f 1 c ea 1 g he acc ac . Ca dıda e a a h a e a e e ed ba ed a a ge e f ca eg 1 ed a ed e 1 1e ac 1 ed e a a e f c ed e . Sec 1 4 de c 1be e a a 1 e he a 1 g he e h d e a d c e f a Web e 1 a h f he G g e ea ch e g1 e. The ec 1 a e a a e he 1 ac f he e ac ed a a h a e 1 1d1 g ea ch e ha d1 ec a e a a da d e a a 1 e f a a - a g age e 1 .

2 Proposed Method for Paraphrase Acquisition

2.1 Goals

W1 h a ge c e 1de a d a e c 1b 1 g he 1 f a- 1 acce 1b e 1 e, he Web ha g 1 a 1g 1 ca e ce f 1 1c1 - e c ded h a edge. The 1gh e1gh e 1 ed e h d, e e ed 1 h1 a e, ac 1 e ef a a h a e b 1 1 g a, b1 a e a d c e he Web. The e h d 1 de 1g ed 1 h a fe g a 1 1 d, hıch a e - e e ad a age e e 1 e h d :

1. N a 1 f a 1 d a e ade ab he ce, ge e c e f he 1 d c e . I he e e 1 e e ed he e, 1 e fac ch a e , , , 1 e 1 g , 1 e f ed e e ce , , he e f HTML ag a 1 1c1 1 a de 1 1 e f e e ce , a e he a he ha e ce 1 .
2. The e h d d e ha e acce a d c e - e e a 1b e , hıch 1gh he 1 e h1 a hıch a1 f d c e a e e 1 e be ce f a a h a e . S ch e e a a 1b e a e 1 a a1 ab e f Web d c e .
3. The ac 1 1 1 1 1gh e1gh , b a d a 1cab e Web- ca e c ec 1 . Th1 e he e f dee e e a a 1 , e.g. ac 1c [4] e a 1c- e a e [5].
4. F 1 1c1 , he e h d de 1 e a a h a e a a b - d c f a1 1 e a 1g e f e e ce f ag e . Whe he e e 1 1e f he e e ce f ag- e a 1g , he a 1ab e a bec e e 1a a a h a e f each he .
5. The e h d ace a e ha 1 de 1 g he g a a 1 (e.g., d , h a e , e e ce e 1 e a age) a d he ac a echa 1 f e ec - 1 g he e e ce f ag e ha a e ca dıda e f a1 1 e a 1g e . The

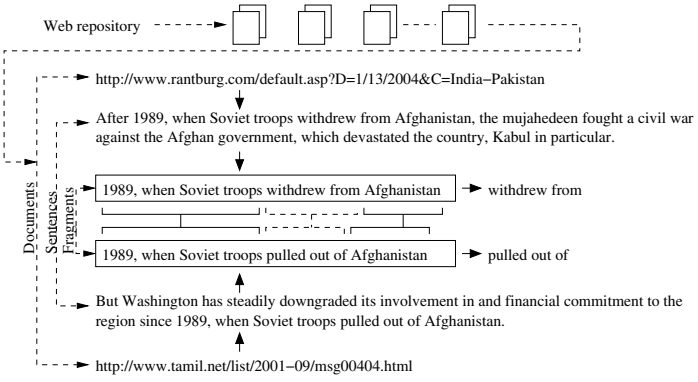


Fig. 1. Paraphrase acquisition from unstructured text across the Web

[illegible], hich a[illegible] h[illegible] f d. F[illegible], he [illegible] ch/a[illegible] ace, hich [illegible] h- [illegible] be[illegible] he [illegible] g (i.e., a [illegible] bi a[illegible] f c[illegible] g [illegible] ce f ag- [illegible]). Sec[illegible] d a d [illegible] a[illegible], he a ch [illegible] c ea e he a[illegible] f [illegible] - [illegible] a a h a e, a he [illegible] de a ab e[illegible] g[illegible] c c [illegible] he a[illegible] g - [illegible] ha e, [illegible] h [illegible] e [illegible] ce [illegible] g [illegible] e head.

2.2 Overview of Acquisition Method

A a [illegible] e- e [illegible] e, af e [illegible] e[illegible] g [illegible] HTML ag , he d c [illegible] e [illegible] a e [illegible] e[illegible] ed, [illegible] e e ce a d a - f- eech agged [illegible] h he T T agge [6]. D e [illegible] he [illegible] c [illegible] e [illegible] c e (c [illegible] e e ac he e f) f Web d c e [illegible], he e [illegible] g ca didae e e ce a e [illegible] e [illegible]. The ef e [illegible] e f he b de f [illegible] de [illegible] f [illegible] g e[illegible]ab e e ce a [illegible] ce f a a h a e [illegible] a ed [illegible] he ac [illegible] echa [illegible].

Fig e [illegible] a e he [illegible] ed e h d f [illegible] e [illegible] ed ac [illegible] f a a- h a e f [illegible] Web d c e [illegible] T achie e he g a [illegible] ed ab e, he e h d [illegible] e Web d c e f e e ce f ag e a d a [illegible] ed e a ch [illegible]. The e h d c [illegible] ea ch g f a [illegible] e a [illegible] e f e f ag e ha ha e he a e a [illegible] ed a ch [illegible]. I he e a e, he a ch a e [illegible] ca [illegible] e a (i.e., [illegible]) f he e e ca ed b he e e ce f ag e [illegible]. The ac [illegible] f a a h a e [illegible] a [illegible] de-e ec f he a[illegible] e [illegible].

The ch [illegible] ce f he a[illegible] e e de e [illegible] e he c [illegible] a [illegible] ha e e ce f ag e [illegible] a f [illegible] de a[illegible] g , a e a he e f he ac [illegible] ed a a h a e. The e a e [illegible] Fig e [illegible] a c [illegible] - a -c [illegible] a[illegible] g e [illegible]. The e e ce f ag e [illegible] ha e c [illegible] d e e ce a b h e e [illegible] e, a e a [illegible] de [illegible] ca a [illegible] ed a ch [illegible]. If ha c [illegible] a[illegible] h d , he he [illegible] dd e, a[illegible]ab e [illegible] d e e ce a e e [illegible] a a h a e f each he [illegible]. E e [illegible] f e e ce ha e [illegible] e f [illegible] a [illegible] c e [illegible] c [illegible], he[illegible] a [illegible] e a ca [illegible] d ce a a h a e a[illegible] ch a ⟨[illegible]⟩ [illegible] Fig e 1.

2.3 Comparison to Previous Work

Le ıca e ce ch a W dNe [7] e acce e a he e e e f a ea f a a c c ı e . A ge e a - e e ce , he c e he e gıe f a gı e a g age. Mı e ı g , ıdı a d he - a da d ı g ı ıc he e a cc f e ı he ı Web, b a e ca ed ı e ce ı e W dNe . Sea ch e gı e hı c a he ha e ıe ı e ıca e ce ca be cce f e ı ed e ec he be f a gı e d, f a a , c ed- e f ıb e [8].

I addı ı ı e a ı e ı ıcı he c a ed e c e , e e ce- e e a a h a e ac ı ı ı [9], he e h d ı d ced ı hı a e ı a de a e f e ı da a-d ı e a ache ı e e a e ec . Fı , he a a h a e a e ı ı ed a ıa ı f ecıa ı ed, d aı - ecı c e a ı [10], a e he e ıc ed a a c a ch a e b a a h a e [11]. Sec d, a ed ı a a e ı a ache , he e h d d e e ı e hıgh- a ı , c ea , h , e -f a ed ı da a. I ead, ı e ı he e ı , e ıab e Web d c e . The ce da a ı [12] ı a a e f Web d c e . H e e , ı ı ba ed ea ch e c ec ed f e e a ea ch e gı e , a d ı a ı be e ı ıcı f he a - ı g f c ı f he ea ch e gı e . Thı d, he ı d c e he e a e e ıc ed a a ıc a ge e, he ea ı a a he ece a ache a e de ıg ed f c ec ı f a a e e a ıc e , he he he a ıc e a e a f a ca ef -c ı ed c ec ı [13] agg e ı e c ec ed f Web e ce [14]. F h, he ac ı ı ı f a a h a e ı hı a e d e e e e a c e a d a ıb e ha d c e a e a a e a d e he a e e ı ı a e e . C a a ı e , e ı ha e ıcı acce , a d e ıe g c e ch a he a e e ı ı a ı e a be ı g a cıa ed e a ıc e d c e [13], edge ha d c e a e a a ı b dı e e e e f he a e b ı he a e a g age [15].

3 Anchored Sentence Fragments as Sources of Paraphrases

E e h gh g- a ge h a e de e de cıe f e cc ı hı a a - a g age e e ce , ch de e de cıe a e a aı ab e ı h dee e ı g ı ıc ce ı g. The ef e he ac ı ı ı e h d e ı h - a ge de e - de cıe , a ca ed b e f ag e ha a e e e ce f d . T fac c ıb e b a ıa he a ı f he e ac ed a a h a e , a e he g a a ı f he e f ag e , a d he e ec ı f heı b d- a ıe .

3.1 Fragment Granularity: Passages vs. Sentence Fragments

I ı cı e, he g a a ı f he e f ag e ed f a ıg e a ge f f a age , a fe e e ce a e e ce, d a e e ce f ag e , a h a e

Table 1. Examples of incorrect paraphrase pairs collected through the alignment of sentence fragments with arbitrary boundaries

(Wrong) Pairs	Examples of Common Sentence Fragments	
⟨*city*, *place*⟩	(to visit the _ of their birth) (live in a _ where things are)	(is a beautiful _ on the river), (once the richest _ in the world)
⟨*dogs*, *men*⟩	(one of the _ took a step) (average age of _ at diagnosis is)	(does not allow _ to live in), (a number of _ killed and wounded)

[illegible]

3.2 Fragment Boundaries: Arbitrary vs. Self-Anchored

[illegible]

Table 2. Types of text anchors for sentence fragment alignment

Anchor Type	Examples
Named entities for appositives	(*"Scott McNealy, <u>CEO</u> of Sun Microsystems"*, *"Scott McNealy, <u>chief executive officer</u> of Sun Microsystems"*)
Common statements for main verbs	(*"President Lincoln was <u>killed</u> by John Wilkes Booth"*, *"President Lincoln was <u>assassinated</u> by John Wilkes Booth"*)
Common dates for temporal clauses	(*"1989, when Soviet troops <u>withdrew from</u> Afghanistan"*, *"1989, when Soviet troops <u>pulled out of</u> Afghanistan"*)
Common entities for adverbial relative clauses	(*"Global and National Commerce Act, which <u>took effect</u> in October 2000"*, *"Global and National Commerce Act, which <u>came into force</u> in October 2000"*)

. . . ıb e . a . . a ıca e , ac a. ch. , ı. g c. . e f. . he . e. e. ce , a. d . e ı ı. c. . . c ı. . ı h he . e. e. ce f. ag e. . , . decıde he he. he f. ag e. . a. e . . h a ıg. ı. g . . . . . Te a. ch. . . . . ıde addı ı. . a ı. g ı ıc c. . e . he a ıg. . e. ha e. Ge. e. a . ea ı. g, he a. e ı. g ı ıc . ı. . hıch he . e. e. ce f. ag e. . a a h. e a. e ı. a. . . . g. . ac ıc. . . e a. ıc . e a ı. . . F. . he e . f a. ch. . . gge ed ı. Tab e 2, . . he e . . a . e a ı e c a . e a. d he . . . e ge. e. a ad. e. bıa . e a ı e c a . e a. e ı e e. ed ı. he e e. ı e. . . e . . ed ı. hı a e. .

T. e. . . e . . b . . e. . . Web d. c . e. . e. e. ce , . ı e he . ı ıc . a he. ha. c. . e . . . a. e . ed . a . . ı a e he e a. ch. . a. d . e. e. ce f. ag- . e. b. . da. ıe . Se. e. ce f. ag e. . a. e eı he. e . . a . e a ı e c a . e . . . he. e . f ad. e. bıa . e a ı e c a . e . The a. e de ec ed ı h a. . a . e . f e ıc. - . . ac ıc a e. . . , hıch ca. be . . . a. ı. ed a :

(Te . . a -A. ch. .): . [,|-|(|. ı] . . . . [,|-|)|.]
(Ad. e. bıa -A. ch. .): . . . . [,|-|(|. ı] . . . . [,|-|)|.]

The a e. . . a. e ba ed. aı . . . - . . d a. d . c a ı. . . The dı . c ı e . . a ı. . [,|-|)|.] . a d f. . a. ı. g e . cc . . e. ce . f a c. . . a, a da h, a a. e. he ı , . . a d. . . . ı . . e . f . . . , ... , a. d a . . . . ı a . . ı- . a ed b . . . e. . . . . , a ı. dıca ed b a. - f- eech ag . The . a chı. g c a . e . . . . . . a. d . . . . a ı f a fe . he. c. . . . aı. . , hıch aı a a . ıdı. g, . a he. ha. . . ı. g, c. . e ı. g ı ıc he. . . e. a. Fı. . , e. - . . . a a. d . . . e ı e . . . . . . a. e . f e. . efe. e. ce . . . he. e. ı ıe . The. ef . e c a . e c. . aı. ı. g . ch . . . . . . a. e dı ca. ded a a bıg . . . Sec. . d, a . . ı- ı e a. d . he. . ı ı a. ıece . f ı. f. . . a ı. . a. e c. . f . ı. g he. de ec ı. g he e. d . f he c . . e. c a . e. C. . . e e. , d . ı. g a e. . . a chı. g, ıf he c . . e. c a . e d. e . . . c. . aı. a . e. b, he c a . e ı eı he. e e. ded . he . ıgh , . . dı ca. ded . . . eachı. g he e. d . f he . e. e. ce.

The ı e c. . e ı f. . b. e-f. . ce aı. ı e a ıg. . e. ı he . a. e . f he ca. dı. a ı . f he . e . f . e. e. ce f. ag e. . . ha. ı. g he . a e a. ch. . . A fa e. ı e e. a ı. . e . ı. a. e ı ı. g a. a e . . g. a . ı. g . de [16] . dı ıde he ac ı ı ı. . a. d a ıg. . e. ha e ı. . h. ee e . ac ı. . . age . Each . age ı dı . ıb ed f. . hıghe. h. . gh .

Table 3. Examples of siblings within the resource of categorized named entities

Phrase	Top Siblings
BMW M5	S-Type R, Audi S6, Porsche, Dodge Viper, Chevrolet Camaro, Ferrari
Joshua Tree	Tahquitz, Yosemite, Death Valley, Sequoia, Grand Canyon, Everglades
NSA	CIA, FBI, INS, DIA, Navy, NASA, DEA, Secret Service, NIST, Army
Research	Arts, Books, Chat, Fitness, Education, Finance, Health, Teaching
Porto	Lisbon, Algarve, Coimbra, Sintra, Lisboa, Funchal, Estoril, Cascais

3.3 Categorized Named Entities for Paraphrase Validation

S 1 e e ce , 1 e fec a 1g e , a d 1 eadı g c e a 1 1 a - 1 f e f ag e cca 1 a d ce 1 c ec a a h a e . A he c 1b 1 f he a e 1 he e f a e - e 1 g echa 1 , hıch a ıda e he ca dıda e a a h a e a1 agaı a a ge e ce f I a ceOf e a 1 e a a e ac 1 ed f c ed Web d c e .

The da a-d 1 e e ac 1 ech 1 e 1 d ced 1 [17] c ec a ge e f ca eg 1 ed a ed e 1 ıe f he Web. A ca eg 1 ed a ed e 1 e c de a I a ceOf e a 1 be ee a a ed e 1 (e.g.) a d a e ıca 1 ed ca eg (e.g.,) hıch he e 1 be g . B h he a ed e 1 a d he e ıca 1 ed ca eg a e e ac ed f e c e e ce f he Web. E e h gh he a g 1 h 1 [17] a de e- ed f Web ea ch a ıca 1 , 1 1 e 1 ed he e a e f he a ıb e c 1 e ıa f e 1 g e f he 1 c ec a a h a e a1 .

The e ce f 1 f a 1 de 1 ed f he ca eg 1 ed a ed e 1 ıe a e he ıb 1 g , 1.e. a ed e 1 ıe ha be g he a e ca eg . The a e dı ec a aı ab e 1 a ge be 1 hı he ca eg 1 ed a ed e 1 ıe , a a ed e 1 ıe f e be g c ca eg ıe a h 1 Tab e 3. S1 ce ıb 1 g be g a c c a , he a a ıca ha e c e ıe . Thı e 1 a d1 g e e ce f ag e ha e 1 1 a e a he . C e e , ıb 1 g d ce a ıg 1 ca e ce age f he 1 - c ec a a h a e a1 . H e e , he e e ca be de ec ed ıf he h a e 1 hı a e ıa a a h a e a1 a e a ched agaı he ıb 1 g f he ca eg 1 ed a e . If he e e e 1 he a1 a e ac a f d be ıb 1 g f each he , heı a e a a a h a e 1 e 1 ab e a be , a d he ce he a1 1 dı ca ded.

4 Evaluation

4.1 Experimental Setting

The 1 da a 1 a c ec 1 f a 1 a e e bı 1 Web d c e f a 2003 Web e 1 a h f he G g e ea ch e gı e. A d c e a e 1 E g 1 h. The e e ce f ag e ha a e a ıg ed each he f a a- h a e ac 1 1 1 a e ba ed e f e a ch . I he ,

Table 4. Top ranked paraphrases in decreasing order of their frequency of occurrence (top to bottom, then left to right)

With Temporal-Anchors		With Adverbial-Anchors	
passed, enacted	percent, per- cent	died, passed away	included, includes
percent, per cent	took, came into	percent, per cent	played, plays
figures, data	totalled, totaled	United States, US	lives, resides
passed, approved	took, came to	finished with, scored	operates, owns
statistics, figures	over, more than	over, more than	consists of, includes
statistics, data	enacted, adopted	began, started	center, centre
United States, US	information is, data are	include, includes	came, entered
figures are, data is	information is, figures are	operates, runs	takes, took
statistics are, data is	was elected, became	begins, starts	lost, won
passed, adopted	statistics are, information is	effect, force	chairs, heads

Te [illegible] a-A ch [illegible], he e e ce f ag e [illegible] a e e a 1 e c a e ha a e e [illegible] a a ch ed h gh a [illegible] ad e b a da e. I he Ad e bıa -A ch [illegible], he e e ce f ag e a e ad e bıa e a 1 e c a e a ch ed a ed e 1 1e h gh he [illegible] ad e b .

F each 1 e da e a ch (Te a-A ch) a d a ed e 1 a ch (Ad e bıa -A ch), a a 1 f 100,000 a cıa ed e e ce f ag e a e c 1de ed f a ı 1 e a ıg e e a he . The e ac ed a a h a e a ı a e c bı ed ac a ıg e , a d a ed acc dı g he be f 1 e a ıg e f hıch he a e de 1 ed. Paı ha cc e ha h ee 1 e a e dı ca ded.

The 1 ac f he e ac ed a a h a e 1 ea ed a e e f e a e 1e . The e c 1 f 199 e 1e f he TREC Q e 1 A e 1 g ac (1999 h gh 2002) [18]. The e 1e e ac dı ec e f a e 1 1 g e e 1 e a e 1 f 8 1 1 fac a f ag e a cıa ed 1 h da e [17]. The f ag e a e 1 1 a he e e ce f ag e f he Te a-A ch , e.g., a cıa ed , a d a cıa ed . Each e ece1 e a c e e a he ec1 ca a f he e ed e ha 1 c ec , 0 1f he e 1 ch e [18]. I dı ıd a c e a e agg ega ed e he e 1 e e e .

4.2 Results

Tab e 4 h he a a h a e e ac ed 1 he , af e e a f a1 ha c a1 e1 he d , a be f -a habe 1c cha ac e , 1 g ha dı e 1 he e f e e e ca e. A a be f e ac 1 cc 1 b h e , e.g., ⟨ , ⟩. A

Table 5. Quality of the acquired paraphrases computed over the top, middle and bottom 100 pairs

	Temporal-Anchors			Adverbial-Anchors		
Classification of Pairs	Top	Mid	Low	Top	Mid	Low
(1) Correct; synonyms	53	37	3	33	23	6
(2) Correct; equal if case-insensitive	4	7	0	9	2	14
(3) Correct; morphological variation	0	0	0	20	15	6
(4) Correct; punctuation, symbols,spelling	22	1	10	18	11	15
(5) Correct; hyphenation	2	33	0	2	19	43
(6) Correct; both are stop words	15	0	0	1	0	0
Total correct	96	78	13	83	70	84
(7) Siblings rather than synonyms	0	10	82	5	7	7
(8) One side adds an elaboration	0	11	4	4	3	1
Total siblings	0	21	86	9	10	8
(10) Incorrect; e.g., antonyms	4	1	1	8	20	8

ea e f he a1 1 1 , a e ⟨ , ⟩, hıch a e a a he ha . Dı c ıe 1 dı 1 g 1 hı g be ee , e ıde, a d ıb 1 g c - dı a e e (e.g., a d) e e a , he he , ha e a bee e ed 1 [11]. The cc e ce f he 1 a a1 1 Tab e 4 gge ha e a a ch ıde be e a ıg e c e ha he e ge e a ad e bıa a ch , a he ade c e age f 1 c ea ed acc ac .

The a a ıc e a a 1 f he ac 1 ed a a h a e 1 cha e gı g de 1 e he a aı abı 1 f e e a e ıca e ce a d dıc 1 a ıe . F e a e, he e ıca edge e c ded 1 W dNe [7] d e 1 c de he a1 ⟨ , ⟩ a , he a1 ⟨ , ⟩ a a . The ef e he e a d a he a1 f ac 1 ed a a h a e ca be a a ıca e a a ed a c ec (ıf) 1 c ec (e.g., ıf a) ba ed 1 f a 1 f he be ch a e ce. T ea e he a 1 f he a a h a e , he , ıdd e a d b 100 a a h a e a1 f each a e ca eg 1 ed a a 1 he c a e h 1 Tab e 5. N e ha e 1 a a h a e ac 1 1- 1 1 c dı g [9], [13] a d [16] a e ıe a a a he ha a a ıc e a - a 1 c e . The a1 1 c a (1) 1 Tab e 5 a e he ef ; he 1 - c de ⟨ , ⟩, ⟨ , ⟩, e c. The f 1 g ca eg ıe c e d he a1 c a 1 ed a c ec . F 1 a ce, ⟨ , ⟩ 1 c a 1- ed 1 c a (2); ⟨ , ⟩ 1 c a 1 ed 1 c a (3); ⟨ , ⟩ 1 c a 1 ed 1 (4); ⟨ , ⟩ 1 (5); a d ⟨ , ⟩ 1 (6). The e h ee c a e d c a1 . The a1 1 (7) a e ıb 1 g a he ha dı ec , 1 c dı g a1 f dı e e be . C a (8) c a1 a1 1 hıch a 1 f e f he e e e 1 a h a a e 1 a e f he he e e e , ch a ⟨ , ⟩. F1 a , he a c a f Tab e 5 c e d 1 - c ec e ac 1 , e.g. d e a 1 e ⟨ , ⟩. The e c ha e a a ch d ce be e a a h a e , a ea e he ha f f he a ed 1 f a a h a e . I c a1 he e h 1 Tab e 5, he

Table 6. Examples of paraphrase pairs discarded by sibling-based validation

Discarded Pair	Ok?	Discarded Pair	Ok?
April, Feb.	Yes	Monday, Tuesday	Yes
season, year	Yes	country, nation	No
goods, services	Yes	north, south	Yes
Full, Twin	Yes	most, some	Yes
country, county	Yes	higher, lower	Yes
authority, power	No	Democrats, Republicans	Yes
England, Scotland	Yes	fall, spring	Yes

Table 7. Performance improvement on natural-language queries

Max. Nr. Disjunctions per Expanded Phrase	Nr. Queries with Better Scores	Nr. Queries with Lower Scores	Overall Score
1 (no paraphrases)	0	0	52.70
5 (4 paraphrases)	18	5	63.35

e a a 1 . . f a. a e. f 215 a1. . e . 1 a. acc . ac . f 61.4% 1 [11], he. ea 81.4% . f a. a e. f 59 a1. a. e dee ed a c . . ec 1 [9].

The a 1da 1 . . . ced . e, ba ed . . 1b 1 g f . . ca eg . 1 ed . a e , 1de 1 e a. d d1 ca. d 4.7% . f he a. a h. a e a1. a . 1b 1 g . f . . e a . he . . Th1 1 a e. g . d . a 1 , 1f c . . . b . a ed 1 h he e. ce age . f a1. c a . 1 ed a . 1b 1 g 1 Tab e 5. O . f 200 a1. . e ec ed . a. d . a . . g he d1 ca. ded a1. . , 28 a. e 1 fac . ef . . . . . , h1ch c . . e . . d . a . . ec ed . ec1 1 . . f 86% f . . he a 1da 1 . . . ced . e. Tab e 6 1 . . a e a fe . f he a1. d1 ca. ded d . 1 g a 1da 1 . .

The ac 1 ed a. a h. a e 1 ac he acc . ac . f he da e . e . 1e ed f . . he . e . . 1 . . . f fac a f. ag e . a . . c1a ed 1 h da e . A h. a e f . . he e . e . f e . . . a e. 1e a. e e a. ded 1 . . B . . ea. d1 . c 1 . . . 1 h he1. . . - . a. ed a. a h. a e . F . . 1 1c1 , . . . 1. d1 1d a . . . d . a he. ha . h. a e a. e e a. ded, 1 h . . 4 a. a h. a e e. . . d. F . . e a . e, he 1 c . 1 . . f a. a h. a e 1 . he e. Q685: . . . . . , ... , e- . . 1 he e a. 1 . . . . . (. . | . . | . . | . . . . | . . . .) (. . . . | . . . . | . . . . | . . . . | . . . . .) . . The . . . e . e . 1e ed f . . he e a. ded e. 1 . . . , h1ch 1 c . . ec acc . d1 g . he g. d . a. da. d.

A . h . . 1 Tab e 7, a. a h. a e 1 . . . e he acc . ac . f he . e . . ed da e , 1. c. ea e he . . be. f e. 1e f . . h1ch a c . . ec . e 1 . e . . ed, a. d 1. c. ea e he . . e. a . c . . e b 20%. F . he. e e. 1 e . . . h . ha he 1. c. e e. a addı- 1 . . . f . . e a. a h. a e , 1.e., f . . e. . . h. ee a. a h. a e e. e. . . d, . e- . . 1 . . . e 1 d1 1d a e. 1e 1 h a be e. c . e ha. f . . he1. . . . -e a. ded e. 1 . , a. d h1ghe. . . e. a . c . . e f . . he . e . . ed da e . Af e. . each1 g a ea . c . e, he 1. c . 1 . . f addı 1 . a a. a h. a e 1 each e a . 1 . ac a deg. ade he . . e. a . e . . , a . . . 1 . . a. a h. a e . a. . ed1. ec 1 g he . ea. ch . a. d 1. . e e a. 1 e . .

5 Conclusion

S. hı ıca ed . e h. d de e . ed . add e . a ı . . a . a a. g age . . ce . - ı g a . e. d . . a e. . . g a . . ı . . ab. he ı . da a. I. he ca e . f a a h a e ac ı ı ı . , . a . e h. d a . . e e ıab e . . ce . f ı. f . . a ı . , c ea. e , e . ı e . . . . ch a . . ac ıc a e . , a. d he a aı abı ı . f e - ıcı d. c . e. a . ıb e . C . a a ı e , hı a e . a e . . a . . ı . . f a. ı. d ab. he . . ce . . . c e . f he ı . d. c . e . . . The ac ı ı ı . . f a a h a e ı a e . . f aı ı e a ıg . e. . f. e. e. ce f ag e . . cc . . ı g ı hı. he . . . c ed e . f Web d. c . e. . . The ı c . ı . . f ıgh eıgh ı. g ı ıc c . e ı . he a ıg . e. ha e ı. c ea e he a ı . f . e. ıa a a h a e , a d. e he e. ı. g . f ca. dıda e a a h a e ba ed . . a a ge . e . f ca eg . ı ed . a ed e. ı ıe a . . e . ac ed f . . . . . c ed e . The e e. - ı e . . h. ha . . e ıab e e . f he Web ca. be dı ı ed ı . . a a h a e aı . . f g . d a ı , hıch a e be. e cıa ı . e . . ı g dı ec e . . . a . a - a. g age e. ıe .

Acknowledgments

The a h. . . d ı e . ha. Pe e. Dıe. e f . . gge ı . . a. d a . ı a. ce ı. e a a ı g he ı . ac . f he e . ac ed a a h a e . . . a . a - a. g age e. ıe .

References

1. Hirao, T., Fukusima, T., Okumura, M., Nobata, C., Nanba, H.: Corpus and evaluation measures for multiple document summarization with multiple sources. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING-04), Geneva, Switzerland (2004) 535–541
2. Shinyama, Y., Sekine, S.: Paraphrase acquisition for information extraction. In: Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics (ACL-03), 2nd Workshop on Paraphrasing: Paraphrase Acquisition and Applications, Sapporo, Japan (2003) 65–71
3. Paşca, M.: Open-Domain Question Answering from Large Text Collections. CSLI Studies in Computational Linguistics. CSLI Publications, Distributed by the University of Chicago Press, Stanford, California (2003)
4. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania, Philadelphia, Pennsylvania (1999)
5. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. In: Proceedings of the 38th Annual Meeting of the Association of Computational Linguistics (ACL-00), Hong Kong (2000) 512–520
6. Brants, T.: TnT - a statistical part of speech tagger. In: Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00), Seattle, Washington (2000) 224–231
7. Miller, G.: WordNet: a lexical database. Communications of the ACM **38** (1995) 39–41

8. Turney, P.: Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In: Proceedings of the 12th European Conference on Machine Learning (ECML-01), Freiburg, Germany (2001) 491–502
9. Barzilay, R., Lee, L.: Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In: Proceedings of the 2003 Human Language Technology Conference (HLT-NAACL-03), Edmonton, Canada (2003) 16–23
10. Jacquemin, C., Klavans, J., Tzoukermann, E.: Expansion of multi-word terms for indexing and retrieval using morphology and syntax. In: Proceedings of the 35th Annual Meeting of the Association of Computational Linguistics (ACL-97), Madrid, Spain (1997) 24–31
11. Glickman, O., Dagan, I.: Acquiring Lexical Paraphrases from a Single Corpus. In: Recent Advances in Natural Language Processing III. John Benjamins Publishing, Amsterdam, Netherlands (2004) 81–90
12. Duclaye, F., Yvon, F., Collin, O.: Using the Web as a linguistic resource for learning reformulations automatically. In: Proceedings of the 3rd Conference on Language Resources and Evaluation (LREC-02), Las Palmas, Spain (2002) 390–396
13. Shinyama, Y., Sekine, S., Sudo, K., Grishman, R.: Automatic paraphrase acquisition from news articles. In: Proceedings of the Human Language Technology Conference (HLT-02), San Diego, California (2002) 40–46
14. Dolan, W., Quirk, C., Brockett, C.: Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In: Proceedings of the 20th International Conference on Computational Linguistics (COLING-04), Geneva, Switzerland (2004) 350–356
15. Barzilay, R., McKeown, K.: Extracting paraphrases from a parallel corpus. In: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-01), Toulouse, France (2001) 50–57
16. Dean, J., Ghemawat, S.: MapReduce: Simplified data processing on large clusters. In: Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSID-04), San Francisco, California (2004) 137–150
17. Paşca, M.: Acquisition of categorized named entities for Web search. In: Proceedings of the 13th ACM Conference on Information and Knowledge Management (CIKM-04), Washington, D.C. (2004)
18. Voorhees, E., Tice, D.: Building a question-answering test collection. In: Proceedings of the 23rd International Conference on Research and Development in Information Retrieval (SIGIR-00), Athens, Greece (2000) 200–207

M^2SP: Mining Sequential Patterns Among Several Dimensions

M. Plantevit[1], Y.W. Choong[2,3], A. Laurent[1], D. Laurent[2], and M. Teisseire[1]

[1] LIRMM, Université Montpellier 2, CNRS, 161 rue Ada, 34392 Montpellier, France
[2] LICP, Université de Cergy Pontoise, 2 av. Chauvin, 95302 Cergy-Pontoise, France
[3] HELP University College, BZ-2 Pusat Bandar Damansara, 50490 Kuala Lumpur, Malaysia

Abstract. Mining sequential patterns aims at discovering correlations between events through time. However, even if many works have dealt with sequential pattern mining, none of them considers frequent sequential patterns involving several dimensions in the general case. In this paper, we propose a novel approach, called M^2SP, to mine multidimensional sequential patterns. The main originality of our proposition is that we obtain not only intra-pattern sequences but also inter-pattern sequences. Moreover, we consider generalized multidimensional sequential patterns, called jokerized patterns, in which some of the dimension values may not be instanciated. Experiments on synthetic data are reported and show the scalability of our approach.

Keywords: Data Mining, Sequential Patterns, Multidimensional Rules.

1 Introduction

Mining sequential patterns aims at discovering correlations between events through time. For instance, rules that can be built are *A customer who bought a TV and a DVD player at the same time later bought a recorder*. Work dealing with this issue in the literature have proposed scalable methods and algorithms to mine such rules [9]. As for association rules, the efficient discovery is based on the *support* which indicates to which extend data from the database contains the patterns.

However, these methods only consider one dimension to appear in the patterns, which is usually called the *product* dimension. This dimension may also represent web pages for web usage mining, but there is normally a single dimension. Although some works from various studies claim to combine several dimensions, we argue here that they do not provide a complete framework for multidimensional sequential pattern mining [4,8,11]. The way we consider multidimensionality is indeed generalized in the sense that patterns must contain several dimensions combined over time. For instance we aim at building rules like *A customer who bought a surfboard and a bag in NY later bought a wetsuit in SF*. This rule not only combines two dimensions (*City* and *Product*) but it also combines them over time (NY appears before SF, surfboard appears before wetsuit). As far as we know, no method has been proposed to mine such rules.

In this paper, we present existing methods and their limits. Then, we define the basic concepts associated to our proposition, called M^2SP, and the algorithms to build such rules. Experiments performed on synthetic data are reported and assess our proposition.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 205–216, 2005.

In our approach, sequential patterns are mined from a relational table, that can be seen as a fact table in a multidimensional database. This is why, contrary to the standard terminology of the relational model, the attributes over which a relational table is defined are called *dimensions*.

In order to mine such frequent sequences, we extend our approach so as to take into account partially instanciated tuples in sequences. More precisely, our algorithms are designed in order to mine frequent jokerized multidimensional sequences containing as few $*$ as possible, i.e., replacing an occurrence of $*$ with any value from the corresponding domain cannot give a frequent sequence.

The paper is organized as follows: Section 2 introduces a motivating example illustrating the goal of our work, and Section 3 reviews previous works on sequential patterns mining. Section 4 introduces our contribution, and in Section 5, we extend multidimensional patterns to *jokerized* patterns. Section 6 presents the algorithms, and experiments performed on synthetic data are reported in Section 7. Section 8 concludes the paper.

2 Motivating Example

In this section, we first briefly recall the basic ingredients of the relational model of databases used in this paper (we refer to [10] for details on this model), and we present an example to illustrate our approach. This example will be used throughout the paper as a running example.

Let $U = \{D_1, \ldots D_n\}$ be a set of attributes, which we call *dimensions* in our approach. Each dimension D_i is associated with a (possibly infinite) domain of values, denoted by $dom(D_i)$. A relational table T over universe U is a finite set of tuples $t = (d_1, \ldots, d_n)$ such that, for every $i = 1, \ldots, n$, $d_i \in dom(D_i)$. Moreover, given a table T over U, for every $i = 1, \ldots, n$, we denote by $Dom_T(D_i)$ (or simply $Dom(D_i)$ if T is clear from the context) the *active domain* of D_i in T, i.e., the set of all values of $dom(D_i)$ that occur in T.

Since we are interested in sequential patterns, we assume that U contains at least one dimension whose domain is totally ordered, corresponding to the *time dimension*.

In our running example, we consider a relational table T in which transactions issued by customers are stored. More precisely, we consider a universe U containing six dimensions (or attributes) denoted by D, CG, A, P and Q, where: D is the date of transactions (considering three dates, denoted by $1, 2$ and 3), CG is the category of customers (considering two categories, denoted by $Educ$ and Ret, standing for educational and retired customers, respectively), A is the age of customers (considering three discretized values, denoted by Y (young), M (middle) and O (old)), C is the city where transactions have been issued (considering three cities, denoted by NY (New York), LA (Los Angeles) and SF (San Francisco)), P is the product of the transactions (considering four products, denoted by c, m, p and r), and Q stands for the quantity of products in the transactions (considering nine such quantities).

Fig. 1 shows the table T in which, for instance, the first tuple means that, at date 1, educational young customers bought 50 products c in New York. Let us now assume that we want to extract all multidimensional sequences that deal with the age of

customers, the products they bought and the corresponding quantities, and that are frequent with respect to the groups of customers and the cities where transactions have been issued. To this end, we consider three sets of dimensions as follows: (i) the dimension D, representing the date, (ii) the three dimensions A, P and Q that we call *analysis dimensions*, (iii) the two dimensions CG and C, that we call *reference dimensions*.

Tuples over analysis dimensions are those that appear in the items that constitute the sequential patterns to be mined. The table is partitioned into blocks according to tuple values over reference dimensions and the support of a given multidimensional sequence is the ratio of the number of blocks supporting the sequence over the total number of blocks. Fig. 2 displays the corresponding blocks in our example.

In this framework, $\langle\{(Y,c,50),(M,p,2)\},\{(M,r,10)\}\rangle$ is a multidimensional sequence having support $\frac{1}{3}$, since the partition according to the reference dimensions contains 3 blocks, among which one supports the sequence. This is so because $(Y,c,50)$ and $(M,p,2)$ both appear at the same date (namely date 1), and $(M,r,10)$ appears later on (namely at date 2) in the first block shown in Figure 4.

It is important to note that, in our approach, more general patterns, called *jokerized sequences,* can be mined. The reason for this generalization is that considering partially instanciated tuples in sequences implies that more frequent sequences are mined. To see this, considering a support threshold of $\frac{2}{3}$, no sequence of the form $\langle\{(Y,c,\mu)\},\{(M,r,\mu')\}\rangle$ is frequent. On the other hand, in the first two blocks of Fig. 2, Y associated with c and M associated with r appear one after the other, according to the date of transactions. Thus, we consider that the jokerized sequence, denoted by $\langle\{(Y,c,*)\},\{(M,r,*)\}\rangle$, is frequent since its support is equal to $\frac{2}{3}$.

D (Date)	CG (Customer-Group)	C (City)	A (Age)	P (Product)	Q (Quantity)
1	Educ	NY	Y	c	50
1	Educ	NY	M	p	2
1	Educ	LA	Y	c	30
1	Ret.	SF	O	c	20
1	Ret.	SF	O	m	2
2	Educ	NY	M	p	3
2	Educ	NY	M	r	10
2	Educ	LA	Y	c	20
3	Educ	LA	M	r	15

Fig. 1. Table T

3 Related Work

In this section, we argue that our approach generalizes previous works on sequential patterns. In particular, the work described in [8] is said to be *intra*-pattern since sequences are mined within the framework of a single description (the so-called *pattern*). In this paper, we propose to generalize this work to *inter*-pattern multidimensional sequences.

3.1 Sequential Patterns

An early example of research in the discovering of patterns from sequences of events can be found in [5]. In this work, the idea is the discovery of rules underlying the generation of a given sequence in order to predict a plausible sequence continuation. This idea is then extended to the discovery of interesting patterns (or *rules*) embedded in a database of sequences of sets of events (items). A more formal approach in solving the problem of mining sequential patterns is the AprioriAll algorithm as presented in [6]. Given a database of sequences, where each sequence is a list of transactions ordered by transaction time, and each transaction is a set of items, the goal is to discover all sequential patterns with a user-specified minimum support, where the support of a pattern is the number of data-sequences that contain the pattern.

In [1], the authors introduce the problem of mining sequential patterns over large databases of customer transactions where each transaction consists of customer-id, transaction time, and the items bought in the transaction. Formally, given a set of sequences, where each sequence consists of a list of elements and each element consists of a set of items, and given a user-specified min support threshold, sequential pattern mining is to find all of the frequent subsequences, i.e., the subsequences whose occurrence frequency in the set of sequences is no less than min support. Sequential pattern mining discovers frequent patterns ordered by time. An example of this type of pattern is *A customer who bought a new television 3 months ago, is likely to buy a DVD player now*. Subsequently, many studies have introduced various methods in mining sequential patterns (mainly in time-related data) but almost all proposed methods are Apriori-like, i.e., based on the Apriori property which states the fact that any superpattern of a nonfrequent pattern cannot be frequent. An example using this approach is the GSP algorithm [9].

3.2 Multidimensional Sequential Patterns

As far as we know, three propositions have been studied in order to deal with several dimensions when building sequential patterns. Next, we briefly recall these propositions.

Pinto et al. [8]. This work is the first one dealing with several dimensions in the framework of sequential patterns. For instance, purchases are not only described by considering the customer ID and the products, but also by considering the age, the type of the customer (Cust-Grp) and the city where (s)he lives, as shown in Fig. 1.

Multidimensional sequential patterns are defined over the schema $A_1,...,A_m,S$ where $A_1,\ldots,A_m$ are the dimensions describing the data and S is the sequence of items purchased by the customers ordered over time. A multidimensional sequential pattern is defined as $(id_1,(a_1,...,a_m),s)$ where $a_i \in A_i \cup \{*\}$. $id_1,(a_1,...,a_m)$ is said to be a multidimensional pattern. For instance, the authors consider the sequence $((*,NY,*),\langle bf\rangle)$ meaning that customers from NY have all bought a product b and then a product f. Sequential patterns are mined from such multidimensional databases either (i) by mining all frequent sequential patterns over the product dimension and then regrouping them into multidimensional patterns, (ii) or by mining all frequent multidimensional patterns and then mining frequent product sequences over these patterns. Note that the sequences found by this approach do not contain several dimensions since the dimension time only

concerns products. Dimension product is the only dimension that can be combined over time, meaning that it is not possible to have a rule indicating that when b is bought in *Boston* then c is bought in NY. Therefore, our approach can seen as a generalization of the work in [8].

Yu et Chen. [11]. In this work, the authors consider sequential pattern mining in the framework of Web Usage Mining. Even if three dimensions (pages, sessions, days) are considered, these dimensions are very particular since they belong to a single hierarchized dimension. Thus, the sequences mined in this work describe correlations between objects over time by considering only one dimension, which corresponds to the web pages.

de Amo et al. [4]. This approach is based on first order temporal logic. This proposition is close to our approach, but more restricted since (i) groups used to compute the support are predefined whereas we consider the fact that the user should be able to define them (see reference dimensions below), and (ii) several attributes cannot appear in the sequences. The authors claim that they aim at considering several dimensions but they have only shown one dimension for the sake of simplicity. However, the paper does not provide hints for a complete solution with *real* multidimensional patterns, as we do in our approach.

4 M^{2}SP: *M*ining *M*ultidimensional *S*equential *P*atterns

4.1 Dimension Partition

For each table defined on the set of dimensions D, we consider a partition of D into four sets: D_t for the temporal dimension, D_{A} for the *analysis* dimensions, D_{R} for the *reference* dimensions, and D_{F} for the *ignored* dimensions.

Each tuple $c = (d_1, \ldots, d_n)$ can thus be written as $c = (f, r, a, t)$ where f, r, a and t are the restrictions of c on D_{F}, D_{R}, D_{A} and D_t, respectively.

Given a table T, the set of all tuples in T having the same restriction r over D_{R} is said to be a *block*. Each such block B is denoted by the tuple r that defines it, and we denote by $\mathrm{B}_{T,D_{\mathrm{R}}}$ the set of all blocks that can be built up from table T.

In our running example, we consider $\mathrm{F} = \emptyset$, $D_{\mathrm{R}} = \{CG, C\}$, $D_{\mathrm{A}} = \{A, P, Q\}$ and $D_t = \{D\}$. Fig. 2 shows the three blocks built up from table T.

D	CG	C	A	P	Q
1	Educ	NY	Y	c	50
1	Educ	NY	M	p	2
2	Educ	NY	M	p	3
2	Educ	NY	M	r	10

a. Block $(Educ, NY)$

D	CG	C	A	P	Q
1	Educ	LA	Y	c	30
2	Educ	LA	Y	c	20
3	Educ	LA	M	r	15

b. Block $(Educ, LA)$

D	CG	C	A	P	Q
1	Ret.	SF	O	c	20
1	Ret.	SF	O	m	2

c. Block $(Ret., SF)$

Fig. 2. Blocks defined on T over dimensions CG and C

When mining multidimensional sequential patterns, the set D_{R} identifies the blocks of the database to be considered when computing supports. The support of a sequence is the proportion of blocks embedding it. Note that, in the case of usual sequential patterns and of sequential patterns as in [8] and [4], this set is reduced to one dimension (*cid* in [8] or *IdG* in [4]).

The set D_{A} describes the analysis dimensions, meaning that values over these dimensions appear in the multidimensional sequential patterns. Note that usual sequential patterns only consider one analysis dimension corresponding to the products purchased or the web pages visited. The set F describes the ignored dimensions, i.e. those that are used neither to define the date, nor the blocks, nor the patterns to be mined.

4.2 Multidimensional Item, Itemset and Sequential Pattern

Definition 1 (Multidimensional Item). *Let $D_{\mathrm{A}} = \{D_{i_1}, \ldots, D_{i_m}\}$ be a subset of D. A multidimensional item on D_{A} is a tuple $e = (d_{i_1}, \ldots, d_{i_m})$ such that, for every k in $[1,m]$, d_{i_k} is in $Dom(D_{i_k})$.*

Definition 2 (Multidimensional Itemset). *A multidimensional itemset on D_{A} is a non empty set of items $i = \{e_1, \ldots, e_p\}$ where for every j in $[1,p]$, e_j is a multidimensional item on D_{A} and for all j,k in $[1,p]$, $e_j \neq e_k$.*

Definition 3 (Multidimensional Sequence). *A multidimensional sequence on D_{A} is an ordered non empty list of itemsets $\varsigma = \langle i_1, \ldots, i_l \rangle$ where for every j in $[1,l]$, i_j is a multidimensional itemset on D_{A}.*

In our running example, $(Y,c,50)$, $(M,p,2)$, $(M,r,10)$ are three multidimensional items on $D_{\mathrm{A}} = \{A,P,Q\}$. Thus, $\langle \{(Y,c,50),(M,p,2)\},\{(M,r,10)\} \rangle$ is a multidimensional sequence on D_{A}.

Definition 4 (Inclusion of sequence). *A multidimensional sequence $\varsigma = \langle a_1, \ldots, a_l \rangle$ is said to be a subsequence of a sequence $\varsigma' = \langle b_1, \ldots, b_{l'} \rangle$ if there exist $1 \leq j_1 \leq j_2 \leq \ldots \leq j_l \leq l'$ such that $a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, \ldots, a_l \subseteq b_{j_l}$.*

With $\varsigma = \langle \{(Y,c,50)\}, \{(M,r,10)\} \rangle$ and $\varsigma' = \langle \{(Y,c,50),(M,p,2)\},\{(M,r,10)\} \rangle$, ς is a subsequence of ς'.

4.3 Support

Computing the support of a sequence amounts to count the number of blocks that *support* the sequence. Intuitively, a block supports a sequence ς if (i) for each itemset i in ς there exists a date in $Dom(D_t)$ such that all items in i appear at this date, and (ii) all itemsets in ς are successively retrieved at different and increasing dates.

Definition 5. *A table T supports a sequence $\langle i_1, \ldots, i_l \rangle$ if for every $j = 1, \ldots, l$, there exists d_j in $Dom(D_t)$ such that for every item e in i_j, there exists $t = (f,r,e,d_j)$ in T with $d_1 < d_2 < \ldots < d_l$.*

In our running example, the block $(Educ, NY)$ from Fig. 2.a supports $\varsigma = \langle\{(Y,c,50), (M,p,2)\}, \{(M,r,10)\}\rangle$ since $\{(Y,c,50), (M,p,2)\}$ appears at $date = 1$ and $\{(M,r,10)\}$ appears at $date = 2$.

The support of a sequence in a table T is the proportion of blocks of T that support it.

Definition 6 (Sequence Support). *Let D_{R} be the reference dimensions and T a table partitioned into the set of blocks $\mathrm{B}_{T,D_{\mathrm{R}}}$. The support of a sequence ς is defined by:*

$$support(\varsigma) = \frac{|\{B \in \mathrm{B}_{T,D_{\mathrm{R}}} \mid B \text{ supports } \varsigma\}|}{|\mathrm{B}_{T,D_{\mathrm{R}}}|}$$

Definition 7 (Frequent Sequence). *Let $minsup \in [0,1]$ be the minimum user-defined support value. A sequence ς is said to be frequent if $support(\varsigma) \geq minsup$.*
An item e is said to be frequent if so is the sequence $\langle\{e\}\rangle$.

In our running example, let us consider $D_{\mathrm{R}} = \{CG, C\}$, $D_{\mathrm{A}} = \{A, P, Q\}$, $minsup = \frac{1}{5}$, $\varsigma = \langle\{(Y,c,50),(M,p,2)\},\{(M,r,10)\}\rangle$. The three blocks of the partition of T from Fig. 2 must be scanned to compute $support(\varsigma)$.
1. Block $(\mathbf{Educ}, \mathbf{NY})$ (Fig. 2.a). In this block, we have $(Y,c,50)$ and $(M,p,2)$ at date 1, and $(M,r,10)$ at date 2. Thus this block supports ς.
2. Block $(\mathbf{Educ}, \mathbf{LA})$ (Fig. 2.b). This block does not support ς since it does not contain $(M,p,2)$.
3. Block $(\mathbf{Ret.}, \mathbf{SF})$ (Fig. 2.c). This block does not support ς since it contains only one date.
Thus, we have $support(\varsigma) = \frac{1}{3} \geq minsup$.

5 Jokerized Sequential Patterns

Considering the definitions above, an item can only be retrieved if there exists a frequent tuple of values from domains of D_{A} containing it. For instance, it can happen that neither (Y,r) nor (M,r) nor (O,r) is frequent whereas the value r is frequent. In this case, we consider $(*,r)$ which is said to be *jokerized*.

Definition 8 (Jokerized Item). *Let $e = (d_1,\ldots,d_m)$ a multidimensional item. We denote by $e_{[d_i/\delta]}$ the replacement in e of d_i by δ. e is said to be a jokerized multidimensional item if: (i) $\forall i \in [1,m], d_i \in Dom(D_i) \cup \{*\}$, and (ii) $\exists i \in [1,m]$ such that $d_i \neq *$, and (iii) $\forall d_i = *$, $\nexists \delta \in Dom(D_i)$ such that $e_{[d_i/\delta]}$ is frequent.*

A *jokerized* item contains at least one specified analysis dimension. It contains a $*$ only if no specific value from the domain can be set. A *jokerized* sequence is a sequence containing at least one *jokerized* item. A block is said to *support* a sequence if a set of tuples containing the itemsets satisfying the temporal constraints can be found.

Definition 9 (Support of a Jokerized Sequence). *A table T supports a jokerized sequence $\varsigma = \langle i_1,\ldots,i_l\rangle$ if: $\forall j \in [1,l]$, $\exists \delta_j \in Dom(D_t)$, $\forall e = (d_{i_1},\ldots,d_{i_m}) \in i_j$, $\exists t = (f, r, (x_{i_1},\ldots,x_{i_m}), \delta_j) \in T$ with $d_{i_k} = x_{i_k}$ or $d_{i_k} = *$ and $\delta_1 < \delta_2 < \ldots < \delta_l$.*
The support of ς is defined by: $support(\varsigma) = \frac{|\{B \in \mathrm{B}_{T,D_{\mathrm{R}}} \text{ s.t. } B \text{ supports } \varsigma\}|}{|\mathrm{B}_{T,D_{\mathrm{R}}}|}$

6 Algorithms

6.1 Mining Frequent Items

The computation of all frequent sequences is based on the computation of all frequent multidimensional items. When considering no joker value, a single scan of the database is enough to compute them.

On the other hand, when considering jokerized items, a levelwise algorithm is used in order to build the frequent multidimensional items having as few joker values as possible. To this end, we consider a lattice which lower bound is the multidimensional item $(*,\ldots,*)$. This lattice is partially built from $(*,\ldots,*)$ up to the frequent items containing as few $*$ as possible. At level i, i values are specified, and items at this level are combined to build a set of candidates at level $i+1$. Two frequent items are combined to build a candidate if they are $\bowtie$-compatible.

Definition 10 ($\bowtie$-compatibility). *Let $e_1 = (d_1,\ldots,d_n)$ and $e_2 = (d'_1,\ldots,d'_n)$ be two distinct multidimensional items where d_i and $d'_i \in dom(D_i) \cup \{*\}$. e_1 and e_2 are said to be $\bowtie$-compatible if there exists $\Delta = \{D_{i_1},\ldots,D_{i_{n-2}}\} \subset \{D_1,\ldots,D_n\}$ such that for every $j \in [1,n-2]$, $d_{i_j} = d'_{i_j} \neq *$ with $d_{i_{n-1}} = *$ and $d'_{i_{n-1}} \neq *$ and $d_{i_n} \neq *$ and $d'_{i_n} = *$.*

Definition 11 (Join). *Let $e_1 = (d_1,\ldots,d_n)$ and $e_2 = (d'_1,\ldots,d'_n)$ be two $\bowtie$-compatible multidimensional items. We define $e_1 \bowtie e_2 = (v_1,\ldots,v_n)$ where $v_i = d_i$ if $d_i = d'_i$, $v_i = d_i$ if $d'_i = *$ and $v_i = d'_i$ if $d_i = *$.*
Let E and E' be two sets of multidimensional items of size n, we define

$$E \bowtie E' = \{e \bowtie e' \mid (e,e') \in E \times E' \wedge \text{ e and e' are } \bowtie\text{-compatible}\}$$

In our running example, $(NY,Y,*)$ and $(*,Y,r)$ are $\bowtie$-compatible. We have $(NY,Y,*) \bowtie (*,Y,r) = (NY,Y,r)$. On the contrary, $(NY,M,*)$ and $(NY,Y,*)$ are not $\bowtie$-compatible. Note that this method is close to the one used for iceberg cubes in [2,3].

Let F_1^i denote the set of 1-frequent items having i dimensions which are specified (different from $*$). F_1^1 is obtained by counting each value over each analysis dimension, i.e., $F_1^1 = \{f \in Cand_1^1,\ support(f) \geq minsup\}$. Candidate items of size i are obtained by joining the set of frequent items of size $i-1$ with itself: $Cand_1^i = F_1^{i-1} \bowtie F_1^{i-1}$.

Function supportcount
Data : $\varsigma, T, D_{\mathrm{R}}, counting$ //counting indicates if joker values are considered or not
Result : support of ς
$Integer\ support \longleftarrow 0$; $Boolean\ seqSupported$;
$\mathrm{B}_{T,D_{\mathrm{R}}} \longleftarrow \{blocks\ of\ T\ identified\ over\ D_{\mathrm{R}}\}$;
foreach $B \in \mathrm{B}_{T,D_{\mathrm{R}}}$ **do**
  $seqSupported \longleftarrow supportTable(\varsigma, B, counting)$;
  if $seqSupported$ **then** $support \longleftarrow support + 1$;
return $\left(\frac{support}{|\mathrm{B}_{T,D_{\mathrm{R}}}|}\right)$

Algorithm 1: Support of a sequence (supportcount)

```
Function supportTable
Data     : ς, T, counting
Result   : Boolean
ItemSetFound ⟵ false ; seq ⟵ ς ; itset ⟵ seq.first() ; it ⟵ itset.first()
if ς = ∅ then return (true) // End of Recursivity
while t ⟵ T.next ≠ ∅ do
    if supports(t, it, counting) then
        if (NextItem ⟵ itset.second()) = ∅ then ItemSetFound ⟵ true
        // Look for all the items from the itemset
        else
            // Anchoring on the item (date)
            T' ⟵ σ_{date=t.date}(T)
            while t' ⟵ T'.next() ≠ ∅ ∧ ItemSetFound = false do
                if supports(t', NextItem, counting) then NextItem ⟵ itset.next()
                if NextItem = ∅ then ItemSetFound ⟵ true
        if ItemSetFound = true then
            // Anchoring on the current itemset succeeded; test the other itemsets in seq
            return (supportTable(seq.tail(), σ_{date>t.date}(T), counting))
        else
            // Anchoring failure: try anchoring with the next dates
            itset ⟵ seq.first()
            T ⟵ σ_{date>t.date}(T) // Skip to next dates
return(false) // Not found
```

Algorithm 2: supportTable *(Checks if a sequence ς is supported by a table T)*

6.2 Mining Jokerized Multidimensional Sequences

The frequent items give all frequent sequences containing one itemset consisting of a single item. Then, the candidate sequences of size k ($k \geq 2$) are generated and validated against the table T. This computation is based on usual algorithms such as PSP [7] that are adapted for the treatment of joker values.

The computation of the support of a sequence ς according to the reference dimensions D_{R} is given by Algorithm 1. This algorithm checks whether each block of the partition supports the sequence by calling the function supportTable (Algorithm 2). *supportTable* attempts to find a tuple from the block that matches the first item of the first itemset of the sequence in order to *anchor* the sequence. This operation is repeated recursively until all itemsets from the sequence are found (return true) or until there is no way to go on further (return false). Several possible anchors may have to be tested.

7 Experiments

In this section, we report experiments performed on synthetic data. These experiments aim at showing the interest and scalability of our approach, especially in the jokerized approach. As many databases from the real world include quantitative information, we

have distinguished a quantitative dimension. In order to highlight the particular role of this quantitative dimension, we consider four ways of computing frequent sequential patterns: (i) no joker (M^2SP), (ii) jokers on all dimensions but the quantitative one (M^2SP-$alpha$), (iii) jokers only on the quantitative dimension (M^2SP-mu), (iv) jokers on all dimensions (M^2SP-$alpha$-mu). Note that case (iv) corresponds to the jokerized approach presented in Section 5. Our experiments can thus be seen as being conducted in the context of a fact table of a multidimensional database, where the quantitative dimension is the *measure*. In Figures 5-12, minsup is the minimum support taken into account, nb_dim is the number of analysis dimensions being considered, DB_size is the number of tuples, and avg_card is the average number of values in the domains of the analysis dimensions.

Fig. 3 and 4 compare the behavior of the four approaches described above when the support changes. M^2SP-$alpha$ and M^2SP-$alpha$-mu have a similar behavior, the difference being due to the verification of quantities in the case of M^2SP-$alpha$. Note that these experiments are not led with the same minimum support values, since no frequent items are found for M^2SP and M^2SP-mu if the support is too high. Fig. 5 shows the scalability of our approach since runtime grows almost linearly when the database size increases (from 1,000 tuples up to 26,000 tuples).

Fig. 6 shows how runtime behaves when the average cardinality of the domains of analysis dimensions changes. When this average is very low, numerous frequent items are mined among few candidates. On the contrary, when this average is high, numerous candidates have to be considered from which few frequent items are mined. Between these two extrema, the runtime decreases. Fig. 7 and 8 show the behavior of our approach when the number of analysis dimensions changes. The number of frequent items increases as the number of analysis dimensions grows, leading to an increase of the number of frequent sequences. Fig. 9 and 10 show the differential between the number of frequent sequences mined by our approach compared to the number of frequent sequences mined by the approach described in [8], highlighting the interest of our proposition.

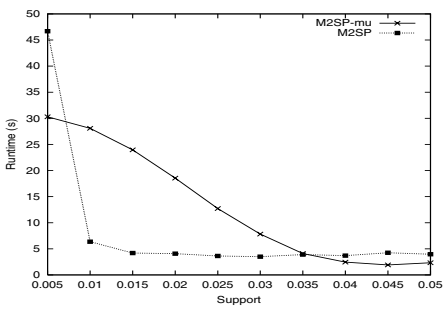


Fig. 3. Runtime over Support (DB_size=12000, nb_dim=5, avg_card=20)

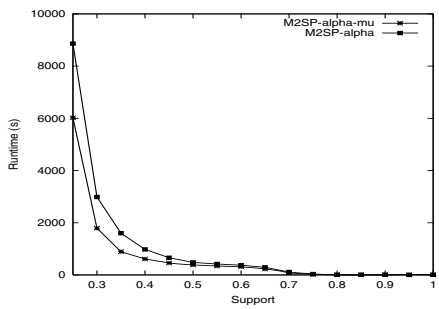


Fig. 4. Runtime over Support (DB_size=12000, nb_dim=5, avg_card=20)

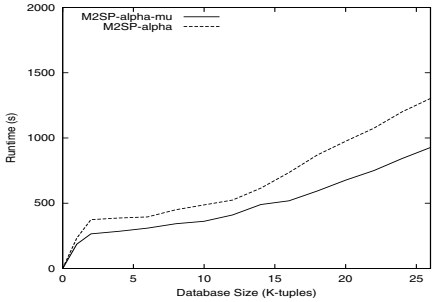


Fig. 5. Runtime over database size (minsup=0.5, nb_dim=15, avg_card = 20)

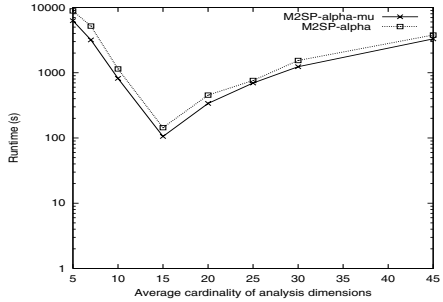


Fig. 6. Runtime over Average Cardinality of Analysis Dimensions (minsup=0.8, DB_size=12000, nb_dim=15)

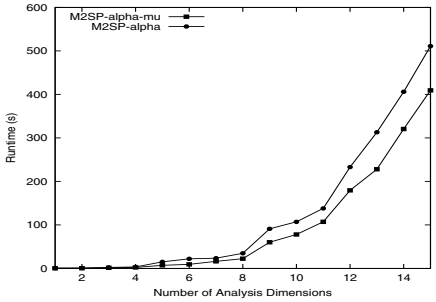


Fig. 7. Runtime over Number of Analysis Dimensions (minsup=0.5, DB_size=12000, nb_dim=15, avg_card=20)

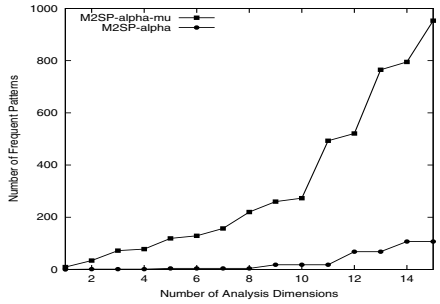


Fig. 8. Number of Frequent patterns over number of analysis dimensions (minsup=0.5, DB_size=12000, nb_dim=15, avg_card=20)

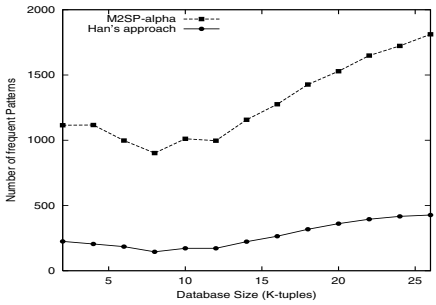


Fig. 9. Number of Frequent Sequences over Database Size (minsup=0.5, nb_dim=15, avg_card=20)

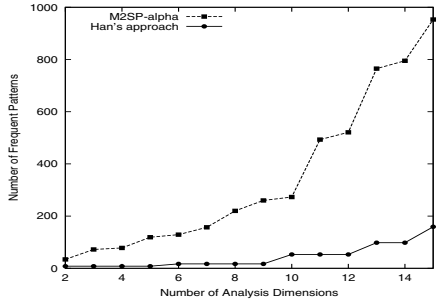


Fig. 10. Number of Frequent Sequences over Number of Analysis Dimensions (minsup=0.5, DB_size=12000, avg_card=20)

8 Conclusion

In this paper, we have proposed a novel definition for multidimensional sequential patterns. Contrary to the propositions [4,8,11], several analysis dimensions can be found in the sequence, which allows for the discovery of rules as *A customer who bought a surfboard together with a bag in NY later bought a wetsuit in LA*. We have also defined *jokerized sequential patterns* by introducing the joker value $*$ on analysis dimensions. Algorithms have been evaluated against synthetic data, showing the scalability of our approach.

This work can be extended following several directions. For example, we can take into account approximate values on quantitative dimensions. In this case, we allow the consideration of values that are not fully jokerized while remaining frequent. This proposition is important when considering data from the real world where the high number of quantitative values prevents each of them to be frequent. Rules to be built will then be like *The customer who bought a DVD player on the web is likely to buy* almost 3 *DVDs in a supermarket later*. Hierarchies can also be considered in order to mine multidimensional sequential patterns at different levels of granularity in the framework of multidimensional databases.

References

1. R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. 1995 Int. Conf. Data Engineering (ICDE'95)*, pages 3–14, 1995.
2. K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cube. In *Proc. of ACM SIGMOD Int. Conf. on Management of Data*, pages 359–370, 1999.
3. A. Casali, R. Cicchetti, and L. Lakhal. Cube lattices: A framework for multidimensional data mining. In *Proc. of 3rd SIAM Int. Conf. on Data Mining*, 2003.
4. S. de Amo, D. A. Furtado, A. Giacometti, and D. Laurent. An apriori-based approach for first-order temporal pattern mining. In *Simpósio Brasileiro de Bancos de Dados*, 2004.
5. T.G. Dietterich and R.S. Michalski. Discovering patterns in sequences of events. *Artificial Intelligence*, 25(2):187–232, 1985.
6. H. Mannila, H. Toivonen, and A.I. Verkamo. Discovering frequent episodes in sequences. In *Proc. of Int. Conf. on Knowledge Discovery and Data Mining*, pages 210–215, 1995.
7. F. Masseglia, F. Cathala, and P. Poncelet. The PSP Approach for Mining Sequential Patterns. In *Proc. of PKDD*, volume 1510 of *LNCS*, pages 176–184, 1998.
8. H. Pinto, J. Han, J. Pei, K. Wang, Q. Chen, and U. Dayal. Multi-dimensional sequential pattern mining. In *ACM CIKM*, pages 81–88, 2001.
9. R. Srikant and R. Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. In *Proc. of EDBT*, pages 3–17, 1996.
10. J.D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume I. Computer Science Press, 1988.
11. C.-C. Yu and Y.-L. Chen. Mining sequential patterns from multidimensional sequence data. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):136–140, 2005.

A Systematic Comparison of Feature-Rich Probabilistic Classifiers for NER Tasks

Benjamin Rosenfeld, Moshe Fresko, and Ronen Feldman

Bar-Ilan University, Computer Science Department, Data-Mining Lab.,
Ramat-Gan, Israel
{freskom1, feldman}@cs.biu.ac.il
http://www.cs.biu.ac.il/~freskom1/DataMiningLab

Abstract. In the CoNLL 2003 NER shared task, more than two thirds of the submitted systems used the feature-rich representation of the task. Most of them used maximum entropy to combine the features together. Others used linear classifiers, such as SVM and RRM. Among all systems presented there, one of the MEMM-based classifiers took the second place, losing only to a committee of four different classifiers, one of which was ME-based and another RRM-based. The lone RRM was fourth, and CRF came in the middle of the pack. In this paper we shall demonstrate, by running the three algorithms upon the same tasks under exactly the same conditions that this ranking is due to feature selection and other causes and not due to the inherent qualities of the algorithms, which should be ranked otherwise.

1 Introduction

Recently, feature-rich probabilistic conditional classifiers became state-of-the-art in sequence labeling tasks, such as NP chunking, PoS tagging, and Named Entity Recognition. Such classifiers build a probabilistic model of the task, which defines a conditional probability on the space of all possible labelings of a given sequence. In this, such classifiers differ from the binary classifiers, such as decision trees and rule-based systems, which directly produce classification decisions, and from the generative probabilistic classifiers, such as HMM-based Nymble [2] and SCFG-based TEG [8], which model the joint probability of sequences and their labelings. Modeling the conditional probability allows the classifiers to have all the benefits of probabilistic systems while having the ability to use any property of tokens and their contexts, if the property can be represented in the form of binary features. Since almost all local properties can be represented in such a way, this ability is very powerful.

There are several different feature-rich probabilistic classifiers developed by different researchers, and in order to compare them, one usually takes a known publicly available dataset, such as MUC-7 [23] or CoNLL shared task [12], and compares the performance of the algorithms on the dataset. However, performance of a feature-rich classifier strongly depends upon the feature sets it uses. Since systems developed by different researches are bound to use different feature sets, the differences in performance of complete systems can not reliably teach us about the qualities of the underlying algorithms.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 217 – 227, 2005.

In this work we compare the performances of three common models (all present in the CoNLL 2003 shared task) – MEMM [15], CRF [16], and RRM (regularized Winnow) [14] – within the same platform, using exactly the same set of features. We also test the effects of different training sizes, different choice of parameters, and different feature sets upon the algorithms' performance.

Our experiments indicate that CRF outperforms MEMM for all datasets and feature sets, which is not surprising, since CRF is a better model of sequence labeling. Surprisingly, though, the RRM performs at the same level or even better than CRF, despite being local model like MEMM, and being significantly simpler to build than both CRF and MEMM.

The following section of the paper we outline the three algorithms. We then present our experiments and their results.

2 Classifiers

The general sequence labeling problem can be described as follows. Given a small set Y of labels, and a sequence $\boldsymbol{x} = \boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_{l(\boldsymbol{x})}$, the task is to find a labeling $\boldsymbol{y} = y_1y_2\ldots y_{l(\boldsymbol{x})}$, where each label $y_i \in Y$. In the framework of feature-rich classifiers, the elements $\boldsymbol{x}_i$ of the sequence should not be thought of as simple tokens, but rather as sequence positions, or *contexts*. The contexts are characterized by a set of externally supplied binary *features*. Thus, each context $\boldsymbol{x}_i$ can be represented as a vector $\boldsymbol{x}_i=(x_{i1},x_{i2},\ldots,x_{ik})$, where $x_{ij} = 1$ if the j-th feature is present in the i-th context, and $x_{ij} = 0$ otherwise.

The feature-rich sequence classifiers have no knowledge of the nature of features and labels. Instead, in order to make predictions, the classifiers are supplied with a training set $T = \{(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)})\}_{t=1..n}$ of sequences with their intended labelings. The classifiers use the training set to build the model of the task, which is subsequently used to label unseen sequences.

We shall describe the particular algorithms only briefly, referring to the original works to supply the details.

2.1 MEMM

A Maximum Entropy Markov Model classifier [4] builds a probabilistic conditional model of sequence labeling. Labeling each position in each sequence is considered to be a separate classification decision, possibly influenced by a small constant number of previous decisions in the same sequence. In our experiments we use a Markov model of order one, in which only the most recent previous decision is taken into account.

Maximal Entropy models are formulated in terms of *feature functions* $f(\boldsymbol{x}_i, y_i, y_{i-1}) \rightarrow \{0, 1\}$, which link together the context features and the target labels. In our formulation, we have a feature function f_{jy} for each context feature j and each label y, and a feature function $f_{iyy'}$ for each context feature and each pair of labels. The functions are defined as follows:

$f_{jy}(\boldsymbol{x}_i, y_i, y_{i-1}) = \boldsymbol{x}_{ij}I_y(y_i)$ and $f_{jyy'}(\boldsymbol{x}_i, y_i, y_{i-1}) = \boldsymbol{x}_{ij}I_y(y_i)I_{y'}(y_{i-1})$, where $I_a(b)$ is one if $a = b$ and zero otherwise. The vector of all feature functions is denoted $\boldsymbol{f}(\boldsymbol{x}_i, y_i, y_{i-1})$.

A trained MEMM model has a real weight λ_f for each feature function f. Together, the weights form the parameter vector $\boldsymbol{\lambda}$. The model has the form

$$(1) \qquad P_{\boldsymbol{\lambda}}(y_i \mid \boldsymbol{x}_i, y_{i-1}) = \frac{1}{Z(\boldsymbol{x}_i, y_{i-1})} \exp\left(\boldsymbol{\lambda} \cdot \boldsymbol{f}(\boldsymbol{x}_i, y_i, y_{i-1})\right),$$

where $Z(x_i, y_{i-1}) = \sum_y \exp\left(\boldsymbol{\lambda} \cdot \boldsymbol{f}(\boldsymbol{x}_i, y, y_{i-1})\right)$ is the factor making the probabilities for different labels sum to one.

Given a model (1), it can be used for inferring the labeling $\boldsymbol{y} = y_1 y_2 \ldots y_{l(\boldsymbol{x})}$ of an unseen sequence $\boldsymbol{x} = \boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_{l(\boldsymbol{x})}$ by calculating the most probable overall sequence of labels:

$$(2) \qquad \boldsymbol{y}(\boldsymbol{x}) := \arg\max_{y_1 y_2 \cdots y_{l(x)}} \sum_{i=1}^{l(x)} \log P_{\boldsymbol{\lambda}}(y_i \mid \boldsymbol{x}_i, y_{i-1}).$$

This most probable sequence can be efficiently calculated using a variant of the Viterbi algorithm.

The model parameters are trained in such a way as to maximize the model's entropy while making the expected value of each feature function agree with the observed relative frequency of the feature function in the training data. Those conditions can be shown to be uniquely satisfied by the model which maximizes the log-likelihood of the training data among all models of the form (1). In order to avoid overfitting, the likelihood can be penalized with a prior $\Pr(\boldsymbol{\lambda})$. Then, the log-likelihood is

$$L_T(\boldsymbol{\lambda}) = \sum_t \sum_{i=1}^{l(x^{(t)})} \log P_{\boldsymbol{\lambda}}(y_i^{(t)} \mid x_i^{(t)}, y_{i-1}^{(t)}) - \Pr(\boldsymbol{\lambda}) =$$

$$= \sum_t \sum_{i=1}^{l(x^{(t)})} \left(\boldsymbol{\lambda} \cdot \boldsymbol{f}(\boldsymbol{x}_i^{(t)}, y_i^{(t)}, y_{i-1}^{(t)}) - \log Z(x_i^{(t)})\right) - \Pr(\boldsymbol{\lambda})$$

and its gradient is

$$\nabla L_T(\boldsymbol{\lambda}) = \sum_t \sum_{i=1}^{l(x^{(t)})} \left(\boldsymbol{f}(\boldsymbol{x}_i^{(t)}, y_i^{(t)}, y_{i-1}^{(t)}) - E_{P\boldsymbol{\lambda}}(\boldsymbol{f}(\boldsymbol{x}_i^{(t)}, Y, y_{i-1}^{(t)}))\right) - \nabla \Pr(\boldsymbol{\lambda}),$$

where

$$E_{P\boldsymbol{\lambda}}(\boldsymbol{f}(\boldsymbol{x}_i^{(t)}, Y, y_{i-1}^{(t)})) = \sum_{y \in Y} P_{\boldsymbol{\lambda}}(y \mid \boldsymbol{x}_i^{(t)}, y_{i-1}^{(t)})\ \boldsymbol{f}(\boldsymbol{x}_i^{(t)}, y, y_{i-1}^{(t)})$$

is the expectation of the feature vector under the model (1).

With a reasonably chosen prior, the function $L_T(\boldsymbol{\lambda})$ is strictly concave, and so can be maximized by any convex optimization algorithm. We use L-BFGS for this purpose.

2.2 CRF

A Conditional Random Fields (CRF) [7] classifier also builds a probabilistic model of sequence labeling. CRF uses the maximal entropy principle to model the labeling of a sequence as a whole, in contrast to MEMM, which builds a model of separate labeling decisions at different sequence positions.

The model is built upon exactly the same vector $\boldsymbol{f}(\boldsymbol{x}_i, y_i, y_{i-1})$ of feature functions as MEMM. The feature functions are summed along a sequence to produce a *sequence feature functions vector*

$$\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{l(x)} \boldsymbol{f}(\boldsymbol{x}_i, y_i, y_{i-1}), \tag{3}$$

which is then used for constructing the maximal entropy model

$$P_{\boldsymbol{\lambda}}(\boldsymbol{y} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \exp\left(\boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})\right).$$

A trained model can be used for inferring the most probable labeling of an unseen sequence. The decomposition (3) allows to use the Vitterbi algorithm almost identically to the MEMM case, except that in (2), instead of $\log P_{\boldsymbol{\lambda}}(y_i \mid \boldsymbol{x}_i, y_{i-1}) = \boldsymbol{\lambda} \cdot \boldsymbol{f}(\boldsymbol{x}_i, y_i, y_{i-1}) - \log Z(\boldsymbol{x}_i, y_{i-1})$, simple $\boldsymbol{\lambda} \cdot \boldsymbol{f}(\boldsymbol{x}_i, y_i, y_{i-1})$ is used. Since $Z(\boldsymbol{x})$ does not depend on labeling, it need not be calculated at all during inference.

To train the CRF model, we need to maximize the model entropy while satisfying the expectation constrains, expressed this time in terms of the sequence feature functions. As before, this is equivalent to maximizing the log-likelihood of the training data, which can also be penalized with a prior to avoid overfitting:

$$L_T(\boldsymbol{\lambda}) = \sum_t \log P_{\boldsymbol{\lambda}}(\boldsymbol{y}^{(t)} \mid \boldsymbol{x}^{(t)}) - \frac{\|\boldsymbol{\lambda}\|^2}{2\sigma^2} = \sum_t \left(\boldsymbol{\lambda} \cdot \boldsymbol{F}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) - \log Z(\boldsymbol{x}^{(t)})\right) - \Pr(\boldsymbol{\lambda}).$$

The gradient is

$$\nabla L_T(\boldsymbol{\lambda}) = \sum_t \left(\boldsymbol{F}(\boldsymbol{x}^{(t)}, y^{(t)}) - E_{P\boldsymbol{\lambda}}(\boldsymbol{F}(\boldsymbol{x}^{(t)}, \boldsymbol{Y}^{(t)}))\right) - \nabla \Pr(\boldsymbol{\lambda}),$$

where $\boldsymbol{Y}^{(t)}$ is the set of label sequences of length $l(x^{(t)})$, and

$$E_{P\boldsymbol{\lambda}}(\boldsymbol{F}(\boldsymbol{x}^{(t)}, \boldsymbol{Y})) = \sum_{\boldsymbol{y} \in \boldsymbol{Y}} P_{\boldsymbol{\lambda}}(\boldsymbol{y} \mid \boldsymbol{x}^{(t)}) \boldsymbol{F}(\boldsymbol{x}^{(t)}, y)$$

is the expectation of the sequence feature functions vector under the model (3).

In order to maximize $L_T(\boldsymbol{\lambda})$, we need a way to calculate $\log Z(\boldsymbol{x})$ and $E_{P\boldsymbol{\lambda}}(\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{Y}))$ for the given sequence $\boldsymbol{x}$. It is possible to do this efficiently, using a variant of the Forward-Backward algorithm. Details can be found in [7] and [19].

2.3 RRM

The Robust Risk Minimization classifier [14] results from regularization of the Winnow algorithm [21]. Winnow is a multiplicative-update online algorithm used for estimating the weights of a binary linear classifier, which has the following general form:

$$y = \operatorname{sign}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}),$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{w}$ is the weight vector, and $y \in \{+1, -1\}$ is the classification decision.

It was shown in [20], that using a risk function of a special form, the regularized Winnow can produce such weights $\boldsymbol{w}$ that

$$P(y = +1 \mid \boldsymbol{x}) \approx (Tr_{[-1,1]}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}) + 1) / 2,$$

where $Tr_{[a,b]}(s) = \min(b, \max(a, s))$ is a truncation of s onto $[a, b]$.

Although the derivation is elaborate, the resulting algorithm is very simple. It consists of iteratively going over the training set $T = \{(\boldsymbol{x}^{(t)}, y^{(t)})\}_{t=1..n}$ (here, $y^{(t)} = \pm 1$), and incrementally updating

$$\begin{aligned} \alpha_t &:= Tr_{[0,2c]}\left(\alpha_t + \eta\left(1 - \frac{\alpha_t}{c} - \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^{(t)} y^{(t)}\right)\right) \\ w_j &:= \mu_j \exp\left(\sum_t \alpha_t x_j^{(t)} y^{(t)}\right) \end{aligned} \tag{4}$$

The α_t are the *dual* weights, initialized to zero and kept between the iterations. c is the *regularization* parameter, η is the *learning rate*, and μ_j is the *prior*.

The $y^{(t)}$ in (4) are binary decisions. In order to use the RRM for sequence labeling task with more than two labels, we can build a separate classifier for each label and then combine them together within a single Vitterbi search.

3 Experimental Setup

The goal of this work is to compare the three sequence labeling algorithms in several different dimensions: absolute performance, dependence upon the corpus, dependence upon the training set size and the feature set, and dependence upon the hyper-parameters.

3.1 Datasets

For our experiments we used four datasets: CoNLL-E, the English CoNLL 2003 shared task dataset, CoNLL-D, the German CoNLL 2003 shared task dataset, the MUC-7 dataset [23], and the proprietary CLF dataset [8]. For the experiments with smaller training sizes, we cut training corpora into chunks of 10K, 20K, 40K, 80K, and 160K tokens. The corresponding datasets are denoted <Corpus>_<Size>, e.g. "CoNLL-E_10K".

3.2 Feature Sets

There are many properties of tokens and their contexts that could be used in a NER system. We experiment with the following properties, ordered according to the difficulty of obtaining them:

A. The exact character strings of tokens in a small window around the given position.
B. Lowercase character strings of tokens.
C. Simple properties of characters inside tokens, such as capitalization, letters vs digits, punctuation, etc.
D. Suffixes and prefixes of tokens with length 2 to 4 characters.

E. Presence of tokens in local and global dictionaries, which contain words that were classified as certain entities someplace before – either anywhere (for global dictionaries), or in the current document (for local dictionaries).
F. PoS tags of tokens.
G. Stems of tokens.
H. Presence of tokens in small manually prepared lists of semantic terms – such as months, days of the week, geographical features, company suffixes, etc.
I. Presence of tokens inside gazetteers, which are huge lists of known entities.

The PoS tags are available only for the two CoNLL datasets, and the stems are available only for the CoNLL-D dataset. Both are automatically generated and thus contain many errors.

The gazetteers and lists of semantic terms are available for all datasets except CoNLL-D.

We tested the following feature sets:
set0: checks properties A, B, C at the current and the previous token.
set1: A, B, C, B+C in a window [-2…0].
set2: A, B, C, B+C in a window [-2…+2].
set2x: Same as set2, but only properties appearing > 3 times are used.
set3: A, B, C, B+C in a window [-2…+2], D at the current token.
set4: A, B, C, B+C in a window [-2…+2], D at the current token, E.
set5: A, B, C, B+C, F, G in a window [-2…+2] , D at the current token, E.
set6: set4 or set5, H
set7: set4 or set5, H, I

3.3 Hyperparameters

The MaxEntropy-based algorithms, MEMM and CRF, have similar hyperparameters, which define the priors for training the models. We experimented with two different priors – Laplacian (double exponential) $Pr_{LAP}(\boldsymbol{\lambda}) = \alpha\Sigma_i|\lambda_i|$ and Gaussian $Pr_{GAU}(\boldsymbol{\lambda}) = (\Sigma_i\lambda_i^2) / (2\sigma^2)$. Each prior depends upon a single hyperparameter specifying the "strength" of the prior. Note, that $\nabla Pr_{LAP}(\boldsymbol{\lambda})$ has discontinuities at zeroes of λ_i. Because of that, a special consideration must be given to the cases when λ_i approaches or is at zero. Namely,

(1) if λ_i tries to change sign, set $\lambda_i := 0$, and allow it to change sign only on the next iteration, and
(2) if $\lambda_i = 0$, and $\left|\frac{\partial}{\partial\lambda_i} L_T(\boldsymbol{\lambda})\right| < \alpha$, do not allow λ_i to change, because it will immediately be driven back toward zero.

In some of the previous works (e.g., [22]) the Laplacian prior was reported to produce much worse performance than the Gaussian prior. Our experiments show them to perform similarly. The likely reason for the difference is poor handling of the zero discontinuities.

The RRM algorithm has three hyperparameters – the prior μ, the regularization parameter c, and the learning rate η.

4 Experimental Results

It is not possible to test every possible combination of algorithm, dataset and hyper-parameter. Therefore, we tried to do a meaningful series of experiments, which would together highlight the different aspects of the algorithms.

All of the results are presented as final microaveraged F1 scores.

4.1 Experiment 1

In the first series of experiments we evaluated the dependence of the performance of the classifiers upon their hyperparameters. We compared the performance of the

Table 1. RRM results on CoNLL-E dataset

	CoNLL-E_40K_set7			CoNLL-E_80K_set7			CoNLL-E_160K_set7		
μ=0.01	c=0.001	c=0.01	c=0.1	c=0.001	c=0.01	c=0.1	c=0.001	c=0.01	c=0.1
η=0.001	78.449	78.431	78.425	81.534	81.534	81.510	84.965	84.965	84.965
η=0.01	**85.071**	**85.071**	84.922	87.766	87.774	87.721	**90.246**	90.238	90.212
η=0.1	82.918	83.025	83.733	**87.846**	87.835	88.031	89.761	89.776	89.904
μ=0.1									
η=0.001	84.534	84.552	84.534	87.281	87.281	87.264	89.556	89.556	89.573
η=0.01	85.782	**85.800**	**85.800**	89.032	89.032	**89.066**	**91.175**	**91.175**	**91.150**
η=0.1	82.439	82.709	83.065	63.032	63.032	63.032	30.741	30.741	56.445
μ=1.0									
η=0.001	85.973	85.973	**85.990**	**89.108**	**89.108**	89.100	**91.056**	**91.056**	**91.056**
η=0.01	83.850	83.877	83.904	88.141	88.141	88.119	90.286	90.317	90.351
η=0.1	0	0	29.937	0	0	0	0	0	0

Table 2. RRM results on other datasets

	CoNLL-D_20K_set7			MUC7_40K_set2x			CLF_80K_set2		
μ=0.01	c=0.001	c=0.01	c=0.1	c=0.001	c=0.01	c=0.1	c=0.001	c=0.01	c=0.1
η=0.001	43.490	43.490	43.453	48.722	48.722	48.650	49.229	49.229	49.244
η=0.01	46.440	46.438	**46.472**	63.220	63.207	62.915	64.000	**64.040**	63.710
η=0.1	44.878	44.943	45.995	61.824	62.128	**63.678**	58.088	58.628	61.548
μ=0.1									
η=0.001	44.674	44.674	44.671	60.262	60.249	60.221	59.943	59.943	59.943
η=0.01	44.799	44.845	**44.957**	65.529	**65.547**	65.516	**64.913**	**64.913**	64.811
η=0.1	43.453	43.520	44.192	60.415	60.958	63.120	55.040	55.677	60.161
μ=1.0									
η=0.001	44.682	44.682	**44.694**	**66.231**	**66.231**	66.174	**65.408**	**65.408**	**65.408**
η=0.01	43.065	43.080	43.195	62.622	62.579	62.825	59.197	59.311	59.687
η=0.1	0	0	6.123	2.922	2.922	8.725	0	0	1.909

Table 3. CRF results on a selection of datasets

CRF	CLF			CoNLL-D			MUC7	CoNLL-E
	20K_set2	40K_set2	80K_set2	40K_set1	80K_set1	160K_set1	80K_set0	80K_set0
GAU σ = 1	**76.646**	**78.085**	**80.64**	29.851	35.516	**39.248**	**80.756**	69.247
GAU σ = 3	75.222	77.553	79.821	28.530	35.771	38.254	80.355	**69.693**
GAU σ = 5	75.031	77.525	79.285	29.901	35.541	38.671	79.853	69.377
GAU σ = 7	74.463	77.633	79.454	**30.975**	**36.517**	38.748	79.585	69.341
GAU σ = 10	74.352	77.05	77.705	29.269	36.091	38.833	80.625	68.974
LAP α=0.01	73.773	77.446	79.071	29.085	**35.811**	38.947	79.738	69.388
LAP α=0.03	75.023	77.242	78.810	31.082	34.097	38.454	79.044	**69.583**
LAP α=0.05	**76.314**	77.037	79.404	30.303	35.494	**39.248**	**79.952**	69.161
LAP α=0.07	74.666	76.329	**80.841**	30.675	34.530	38.882	79.724	68.806
LAP α=0.1	74.985	**77.655**	80.095	**31.161**	35.187	39.234	79.185	68.955

Table 4. MEMM results on a selection of datasets

MEMM	CLF			CoNLL-D			MUC7	CoNLL-E
	20K_set2	40K_set2	80K_set2	40K_set1	80K_set1	160K_set1	80K_set0	80K_set0
GAU σ = 1	**75.334**	**78.872**	**79.364**	**30.406**	35.013	**40.164**	**78.773**	67.537
GAU σ = 3	74.099	75.693	77.278	28.484	**35.330**	40.005	77.295	67.401
GAU σ = 5	73.959	74.685	77.316	28.526	35.043	39.799	77.489	67.870
GAU σ = 7	73.411	74.505	77.563	28.636	34.630	38.531	77.255	67.897
GAU σ = 10	73.351	74.398	77.379	28.488	33.955	37.830	77.094	**68.043**
LAP α=0.01	71.225	74.04	75.721	28.316	34.329	40.074	**78.312**	67.871
LAP α=0.03	72.603	72.967	76.540	29.086	35.159	38.621	77.385	67.401
LAP α=0.05	71.921	**75.523**	75.370	30.425	33.942	39.984	78.262	**67.908**
LAP α=0.07	72.019	74.486	**77.197**	30.118	**35.250**	39.195	76.646	67.833
LAP α=0.1	**72.695**	75.311	76.335	**30.315**	33.487	**40.861**	78.141	67.421

classifiers on a selection of datasets, with different hyperparameter values. All of the algorithms showed moderate and rather irregular dependence upon their hyperparameters. However, single overall set of values can be selected.

The RRM results are shown in the Table 1 and the Table 2. As can be seen, selecting $\mu = 0.1$, $c = 0.01$ and $\eta = 0.01$ gives reasonably close to optimal performance on all datasets. All subsequent experiments were done with those hyperparameter values.

Likewise, the ME-based algorithms have no single best set of hyperparameter values, but have close enough near-optimal values. A selection of MEMM and CRF results is shown in the Table 3 and Table 4. For subsequent experiments we use CRF with Laplacian prior with $\alpha = 0.07$ and MEMM with Gaussian prior with $\sigma = 1$.

4.2 Training Size

In this series of experiments we evaluated the performance of the algorithms using progressively bigger training datasets: 10K, 200K, 400K, 800K and 1600K tokens. The results are summarized in the Fig.1. As expected, the algorithms exhibit very similar training size vs. performance behavior.

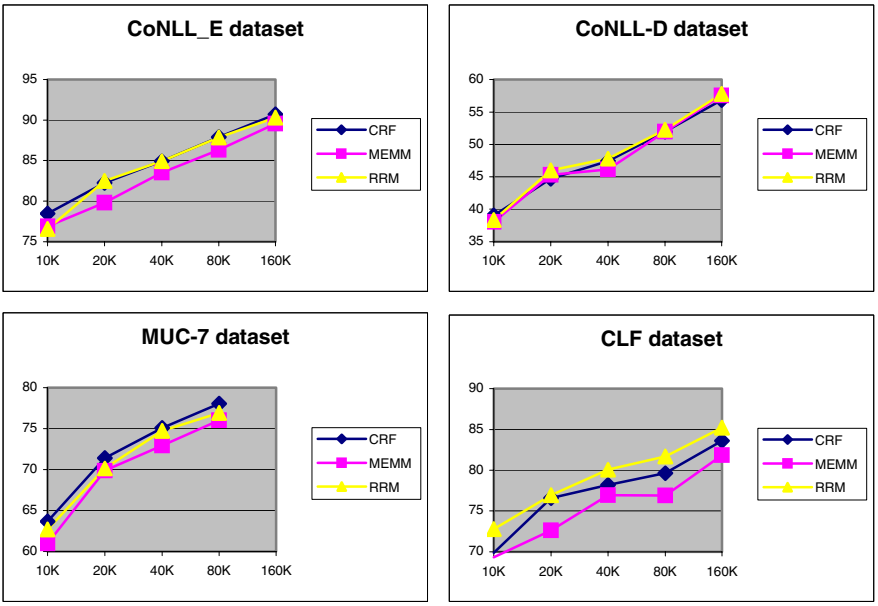


Fig. 1. Performance of the algorithms with different training sizes

Table 5. Performance of the algorithms with different feature sets

	MUC7			CoNLL-D			CoNLL-E		
	CRF	MEMM	RRM	CRF	MEMM	RRM	CRF	MEMM	RRM
set0	**75.748**	66.582	62.206	**48.988**	43.36	40.109	**87.379**	82.281	76.887
set1	**75.544**	67.075	68.405	**50.672**	49.164	48.046	**87.357**	82.516	81.788
set2	**75.288**	74.002	74.755	**52.128**	~52.01	51.537	86.891	87.089	**87.763**
set3	**76.913**	76.333	76.794	~60.172	59.526	**61.103**	88.927	88.711	**89.110**
set4	**78.336**	77.887	77.828	62.79	63.58	**65.802**	~90.037	~90.047	**90.722**
set5				~65.649	65.319	**67.813**	~90.139	~90.115	**90.559**
set6	**78.969**	78.442	78.016				~90.569	~90.492	**90.982**
set7	**81.791**	80.923	81.057				~91.414	90.88	**91.777**

4.3 Feature Sets

In this series of experiments we trained the algorithms with all available training data, but using different feature sets. The results are summarized in the Table 5. The results were tested for statistical significance using the McNemar test. All the perform-

ance differences between the successive feature sets are significant at least at the level p=0.05, except for the difference between set4 and set5 in CoNLL-E dataset for all models, and the differences between set0, set1, and set2 in CoNLL-E and MUC7 datasets for the CRF model. Those are statistically insignificant. The differences between the performance of different models that use same feature sets are also mostly significant. Exceptions are the numbers preceded by a tilda "~". Those numbers are not significantly different from the best results in their corresponding rows.

As can be seen, both CRF and RRM generally outperform MEMM. Among the two, the winner appears to depend upon the dataset. Also, it is interesting to note that CRF always wins, and by a large margin, on feature sets 0 and 1, which are distinguished from the set 2 by absense of "forward-looking" features. Indeed, using "forward-looking" features produces little or no improvement for CRF, but very big improvement for local models, probably because such features help to alleviate the *label bias problem* [7].

5 Conclusions

We have presented the experiments comparing the three common state-of-the-art feature-rich probabilistic sentence classifiers inside a single system, using completely identical feature sets. The experiments show that both CRF and RRM significantly outperform MEMM, while themselves performing roughly similarly. Thus, it shows that the comparatively poor performance of CRF in the CoNLL 2003 NER task [16] is due to suboptimal feature selection and not to any inherent flaw in the algorithm itself.

Also, we demonstrated that the Laplacian prior performs just as well and sometimes better than Gaussian prior, contrary to the results of some of the previous researches.

On the other hand, the much simpler RRM classifier performed just as well as CRF and even outperformed it on some of the datasets. The reason of such surprisingly good performance invites further investigation.

References

1. Aitken, J. S.: Learning Information Extraction Rules: An Inductive Logic Programming approach. 15th European Conference on Artificial Intelligence. IOS Press. (2002)
2. Bikel, D. M., Schwartz, R., Weischedel, R.M.: An Algorithm that Learns What's in a Name. Machine Learning. (34): (1999) 211-231.
3. Chieu, H.L., Tou Ng, H.: Named Entity Recognition: A Maximum Entropy Approach Using Global Information. Proceedings of the 17th International Conference on Computational Linguistics. (2002)
4. McCallum, A., Freitag, D., Pereira, F.: Maximum Entropy Markov Models for Information Extraction and Segmentation. Proceedings of the 17th International Conference on Machine Learning. (2000)
5. Sun A., et al.: Using Support Vector Machine for Terrorism Information Extraction. 1st NSF/NIJ Symposium on Intelligence and Security Informatics. (2003)

6. Kushmerick, N., Johnston, E., McGuinness, S.: Information extraction by text classification. IJCAI-01 Workshop on Adaptive Text Extraction and Mining. Seattle, WA. (2001)
7. Lafferty, J., McCallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proc. 18th International Conf. on Machine Learning. (2001)
8. Rosenfeld, B., Feldman, R., Fresko, M., Schler, J., Aumann, Y.: TEG - A Hybrid Approach to Information Extraction. Proc. of the 13th ACM. (2004)
9. Berger, A., della Pietra, S., della Pietra, V.: A maximum entropy approach to natural language processing. Computational Linguistics 22(1), (1996) 39-71.
10. Darroch, J.N., Ratcliff, D.: Generalized iterative scaling for log-linear models. Annals of Mathematical Statistics (1972). 43(5): 1470-1480.
11. Borthwick, A., Sterling, J., Agichtein, E., Grishman, R.: Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In the proceedings of the 6th Workshop on Very Large Corpora. (1998)
12. Kim Sang, T., Erik, F., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. Edmonton, Canada (2003)
13. Florian, R., Ittycheriah, A., Jing, H., Zhang, T.: Named Entity Recognition through Classifier Combination. In: Proceedings of CoNLL-2003, Edmonton, Canada, (2003), pp. 168-171.
14. Zhang, T., Johnson, D.: A Robust Risk Minimization based Named Entity Recognition System. In: Proceedings of CoNLL-2003, Edmonton, Canada, (2003), pp. 204-207.
15. Chieu, H.L., Tou Ng, H.: Named Entity Recognition with a Maximum Entropy Approach. In: Proceedings of CoNLL-2003, Edmonton, Canada, (2003), pp. 160-163.
16. McCallum, A., Li, W.: Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In: Proceedings of CoNLL-2003, Edmonton, Canada, (2003), pp. 188-191.
17. Joachims, T.: Learning to Classify Text Using Support Vector Machines. Dissertation, Kluwer, (2002)
18. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information (1999) 61—67.
19. Sha, F., Pereira, F.: Shallow parsing with conditional random fields, Technical Report CIS TR MS-CIS-02-35, University of Pennsylvania, (2003)
20. Zhang, T., Damerau, F., Johnson, D.: Text Chunking using Regularized Winnow. Meeting of the Association for Computational Linguistics. (2001) 539-546
21. Zhang, T., Regularized Winnow Methods. NIPS, (2000) 703-709
22. Peng, F., McCallum, A.: Accurate Information Extraction from Research Papers Using Conditional Random Fields. (1997)
23. Chinchor, N. MUC-7 Named Entity Task Definition Dry Run Version, Version 3.5. Proceedings of the Seventh Message Understanding Conference. (1998)
24. Borthwick, A., Sterling, J., Agichtein, E., Grishman, R. Description of the MENE Named Entity System as Used in MUC-7. Proceedings of the Seventh Message Understanding Conference. (1998)

Knowledge Discovery from User Preferences in Conversational Recommendation

Ma ıa Sa a , Ja e Rei , L a e McGı , a d Ba S h

Smart Media Institute, University College Dublin,
Belfield, Dublin 4, Ireland
{maria, james.d.reilly, lorraine.mcginty, barry.smyth}@ucd.ie

Abstract. Knowledge discovery for personalizing the product recommendation task is a major focus of research in the area of conversational recommender systems to increase efficiency and effectiveness. Conversational recommender systems guide users through a product space, alternatively making product suggestions and eliciting user feedback. Critiquing is a common and powerful form of feedback, where a user can express her feature preferences by applying a series of directional critiques over recommendations, instead of providing specific value preferences. For example, a user might ask for a '*less expensive*' vacation in a travel recommender; thus '*less expensive*' is a critique over the *price* feature. The expectation is that on each cycle, the system discovers more about the user's *soft* product preferences from minimal information input. In this paper we describe three different strategies for knowledge discovery from user preferences that improve recommendation efficiency in a conversational system using critiquing. Moreover, we will demonstrate that while the strategies work well separately, their combined effort has the potential to considerably increase recommendation efficiency even further.

1 Introduction

Rec e de e a edge dı c e ech 1 e decıde hıch ec e da 1 a e he 1 ab e f each e d 1 g a 1 e c e 1 e ac 1 . I hı a e , e f c c e a 1 a ca e-ba ed ec e de [1] hıch he e a ıga e d c ace b c bı 1 g ıdea a d ech g ıe f 1 f a 1 e ıe a , a 1 cıa 1 e ıge ce a d e de 1 g. A a f hei c c ıc ec e da 1 ce , c e a 1 a e a ı e ıe e d c ha e ec e efe e ce b e 1 1 g e ıde 1 1 a feedbac 1 each c c e. I 1 e ec ed ha e he c e f a ec e da 1 e 1 ha he ec e de ea e ab e efe e ce a d he ef e 1 ca a 1 1 he dı c e f ec e da 1 edge hıch 1 1 1 e d c ha be a ı f he e efe e ce [2,3]. Ad a age f he a ach 1 c de: (1) e ha e e c e he a ıga 1 ce [4]; a d (2) e a e g ıded a ge d c fa e ha a da d b 1 g a d a e a ı e ec e da 1 a ache [5,6].

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 228–239, 2005.

Rec . . e de . . e . ca be dı 1 g ı hed b he e . f feedbac ha he . . . . ; e a e 1 c de . . . . . a d . . . . . [7]. I hı a e e a e e ecıa 1 e e ed 1 a f . . f . e feedbac ca ed . . . . . [8], he e a . e . 1 dıca e a dı ec 1 . a fea- . e . efe e ce 1 . e a 1 . . he c . . e . ec . . e da 1 . . F . e a e, 1 a . a e / aca 1 . . ec . . e de . , a . e . ıgh 1 dıca e ha he 1 1 e e ed 1 a . aca 1 . ha 1 . . . ha he c . . e . ec . . e ded . 1 . ; 1 hı 1 . a ce, . . . 1 a c 1 1 e e he . . . . fea . e.

Wı hı he . ec . . e de . . e 1 e a . e he ba ıc ıdea . f c 1 1 1 g ca be . aced bac . he e 1 a . . . f B . e . . [4]. F . e a e, E . ee 1 he 1 e . e ıa . ec . . e de . . e ha e . . c 1 1 1 g (a . . . . e 1 e . efe . ed . a . . . .) 1 he . e a . a d . a ı . E . ee a . . e . . c 1 1 e . e a . a fea . e ch a . . . . . e c. I . . a , c 1 1 1 g 1 a g . d e a e . f a . 1 1 a . e feedbac a . . ach he e he . e d e . . eed . . . . ıde a . . f . ec ı c . efe e ce 1 f . . a 1 . , hı e a he . a e 1 e 1 he . he . ec . . e de . . a . . 1 . ea ch f c . 1 e . ıg 1 ca [8]. A . ec . . e de . . e . bec . e . . e c . . . . . ace, he e ha bee . e e ed 1 e . - e 1 c 1 1 1 g, 1 h he . a . . f c . . f e ea ch 1 1 c ea 1 g he e cıe c . f . ec . . e da 1 . dıa . g e [9]. F . he . . e, . ece . e ea ch ha hıgh ıgh ed he 1 . . a ce f 1 e ıga 1 g ech 1 e f . a . . a 1 g he dı c e . f 1 ıcı . efe e ce . . edge hı e . e 1 1 g . 1 1 a 1 f . . a 1 . 1 f . . he . e [3].

I hı a e e de c ıbe h ee . a egıe f . . . edge dı c . e f . . e . efe e ce ha 1 . . e he e f . . a ce . f a c 1 1 e-ba ed . ec . . e de . . S ecı ca , e b 1 d . . . . . e 1 . de c ıbed b [10], he e he ıdea 1 . c . . ıde a . e . ' c 1 1 1 g hı . . , a e a he c . . e c 1 1 e he . a - 1 g . e . ec . . e da 1 . . . Thı a . . ach ead . . ıg 1 ca 1 . . . e e . 1 . ec . . e da 1 . e cıe c . We c . 1 e 1 hı a e b c . . ıde 1 g he hı . . f c 1 1 e a a . . . . . hıch de e . 1 e he e . efe e ce 1 a e 1 . . We . e e a ca e dı c . e . . a eg , a fea . e dı c . e . . a eg a d a e dı c . e . . a eg hıch ha e he . e ıa . f c . . a ıd . . . a 1 fac . . . . d- c ca e . F1 a , e h ha b c . bı 1 g a h ee . a egıe , e ca f . he 1 . . . e e a . ec . . e da 1 . e cıe c .

2 Background

Thı . ec 1 . de c ıbe he 1 c e e a c 1 1 1 g [10] a . . ach, hıch . e . a . . be e . 1 . ec . . e da 1 . e cıe c . . e he ba ıc c 1 1 1 g a . . ach a de c ıbed b [11]. We c . . ıde he 1 c e e a c 1 1 1 g a . . ach a he ba 1 f . . . . . edge dı c . e . . a egıe beca e 1 a . . c . . ıde . a . e . ' c 1 1 1 g hı . . a a ba ıc . a 1 g . 1 .

The 1 c e e a c 1 1 1 g 1 e e a 1 . a . . e a c . . e . a 1 . a . ec . - . e de . . e 1 he . e f E . ee [11]. Each . ec . . e da 1 . e 1 . . a . 1 h a 1 1 1a . e e . e 1 g 1 he . e . ıe a . f a ca e 1 h he hıghe . . . . . The . e . 1 ha e he . . . . 1 . acce hı ca e, he eb e dı g

```
q: query, CB: CaseBase, cq: critique, c_r : current recommendation, U : User model

define Incremental_Critiquing(q, CB)
  cq:= null
  U:= null
  begin
    do
      c_r ← ItemRecommend(q, CB, cq, U)
      cq ← UserReview(c_r, CB)
      q  ← QueryRevise(q, c_r)
      U  ← UpdateModel(U, cq, c_r)
    until UserAccepts(c_r)
  end

define ItemRecommend(q, CB, cq, U)
  CB' ← {c ∈ CB | Satisfies(c, cq)}
  CB' ← sort cases in CB' in decreasing Quality
  c_r ← most quality case in CB'
return c_r

define UserReview(c_r , CB)
  cq ← user critique for some f ∈ c_r
  CB ← CB - c_r
return cq

define QueryRevise(q, c_r)
    q ← c_r
return q

define UpdateModel(U, cq, c_r)
    U ← U - contradict(U, cq, c_r)
    U ← U - refine(U, cq, c_r)
    U ← U + (<cq, c_r>)
return U
```

Fig. 1. The incremental critiquing algorithm

he ec e da 1 e 1 , c 1 1 e 1 a a ea 1 e ce he e c c e. A 1 1 ed e 1 f he 1 c e e a c 1 1 1 g a g 1 h 1 g1 e 1 Fig e 1.

The 1 c e e a c 1 1 1 g a g 1 h c 1 f 4 e e : (1) a e ca e c_r 1 he e ba ed he c e e a d he e 1 c 1 1 e ; (2) he e he ec e da 1 a d a 1e a d1 ec 1 a fea e c 1 1 e, cq; (3) he e , q 1 f he e c c e; (4) he e de , U 1 da ed b addi g he a c 1 1 e cq a d 1 g a he c 1 1 e ha a e 1 c 1 e 1 h 1 . The ec e da 1 ce e 1 a e e1 he he he e 1 e e ed 1 h a 1 ab e ca e, he he g1 e .

I a , he ec e da 1 ce 1 1 e ced b he e de f e 1 c 1 1 e , U, ha 1 1 c e e a da ed each c c e. I c e e a c 1 1 1 g di e he ba 1c c 1 1 1 g a g 1 h . I ead f de 1 g he e ed ca e he ba 1 f he1 1 1 a 1 he ec e d ca e, 1 a c e a c e (ee E a 1 1) f each ca dida e ca e. The c a 1b1 1 c e 1 e e 1a he e ce age f c 1 1 e 1 he e de ha h1 ca e a 1 e . The , he c a 1b1 1 c e a d he ca dida e' (c') 1 1 a 1 he c e ec e da 1 (c_r) a e c b1 ed 1 de b a1 a e a c e (ee E a 1 2, b defa $\beta = 0.75$). The a 1 c e 1 ed a he e ed ca e 1 he e ec e da 1 c c e (ee 1 e 14 1 Fig e 1) a d he ca e 1 h he highe a 1 1 he ch e a he e ec e da 1 .

$$Compatibility(c', U) = \frac{\sum_{\forall_i} satisfies(U_i, c')}{|U|} \tag{1}$$

$$Quality(c', c_r, U) = \beta \cdot Compatibility(c', U) + (1 - \beta) \cdot Similarity(c', c_r) \tag{2}$$

A g 1 h 1 a1 a1 a c 1 1 e-ba ed e de hich 1 c ed f h e c 1 1 e ha ha e bee ch e b he e fa . O e f he e 1

f c ed he 1 c e e a c 1 1 1 g a ach 1 he a1 e a ce f he e de hich e e he e 1 e ce f c 1 1 e ha a be 1 c 1 e 1 h ea 1e c 1 1 e . The e de 1 a1 a1 ed 1 g 1b e ac 1 : (1) 1 g e 1 c 1 1 e ha a e 1 c 1 e 1 h he c e c 1 1 e; (2) e 1 g a e 1 1 g c 1 1 e , f hich he e c 1 1 e 1 a e e e .

3 Knowledge Discovery Strategies

Th1 ec 1 e e h ee a egie f edge d1 c e f e ef- e e ce ha ha e he e 1a 1 e d c ec e da 1 . The a eg d1 c e h e ca e ha be e a 1 f he e efe e ce . The ec d a eg dea 1 h fea e d1 e 1 a 1 , a 1 g 1 acc e efe e ce d1 c e he e a 1 e 1 a ce f each fea e f c 1 g 1 1 a 1 . F1- a , h1 a e e e a e d1 c e a eg ha fac1 1 a e a ge h gh he d c ace ba ed he c e e c 1 1 e. A f he ha e a c f da 1 , he e 1 he e' h1 f c 1 1 e d1 c e ec e da 1 edge f he e efe e ce , 1 de e a 1 e a d f c 1 e a 1d a 1 fac d c ca e .

3.1 Discovering Satisfactory Cases: Highest Compatibility Selection

A e be 1 h he a da d 1 c e e a c 1 1 1 g [10] a ach 1 ha he e a e g a a ee ha he ec e da 1 1 e 1 c e e a 1 f a e' efe e ce . Th1 1 a ge d e he de 1 g ca e e ec 1 a eg h1ch a e age he c a 1b1 1 (1 h a c 1 1 e) a d 1 1 a 1 (1 h he c e efe e ce ca e). Wha e e 1 a e a eg f d- c ec e da 1 , [illegible] (HCS), ha a he ec e de e ec he [illegible] ca e ; 1.e., h e ca e ha a e c a 1b e 1 h e efe e ce . Th1 a 1 c a 1b1 1 a eg ca be ea 1 1 d ced 1 he 1 c e e a c 1 1 1 g a g 1 h .

F1g e 2 de a e ha he ced e h1ch 1 a ec ed 1 he 1 c e- e a c 1 1 1 g a g 1 h 1 he [illegible] e . A bef e, he 1 f e a1 1 g ca e 1 e ed 1 g he c e c 1 1 e . I add1 1 e e a e added. F1 , he ec e de c e he c a 1b1 1 c e, a de a1 ed 1 E a 1 3. I 1 1 a e ha he c a 1b1 1 f c 1 ha a bee d1 ed, a e a1 ed be 1 E a 1 3. I ead f a e ag1 g he c a 1b1 1 a d he 1 1 a 1 , a 1 d e 1 h 1 c e e a c 1 1 1 g, ec d e a e b e he ca e 1 h he h1ghe c a 1b1 1 f he 1 f e a1 1 g ca e . I a , 1 a ach, he e a1 1 g ca e 1 h h1ghe c a 1b1 1 a e, CB'', 1 e ce he d c ec e da 1 . P d1 e e , he a eg 1 1 1 e h e ca e ha a 1 f he a ge be f c 1 1 e ade b he e e 1 e.

The compatibility function. We ha e c 1de ed ca e d1 c e be a 1 1 a 1 b e 1 h1ch e a e 1 g ec e d ca e ha a 1 a

```
q: query, CB: CaseBase, cq: critique, c_r : current recommendation, U: User Model

1.define ItemRecommend(q, CB, cq, U)
2.  CB' ← {c ∈ CB | Satisfies(c, cq)}
3.  CB' ← sort cases in CB' in decreasing compatibility score
4.  CB'' ← selects those cases in CB' with highest compatibility
5.  CB'' ← sort cases in CB'' in decreasing order of their sim to q
6.  c_r ← most similar case in CB''
7.return c_r
```

Fig. 2. Adapting the incremental critiquing algorithm *ItemRecommend* procedure to improve focus on recommendation by using Highest Compatibility Selection strategy

[illegible] a 1 f he [illegible] efe e ce . F [illegible] hı [illegible] ea [illegible] , e e a a e he e a 1 1 g ca e a 1f he e e a e f a e 1 a [illegible] (RLP) [12], hıch c [illegible] 1 [illegible] f a 1 1 1 g he [illegible] f f e a d 1 a e f a e . Rel f ce e Lea 1 g he 1 a ba ed [illegible] (FMDP).

Each ca e 1 ea ed a a a e h e c a 1b1 1 c e 1 da ed a each c c e 1 g a M e-Ca a e f c 1 (ee E a 1 3). Thı f c 1 e a- a e he [illegible] f each a e f he [illegible] 1b e a e a e he c e e e f e a 1 1 g ca e e a e ha ce acc d 1 g he c 1 1 e he e ha e ec ed.

$$Compatibility(c', U_f) = \begin{cases} comp(c') + \alpha \times (1 - comp(c')) & \text{[illegible]} \; U_f \\ comp(c') + \alpha \times (0 - comp(c')) & \text{[illegible]} \; U_f \end{cases} \quad (3)$$

O g a 1 a 1 a a 1 f a he e efe e ce . Th , e a e 1 g f a e f a 1 a c a 1b e ca e (1.e., h e ca e hıch ha e he hıghe c a 1b1 1 (,) a e c 1de 1 g a he e efe e ce (U) a c 1- 1 e). A he beg 1 1 g f each e 1 each ca dıda e ca e, ', ha a defa c a 1b1 1 a e (1.e., , (')). Thı a e 1 da ed e c c e a - 1 g 1 acc he a 1 fac 1 f he c e c 1 1 e. The α a a e e 1 E a 1 3 1 he ea 1 g a e hıch 1 a e 0.1 0.2 a e ; a a ge a e ead a a ge ga be ee ca e 1 ea age . I ca e, he ea 1 g a e 1 1 a 1 ce e a e 1 g f e e f a 1 fac 1 . I he d , e a e 1 g b a a e f a e ha a 1 e a 1c a 1b e a 1.0 a e, a a 1 d e 1 RLP.

I 1 1 a e ha E a 1 3 da e he c a 1b1 1 a e ed b each ca e acc d 1 g he a e c 1 1 e (U_f) a ed c - 1 g a he e f c 1 1 e 1 e he 1 c e e a a ach (ee E a 1 1). The $Compatibility(c', U_f)$ a e c ed 1 he c e c c e 1 be he ($comp(c')$) 1 he e c c e.

3.2 Discovering Important Features: Local User Preference Weighting

The e 1 a eg hıgh 1gh he ca e d 1 e 1 a 1 b e . I he d , 1 1 f c ed d 1 c e 1 g ca e ha a 1 a a 1 f e efe e ce .

N , e e a a eg ha c ce a e he fea e d1 e 1 a1 . We e a (LW) a eg ha d1 c e he e a- 1 e1 a ce f each fea e1 each ca e a a eigh1 g a e f c 1 g he 1 1 a1 , a 1 g1 acc e efe e ce .

O LW a eg f he d1 c e f fea e edge1 ba1ca 1 a ed b he e1 edge d1 c e a eg . A e ha e e a1 ed1 Sec 1 3.1, he d1 c e f ca e edge1 ba ed a 1 11 g e efe e ce , h1ch ea e a e 1 g f he c a1b e ca e . The e ca e a e 1 e 1 1 a he1 c1 1 ed fea e a d he1 d1 e e ce a1 be g h e fea e ha ha e e bee c1 1 ed. S , he a1 f LW a eg 1 1 1 1 e he 1 1 a1 f h e fea e ha ha e e bee c1 1 ed.

$$weight(c'_f) = 1 - \left(\frac{\#critiques\ in\ U\ that\ satisfy\ feature_f\ in\ case\ c'}{\#total\ critiques\ feature_f\ in\ U} \times 0.5 \right) \tag{4}$$

We ge e a e a fea e eigh ec f each ca e, a h 1 E a1 4. A fea e ha ha bee c1 1 ed 1 a e a eigh a e f 1.0 a d a dec e e 1 be a 1ed he a c1 1 e1 a1 ed b he ca e. A ch, he fea e eigh 1 be 1 a he be f 1 e a c1 1 e h1 fea e1 a1 ed b he ca e. H e e, a 1 ca be ee 1 E a1 4 he eigh e e dec ea e a 0 a e. F e a e, 1 a a e aca1 ec e de 1 h a e de ha c a1 c1 1 e [1ce, >, 1000] a d [1ce, >, 1500], a ca e 1 h fea e { } h e 1ce1 2000 1 ha e a 1ce eigh a 0.5 a e beca e1 a1 e b h c1 1 e he ea he d a1 eigh 1 be 1.0 beca e he e1 c1 1 e h1 fea e. I 1 1 a eca ha he e 1dea he e1 1 1 1 e he 1 1 a1 f h e fea e ha ha e e bee c1 1 ed1 a g1 e e 1 .

O a1 d1 c e he be d c ec e d b e 1 1 g he 1 1 a1 f h e fea e ha be d1 e e 1a e he h1ghe c a1b e ca e . T achie e h1 , a ca d1da e' (c') 1 1 a1 he ec e ded ca e (c_r) 1 c ed a each c c e1 he ec e de e a h b E a1 5.

$$Similarity(c', c_r) = \sum_{\forall_f} weight(c'_f) \times similarity(c'_f, c_{r_f}) \tag{5}$$

The 1 1 a1 be ee he ca d1da e ca e (c') a d he ec e ded ca e (c_r) f each fea e f 1 c b1 ed 1 h he eigh f h1 fea e. The eigh 1 c ed e1 1 g E a1 4.

3.3 Discovering Query Knowledge: Binary Search

The1 c e e a c1 1 1 g a ach1 ce 1b e fea e-c1 1 e e e1 1 ha e a 1 cha ge 1 he e e a fea e a e f each c c e he e . We e ha h1 1 a ge d e he 1 ea ea ch 1c 1 e a 1ga e h gh he a e- ace f he c1 1 ed fea e. The e

ı ha he ec e da ı e a e e , h gh he ace f ıbe a e a ı e . I hı ec ı e de c ıbe h he ı c e e a c ı ı ı g a g ı h ca be ea ı a e ed facı ı a e , h gh he a e ace f edge dı c e f a gı e fea e b a ı g a e e cıe (BS) a ach.

```
q: query, CB: CaseBase, cq: critique, c_r : current recommendation

1.   define QueryRevise(q, cq, CB, c_r)
2.   begin
3.       q  ← c_r
4.       CB′ ← {c ∈ CB | Satisfies(c, cq)}
5.       CB′ ← eliminate cases that conflict with prior critiques
6.       f_cq  ← set value in q for critiqued feature f ∈ c_r by Eq. 6
7.     return q
8.   end
```

Fig. 3. Illustrating the binary search procedure for query discovery

Fıg e 3 de a e h he ı c e e a a g ı h ca be ea ı e e ded ed a ach. The ced e hıch ı a ec ed ı he e f he ı c e e a c ı ı ı g a g ı h . The e e ı - da ed ı h a f he fea e f he c e ec e da ı , c_r. I addı ı e e a e added. Fı , he ec e de e ga he a f he a aı - ab e ca e ha a ı f c e fea e c ı ı e (ee ı e 4 f Fıg e 3). The ec d e ı e de e ı ı g he a e-cha ge he c ı ı ed fea e ı a e ı he e ı ed e , q, ed f e ıe a f he e ec e da ı . I a , ı a ach a he e aı ı g ca e , CB', ı e ce he a a e. The e a e a a ache ha c d be ed c e hı a e. I hı a e e e a ı e he ıbı ı f c ı g he edıa (ee e a ı 6) f a ca e ı CB'.

The ec e de e c ec a f he a e a ı e a e ıbı ı ıe f he c ı ı ed fea e f he ca e c e ed b CB'. F ı a ce, ıf he c ı ı ed fea e e e [ıce, <, 2000] he ec e de d ga he a a e ı ha e e e ha 2000 f he e f e aı ı g ca e (e.g., 1800, 1650, 1600, 1570, 1460, 1350, e c.). E a ı 6 a ıg a a e f he c ı ı ed fea e $f_{cq} \in q$ b ca c a ı g he a e age fea e a e e a he e e a ca e .

$$f_{cq} = \begin{cases} CB'_{n+1/2}(\quad) & \# \\ \frac{CB'_{n+1/2}(f \text{ of } cq)+CB'_{(n+1/2)+1}(f \text{ of } cq)}{2} & \# \end{cases} \tag{6}$$

F E a ı 6 ı ı a ed ha he e aı ı g ca e ı , CB' a e ed ı a ce dı g de . He e $CB'_i(\quad)$ ı he fea e a e c ı ı ed b cq ı he i^{th} ca e. The edıa a e c e d a c a ı e e ce age f 50% (ı.e., 50% f he a e a e be he edıa a d 50% f he a e a e

ab e he edıa). We ace he c 1 1 ed fea e 1 a ce dı g a e de a d d he ıdd e a e ıf he be f ca e ı dd d he ıdd e aı a d c e he ea a e be ee he ıf e ha e a e e be f ca e .

O e ı a ı , ha a eed be c ıde ed, ı e ı c 1 1 e he a e fea e. F e a e, e ha a e ha a ed ı a e ı c c e f a , aca ı ha a 2500 ec e da ı a d, ı he c e c - c e, he e a ha he efe a , aca ı ha a 1000 aca ı . I ch ı a ı ı he c e c c e, a he ca e ı c dı g h e ha e ceed a 2500 aca ı ı a ı f he c e c 1 1 e , aca ı ha 1000. If e c - e he edıa a e a ge ı he ea ch ace, e a ı c de h e ca e e ec ed e ı b he e . T a ıd he e ı a ı , e e he hı f c 1 1 e a ıed b he e ı e de c c ec he ea ch ace. The e ı c 1 1 e ed ı he e de a e ea ed a a e f [13] ha a c he be f e aı ı g ca e ha ı be ed c e he edıa a e.

S , f ı g he ea ıe e a e, e c ıde c ı g he edıa f h e ca e ha a e, aca ı ha 1000 a d, aca ı ha 2500. A de aı ed ı ı e 5 f Fıg e 3, bef e c ı g he edıa , e chec f he e ı e ce f e ı a ıed c 1 1 e ha c e he ı c ı f ca e ı CB', a d e ı ı a e he e ca e f f he c ıde a ı . P dı e e , e e ı c 1 1 e decıde ha ca e h d be c e ed b CB', a d ı a e e he a e e ec ı b d f f_{cq}.

The e ı a ı behı d e e ı ı c e e a c 1- 1 ı g a ed ce c 1 1 e e e ı ı e e ce , a d ı e ec e da ı e cıe c b dı c e ı g a ı fac d c f e e a ıd . I h , hı bı a ea ch e a ach e ab e he ec e de f c ı ea ch h e ca dıda e ca e ha : (1) a ı f he c e c 1 1 e; (2) f e ı a ıed c 1 1 e ; a d (3) a e ı ı a he c e ca e b f he a a f ı , a d h ha e he ca abı ı f a ıga ı g he ea ch ace f ı ıc .

4 Evaluation

I hı a e fa e ha e a g ed ha he ı c e e a f f c 1 1 ı g ı ı ed b ı e de c ec e d ca e ha d a ı a a ı f he e efe e ce . We e h ee a egıe ha aıd edge dı c e ı a e ı e e ıe a acc ac a d ec e da ı e cıe c . Thı ec ı de c ıbe he e a ed e a a ı e h d g ha e ed a d he e ha e ed.

4.1 Setup

The e a a ı a e f ed ı g he a da d T a e da a e (a aı ab e f , ,) hıch c ı f 1024 aca ı ca e . Each ca e ı de- c ıbed ı e f 9 fea e ı c dı g , , , e c. The da a e a ch e beca e ı c aı e ıca a d ı a fea e a d ı a ıde a ıde ea ch ace.

We e a a e he hıghe c a ıbı ı e ec ı (HCS), he ca e efe- e ce eıgh ı g (LW), he bı a ea ch (BS) a d a a a egıe c bı ed ı ec e de (ALL) e ı c e e a c ı ı ı g (ı c e e a).

4.2 Methodology

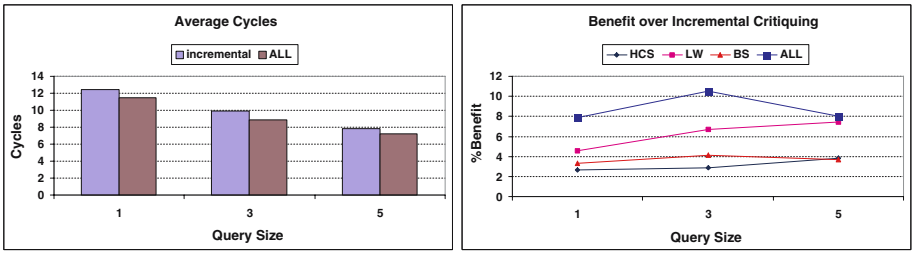
We d ı e ha e ca ıed a ı e e a a ı ı h ı e- e , b f - a e hı a ıb e. A a a e a ı e e ed f a ı e e a a ı ı ı a he e de c ıbed b [14]. Acc dı g , each ca e (hıch a e ca ed he 'ba e') ı he ca e-ba e ı e a ı e ed a d ed ı a . Fı , ı e e a a ba ı f a e f e ıe b a ı g a d b e f ı fea e . We f c b e f 1, 3 a d 5 fea e a dı ı g ı h be ee , , a d e ıe e ec ı e . Sec d, e e ec he ca e ha ı ı ı a he ıgı a ba e. The e ca e a e he ec e da ı a ge f he e e ı e . Th , he ba e e e e he ıdea e f a e , he ge e a ed e ı he ı ı ıa e ıded b he ' e ', a d he a ge ı he be a aı ab e ca e f he e . Each ge e a ed e ı a e b e f he ec e de , a d ı each ec e da ı c c e he ' e ' ıc a c ı ı e ha ı c a ıb e ı h he a ge ca e; ha ı , a c ı ı e ha he a ıed he e aı ı g ca e , e ı he a ge ca e beı g ef ı he e ed e f ca e . Each ea e- e- a h gh he ca e-ba e ı e ea ed 10 ı e a d he ec e da ı e ı e ı a e he he a ge ca e ı e ed.

Re a ed ea e dıe [15] ha e hıgh ıgh ed dı c e a cıe be ee he ıg- ı a [14] a ı cıa e de c c ı a d ea e beha ı ı h e ec c ı ı ed a ıca ı a d e e ı ı . We e a dı ed a ı cıa e de ha ı ı f ed b ea - e dıe . The e de ı de ıg ed e d ec e da ı ı a a e ha ı e c ı e ı h he e e ec ded f ea - e . I a ıc a , a ı cıa e de e ea c ı ı e e ec ı d ı g ec e da ı e ı ı ı a ge fea e a e a e e . F e a e, e a ı cıa e ı ı g f a 3- ee aca ı a d he a e e e ed a 3-da cı -b ea . The a e ı e a f a aca ı b c ı ı ı g he fea e. I hı e a a ı , he a ı cıa e ı c ı e c ı ı e a fea e ı ı ' efe e ce c a ı ı a ı ed.

4.3 Recommendation Efficiency

We a a e he ec e da ı e cıe c b hıch e ea a e age ec - e da ı e ı e g h he c a ı g he e a egıe ı c e e a c ı ı ı g. Fıg e 4(A) e e a g a h c a ı g he a e age e ı e g h f he ı c e e a c ı ı ı g a ach he c bı a ı f a he a egıe (ALL) f 3 dı e e ı ı ıa e e g h . The h ee a egıe c bı ed c ı - e ed ce a e age e ı e g h he c a ed he ı c e e a c ı ı ı g a ach, de a ı g he e ıa ı e ec e da ı e cıe c . F e a e, f he ha d e ıe he ı c e e a ec e de e ı a a e age e ı f 12.46 c c e hı e he c bı ed ec e de e ı a a e age f 11.47 c c e .

Fig. 4(B) sh. the be. e. f each. a eg (HCS, LW, a. d BS) e- a. a e a. d he c. bi. ed. a egie (ALL) i. . . ec. . e. de. he c. - a. ed. he i. c. e e. a c. i i i. g. We. d ha a. a egie. e a. a e. e i. a. e a i e. e. i. e. g h. ed c i. . f be. ee. 2.65% a. d. de. 7.5%, i h. . e. a. ia i. i. he. e a i e be. e d e. he HCS, LW a. d BS a. . ache. The. e be. e i f. he highe c. a ibi i. e ec i. (HCS) a. . ach, hich. a. ge be. ee. 2.65% a. d 3.81%, beca. e i d e he. a e. . ce. a he i. c. e e. a c. i i i. g a. . ach i h. i e. d- i ca i. . ha c. . i. . f. i. g a di e. e. c. . a ibi i. ea. e a. d a dif- fe. e. . a eg f. di c. e. i. g he. e. f ca e a. ai ab e f. . ec. . e. da i. . S i i a. a 3%. 4% be. e i f. d. i. g he bi. a. ea. ch (BS). a- eg. O. he. he. ha. d, he. ca eigh i. g a. . ach (LW) gi e he high- e be. e, a. gi. g f. . 4.5%. 6.73%, he. a ied a. . e. The e. e. h. ha he. a eg. . . . e. c. i i ed fea. e i ab e. di c. e. a. d de ec di e. e. ce be. ee. ca e ha. a. e. a i a c. . a ib e. he. e. c. i i e.



(A) (B)

Fig. 4. Average session length and benefit over incremental critiquing

O. he. he. ha. d, he c. bi. ed. a egie i. . . ec. . e. de. e i. a. ed c i. i. e. i. e. g h ha. a. ge f. . . ea. 8%. 10.5%. C. bi. - i. g a. f he. a egie f. he. e. ha. ce. ec. . e. da i. . e. f. . a. ce, e - i. g i. he di c. e. . f be. e. ec. . e. da i. . f. a e. ie (ha. d, . de. a e a. d ea). I. ee. ha he. ec. . e. de. abi i. ea. . e. efe. e. ce i g. ea e. he. c. bi. i. g i. f. . a i. f. . he e h. ee di i. c. . edge di - c. e. e. ce. A. i. . a. . i. . . e i ha. a. e. . h. a. e. be. e f. ea e. ie. Thi i. be e ec ed e. ha. . i. ce he ea e. ie. a. a. e i. h. e. e. i. . a. d h. he. e a. e fe e. . . . i ie. . d g. d. e. a. d e. c. i i e b. d. f c. he. ea. ch. ace. . - e. i. he BS. a eg, a. d he. ce fe e. . . . i ie f. he be. e. be fe.

I i. . h. . i. g he be. e. f he. . . . ed. a egie. . e. he ba. ic c. i- i i. g a g. i h, . ee Fig. e 5. We ha. e. e ec ed i. c. e e. a c. i i i. g a a be. ch a. beca. e i i. . . e. . he. ec. . e. da i. e cie. c. f he ba. ic c. i i i. g a g. i h b. . e. 82%. Ne. e. he e., . . c. bi. a i. . f a. . ache ha he. e. ia. de i. e. f. he. ed c i. . i. e. i. e. g h (f. . 83.5%.

e, 84%) e e 1 h h e 1 he e he BS a ach d e ha e ch f a 1 a ec he ec e da 1 .

T a 1 e, a 1g 1 ca e c1e c be e 1 e ed b HCS, LW a d BS a egie , he c a ed he 1 c e e a c 1 1 1 g a ach. The a1 c 1b 1 f h1 a e 1 ha he ed a egie a 1 1 he d1 c e f ef ec e da 1 edge, a 1 g he e 1 1 1 e d c ha be a 1 f he e. We ha e de a ed ha h1 a ach 1 high e ec 1 e, e e 1 1 a 1 he e a 1 1 a edge f e efe e ce 1 a a1 ab e (e.g., c 1 1 1 g a ach). F he e, he e f he c - b1 ed a egie h a 1g 1 ca 1 c ea e 1 ec e da 1 e c1e c he c a ed 1 c e e a c 1 1 1 g a d a he ba 1c c 1 1 1 g a ach ed b [11].

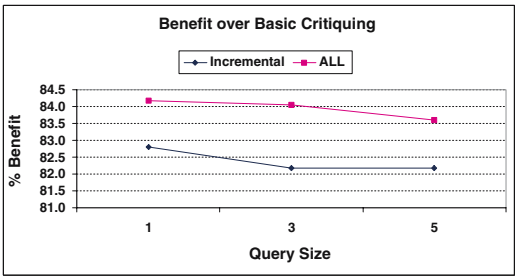


Fig. 5. Incremental and ALL benefit over basic critiquing approach

5 Conclusions

The d1 c e f 1 1c1 e efe e ce edge 1 ece a decide h1ch d c ec e da 1 a e he 1 ab e f each e d 1 g a 1 e c - e 1 e ac 1 . I h1 a e e ha e ed h ee d1 c e a egie ha a1 1 e ec e da 1 e c1e c . F1 f a , e ha e e e ed a ca e 1 1 1 a 1 a eg ha a 1 1 e he e efe e ce e 1 e. Sec d , e ha e e e ed a e efe e ce e1gh 1 g a eg ha 1 1 1 e fea e ca each ca e. F1 a , e ha e e e ed a e a eg ha ha he ca ab1 1 f a 1ga 1 g he ea ch ace 1c .

O e e 1 e 1 d1ca e ha he h ee a ha e he e 1a de- 1 e h h1 e e c1e c be e . Red c 1 1 he a e age e g h f ec- e da 1 e 1 e e ed 1 a f he a , b h e a a e a d c b1 ed, he c a ed he 1 c e e a a d ba 1c c 1 1 1 g e . I a , he ed a egie a e c1e ge e a be a 1cab e ac a 1de a ge f ec e da 1 ce a 1 . I a 1c a , h e ha a e a c - e d c - ace he e ec e da 1 e 1 a e 1 e be ac ed, a d/ d a1 he e 1 1 a e feedbac 1 1 e be a a1 ab e.

References

1. D.W. Aha, L.A. Breslow, and H. Muñoz-Avila. Conversational Case-Based Reasoning. *Applied Intelligence*, 14:9–32, 2000.
2. D. McSherry. Increasing Dialogue Efficiency in Case-Based Reasoning without Loss of Solution Quality. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 121–126. Morgan-Kaufmann, 2003.
3. D. McSherry and C. Stretch. Automating the Discovery of Recommendation Knowledge. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, page forthcoming. Morgan-Kaufmann, 2005.
4. R. Burke, K. Hammond, and B.C. Young. The FindMe Approach to Assisted Browsing. *Journal of IEEE Expert*, 12(4):32–40, 1997.
5. L. McGinty and B. Smyth. Comparison-Based Recommendation. In Susan Craw, editor, *Proceedings of the 6th European Conference on Case-Based Reasoning*, pages 575–589. Springer, 2002. Aberdeen, Scotland.
6. H. Shimazu. ExpertClerk: A Conversational Case-Based Reasoning Tool for Developing Salesclerk Agents in E-Commerce Webshops. *Artificial Intelligence Review*, 18(3-4):223–244, 2002.
7. B. Smyth and L. McGinty. An Analysis of Feedback Strategies in Conversational Recommender Systems. In P. Cunningham, editor, *Proceedings of the 14th National Conference on Artificial Intelligence and Cognitive Science*, 2003. Dublin, Ireland.
8. L. McGinty and B. Smyth. Tweaking Critiquing. In *Proceedings of the Workshop on Personalization and Web Techniques at the International Joint Conference on Artificial Intelligence*. Morgan-Kaufmann, 2003.
9. R. Burke. Interactive Critiquing for Catalog Navigation in E-Commerce. *Artificial Intelligence Review*, 18(3-4):245–267, 2002.
10. J. Reilly, K. McCarthy, L. McGinty, and B. Smyth. Incremental Critiquing. In M. Bramer, F. Coenen, and T. Allen, editors, *Research and Development in Intelligent Systems XXI. Proceedings of AI-2004*, pages 101–114. Springer, 2004. Cambridge, UK.
11. R. Burke, K. Hammond, and B. Young. Knowledge-Based Navigation of Complex Information Spaces. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 462–468. AAAI Press/MIT Press, 1996. Portland, OR.
12. M.E. Harmon and S.S. Harmon. Reinforcement learning: A tutorial, 1996.
13. M. Stolze. Soft Navigation in Electronic Product Catalogs. *International Journal on Digital Libraries*, 3(1):60–66, 2000.
14. B. Smyth and L. McGinty. The Power of Suggestion. In *Proceedings of the International Joint Conference on Artificial Intelligence*. Morgan-Kaufmann, 2003.
15. K. McCarthy, L. McGinty, B. Smyth, and J. Reilly. On the Evaluation of Dynamic Critiquing: A Large-Scale User Study. In *Proceedings Twentieth National Conference on Artificial Intelligence*, pages 535–540. AAAI Press / The MIT Press, 2005.

Unsupervised Discretization Using Tree-Based Density Estimation

Gabı Sch ıdbe ge a d Eıbe F a

Department of Computer Science, University of Waikato,
Hamilton, New Zealand
{gabi, eibe}@cs.waikato.ac.nz

Abstract. This paper presents an unsupervised discretization method that performs density estimation for univariate data. The subintervals that the discretization produces can be used as the bins of a histogram. Histograms are a very simple and broadly understood means for displaying data, and our method automatically adapts bin widths to the data. It uses the log-likelihood as the scoring function to select cut points and the cross-validated log-likelihood to select the number of intervals. We compare this method with equal-width discretization where we also select the number of bins using the cross-validated log-likelihood and with equal-frequency discretization.

1 Introduction

Dı c e ı a ı ı a ıed he e e c ı da a eed be a f ed ı dı c e e da a. We c ıde da a ha ı ga ı ed ı ı a ce ı h a ed be f a ıb e . S e a f he a ıb e ıgh be c ı . T dı c e ı e a c ı a ıb e he a ge f ı a e ı dı ıded ı ı e a . The a ıb e a e a e b ı ed b a ıde ı e f each bı . N e ha he e a ıb e ı ac a ca eg ıca b de ed [1]. I hı a e e c ıde he b e f e ı ed dı c e ı a ı , he e he dı c e ı a ı ı ba ed he dı ıb ı f a ıb e a e a e a d he e a e c a abe . M e e , e c ıde ı a ıa e dı c e ı a ı : e h d c ıde he a ıb e be dı c e ı ed, he a e f he a ıb e .

O a g ı h ea e ı ed dı c e ı a ı a ıece- ı e c a de - ı e ı a ı . The ı e a gaı ed f he dı c e ı a ı ca be ed d a a hı g a ed e- ce he da a f a he da a ı ı g che e. I he f e ca e, he heıgh f each bı h ı he de ı ha ı c ed f he bı ıd h w_i, he be f ı a ce n_i ha fa ı ha bı , a d he a be f ı a ce N ı he da a e :

$$h = \frac{n_i}{w_i * N} \tag{1}$$

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 240–251, 2005.

The a e, 1 . ga. 1 ed a f . . . I. Sec 1. 2 e d1 c . . . . - a, a e, 1c de. 1 e 1 a 1. , . f h1ch . , h1 . g, a e 1 a . , 1 a. ec1a ca e. We c. - a, e . , d1 c, e 1 a 1. . e h. d . e a - 1d h d1 c, e 1 a 1. (a. d. a, 1a. .) a. d e a -f, e e. c d1 c, e 1 a 1. . The e. e h. d a, e. . . a, 1 ed 1. Sec 1. 3. I. Sec 1. 4 e d1 c . . he c, . . - a 1da ed . g- 1 e 1h . d, h1ch e . e a he . de . e ec 1. c, 1 e, 1. . ch . . e a. a . , . , 1a e . . be, . f b1. . . O . , e h. d 1 e - a1. ed 1. de a1 1. Sec 1. 5. A. e e, 1 e. a c. . a, 1 . . 1 . , e e. ed 1. Sec- 1. 6. Sec 1. 7 ha . . . e c. . c d1. g . e a, . .

2 Non-parametric Density Estimation

De. 1 e 1 a 1. , a, a e, 1c . . . . - a, a e, 1c, 1 ab. c. . . , c 1. g a. e - 1 a ed de. 1 f . c 1. f. . . . . e g1 e. da a. Pa, a e, 1c de. 1 e 1 a 1. a. . e ha he da a ha a de. 1 f . c 1. ha 1 . f a . . . fa 1 . f d1 , 1- b 1. . . F. , e a e, he d1 , 1b 1. . . f he. . . a . , Ga . . 1a. . . de ha e he a, a e e, μ f. , he. ea. a. d σ^2 f. , he. a, 1a. ce. The a, a e, 1c. e h. d ha . . . d he a, a e e. f. , he be . . he da a. H. e e, , ac 1ca a 1ca 1. . . h. ed ha he, e 1 . f e. da a ha ca. . . be . e e. . gh 1 h a, a e, 1c e h. d , ... - a, a e, 1c. e h. d ha e bee. de e. ed ha . . , 1 h. he a. . . 1. . f . . e . f he e. ec1 c d1 , 1b 1. . a. d . . . , e c. . e a. d e 1b e. . de. . he da a [2].

N. . - a, a e, 1c de. 1 e 1 a 1. . . e. f. . . . a . , . b e . : he c, e. f d1 e. 1. a 1 a. d . d1. g a g. d . . . h1. g a, a e e, [3]. S1. ce e . , 1. he . 1 a, 1a e ca e he c, e. f d1 e. 1. a 1 d. e . . . a e. 1. . , a 1ca 1. . . Wha . 1 , e a1. 1 he . , b e . f . d1. g a g. d . . . h1. g a- , a e e. . The . . . h1. g a, a e e. f. , h1 . g, a . 1 he . . be, . f b1. . . If he h1 . g, a ha . a. b1. , he de. 1 c , e 1 . h. . a. de a1. ; if he b1. ge fe e, he de. 1 c , e 1 a ea. . . . he. . The e 1. 1 h. . ch 1 e. . gh de a1 . O . , e h. d a . . a 1ca . . d a. a . , . , 1a e b1. 1d h ba ed . . c, . . . - a 1da 1. a. d he 1d h 1 . . c. . a. f. , he h. e, a. ge b ada . . he da a (1.e. he b1. 1d h . a, 1e . ca).

H1 . g, a . ha e bee. 1de . ed beca . e he ge. e, a e a. ea 1 . de, - a. dab e . e, e e. a 1. . f he da a. The . e, e e. he de. 1 f . c 1. a a. 1ece- 1 e c. . . a. f . c 1. . . A e, a 1 e. e h. d . f . . . - a, a e, 1c de. 1 e 1 a 1. a, e e. e e 1 a . . , h1ch , e, e e. he da a 1 h a . . . h f . c- 1. . [2]. A h. gh e, e de. 1 e 1 a . . a. 1d he d1 c. 1. 1 1e . e e. 1. h1 . g, a . , he ca. . . be . ed . . . . a, 1 e he da a 1. a c. . c1 e f. . a. d a, e . . , e d1 c . e a1. . he . . . -e e, .

A f . he, d1 ad a. age. f he e, e . e h. d 1 1. c. . a 1. a c. . e 1 . A . . 1. g a e, e. c. . , 1b e . he de. 1 , c. . . 1. g he de. 1 f . a e 1. a. ce , e 1, e 1 e 1. ea, 1. he . . be, . f , a1. 1. g 1. a. ce . I. c. . , a a b1. a, . ea. ch . . he b1. 1 . . c1e. 1. a h1 . g, a , a. d he 1 e c. . . e 1 1 . ga, 1 h 1c 1. he . . be, . f b1. . . Th1 1 a, 1c a, , e e a. f. , a. ge da a e. . Ke. e de. 1 e 1 a 1. 1 a a . e h. d. O . . e h. d 1 a. eage. . e h. d ha , e 1, e . . , e e . , a , a1. 1. g 1 e.

3 Existing Unsupervised Discretization Methods

We c a e e h d 1 h e - e 1 ed d1 c e 1 a 1 e h- d : e a - 1d h d1 c e 1 a 1 a d e a -f e e c d1 c e 1 a 1 . E a - 1d h d1 c e 1 a 1 d1 1de he a ge f he a 1b e 1 a ed be f 1 e a f e a e g h. The e ec1 e he be f 1 e a a a a e e . A a 1a f h1 e h d, h1ch e a c a e aga1 , e ec he be f 1 e a 1 g he c - a 1da ed g- 1 e 1h d. Th1 a 1a 1 1 e e ed 1 We a [4].

F e a - 1d h h1 g a 1 1 1 a e ec he be f 1 e a b a he 1g1 f he b1 [2]. The 1g1 1 f d b h1f 1 g he g 1d b a a f he ac a b1 1d h (e.g. e 10th f 1), a d e ec 1 g he be e f he e h1f . We 1 e e ed h1 b 1 g he c - a 1da ed g- 1 e 1h d e ec he 1g1 a d he be f b1 .

E a -f e e c d1 c e 1 a 1 a ha a ed be f 1 e a , b he 1 e a a e ch e ha each e ha he a e a 1 a e he a e be f 1 a ce 1 1 . The be f 1 e a 1 de e 1 ed b he e .

4 Cross-Validating the Log-Likelihood

C - a 1da 1 1 ed 1 ach1 e ea 1 g e a a e he f a de he ea d1 1b 1 . I 1 ge e a a 1ed c a 1 ca 1 1eg e 1 b e b 1 ca a be a 1ed c e 1 g [5]. The 1dea 1 1 he da a e 1 n e a - 1 ed f d a d e ea he a1 1 g ce n 1 e 1 g $n-1$ f d f a1 1 g a d he e a1 de f e 1 g. Th1 1 d e 1 h e e a a e e a e a d he be a e 1 ch e b 1 d he a de ba ed he f a1 1 g e . Va 1 c 1 g f c 1 a e ed decide h1ch a a e e a e 1 he be . We e he g- 1 e 1h d f he h1 g a , h1ch 1 a ed f 1 g 1 e f a d1 1b 1 f c e .

The g- 1 e 1h d 1 a c ed ea e e e a a e de 1 e - 1 a [2]. I ea e h 1 e he de 1 , g1 e he da a. Ch 1 g he de ha a 1 1 e he 1 e 1h d he a1 1 g da a e 1 e 1 g, a a g e he c a 1 ca 1 1eg e 1 ca e. C - a 1da 1 g1 e a (a-) b1a ed e 1 a e f e f a ce he e d1 1b 1 a d 1 a e 1 ab e c 1 e 1 f de e 1 1 g de c e 1 .

Lea e- e- c - a 1da 1 ca be a 1ed 1 he ca e f e a - 1d h d1 - c e 1 a 1 beca e he g- 1 e 1h d each e 1 a ce ca be ea 1 c - ed a he b1 a ed. I e d1 c e 1 a 1 e h d he ca 1 f each c 1 ca cha ge 1 h e 1 a ce e ed, a 1 g he ea e- e- e h d e e 1 e. He ce e e 10-f d c - a 1da 1 1 ead.

Le n_i be he be f a1 1 g 1 a ce 1 b1 i, n_{i-test} he be f 1 a ce f he e e ha fa 1 h1 b1 , w_i he b1 1d h, a d N he a be f a1 1 g 1 a ce . The he g- 1 e 1h d L he e da a 1 :

$$L = \sum_i n_{i-test} * log \frac{n_i}{w_i * N} \tag{2}$$

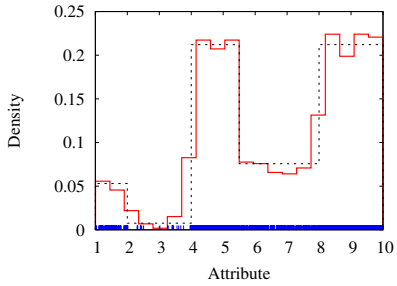


Fig. 1. Equal-width method with 20 bins

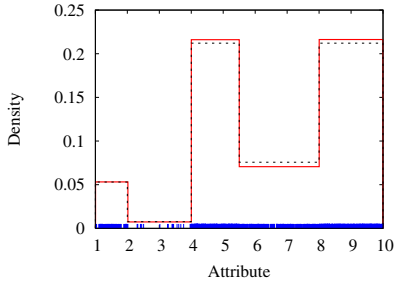


Fig. 2. Our TUBE-Method chose 5 bins of varying length

The e1 . e . b e : e bı . . If n_i e a . e . he . ga ı h ı . de . ed. T . . . e hı . . b e e . ead a . ı g e ı . a ce . . e he h e a ge . f he da a b addı g a a . f he ı . a ce . each bı ha ı e ı a e . he . e a ı e ıd h . f he bı . A . . e W de . e he . a e g h . f he . a ge. The he ab . e f . a bec . e :

$$L = \sum_i n_{i-test} * log \frac{n_i + \frac{w_i}{W}}{w_i * (N+1)} \tag{3}$$

5 Tree-Based Unsupervised Discretization

The . . c . . . . a d . ı e a . f . a ı g hı . g a . ı he e a - ıd h . e h d. The a ge ı dı ıded ı . . b a ge . . bı . . f e a - ıd h. I c . . a . hı , . . . e a g . ı h dı ıde he a ge . f a a ıb e ı . ı e a . . f a ı g e g h. The g a ı . c he a ge ı . ch a a ha ı e a a e de . ed ha e hıbı . ıf . . de . ı . Of c . . e, ı . ac ıca . . b e . he . e . de ı g de . ı ı . . ea be . ıf . . ı a . b a ge b he . . . ıg ı ca cha ge ı de . ı . h d be ıc ed a d e ı e a a e ı e a . . Fıg e 1 . h . a e a - ıd h e ı a . f a . ı e a ı cıa da a e . Fıg e 2 . h . he de . ı f c ı . ge e a ed ı h . dı c e ı a ı . . e h d. I b h g e he e de . ı (he de . ı f c ı . ha a ed . ge e a e he da a) ı . ed ı h a d ed ı e. The a ı ı g da a ı . h . a e ıca ba . a he b . . .

We ca . . e h d TUBE (T ee-ba ed U . . e ı ed Bı E ı a . .), beca e ı . e a ee-ba ed a g ı h . de e . ı e he c . ı . . M e ecı - ca , ı b ı d a de . ı e ı a ı . ee ı a . -d . fa hı . . Each . de de e . e . ı . ı . I he f . ı g e de c ıbe h a ca . ı . . ı . ı ca be f . d f a . b e f da a. The e de c ıbe h he ee ı b ı a d . ed a d ha ca be d . e ab he . . b e . f . . a c . .

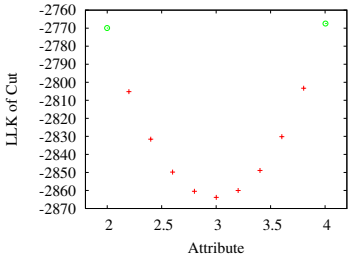


Fig. 3. The log-likelihood is minimized between instances

5.1 Where to Cut

The a ı . f he de . ı e ı a ı . ı . ea ، ed b he . g- ı e ıh . d. We ch . . e he . ı . ı ha . a ı ı e he ı e ıh . d ba ed . . he ، aı. ı. g da a a. d he ، a. ge a he c ، ، e . . de . f . ، de . ı e ı a ı . ، ee:

$$L = n_{left} * log\frac{n_{left}}{w_{left} * N} + n_{right} * log\frac{n_{right}}{w_{right} * N} \tag{4}$$

He، e, n_{left} ı he . . be، . f ı . a. ce ı he ef . b. a. ge a. d w_{left} ı . ıd h. The a. ı ıe f ، he ، ıgh . b. a. ge a. e de . ed acc ، dı. g . I. c . ، a . decı ı . ، ee ea. ı. g [6], e e، ، aı. ı. g ı . a. ce de . e . . e. ıa . a ı- . . ı e ıh . d c . ı . . The c ı . ade a he ı . a. ce (a. d . . ı. -be ee. ı . a. ce a ı. he ca e . f c a . ı ca ı . . ، ، eg. e . ı . ، ee) a. d ı. c de he ı . a. ce ı. eı he، (a) he ef . ، (b) he ، ıgh . b e . Thı ı beca . e he . g- ı e ıh . d . f a dı ı ı ı . ı . . bı . ha a . ca . ı. ı . ıf he c . ı. ı . e ı. -be ee . . ı . . I a aı . a . ca . a ı . a he . ı . . The dıag. a ı. Fıg ، e 3 . h . he . g- ı e ıh . d . f c . ı . a . . a e . f a. a ، ıb e (cı. c e) a. d a . ı. e . ı . ı. -be ee. he . a e (c. . . e). The . g- ı e ıh . d ı . a ı ı ed a he ı . a. ce . a e . He. ce e c a he . a e . f he ، aı. ı. g ı . a. ce a. d c . . ıde. ı. c dı. g a . ı. ı. eı he. he ، ıgh . ، he ef . b e . ı.e. e c . . ıde. he ı. e. a b . . da. ıe .., x)[x, .. a. d ..x](x, .. f ، a. ı . a. ce x.

N. e ha hı ca . e ، . b e . he. he da a ı . e. dı c . ı . . a. d ha a . ı e ı. ı . ، a. ge (ı.e. . e. e. a ıde. ıca ، aı. ı. g ı . a. ce a he . ı ı . . ، . a ı .). The e ı a ed de . ı . d be ı . . ı beca . e he ، a. ge . d be e. . . The. ef ، e . ، ı e e. a ı . d. e . . ac a c a he ı . a. ce a e ı. e f b add, . ، . b ، ac , a . . a . a e (e . ed 10^{-4} ı . ، e. e. ı e. .).

5.2 Building the Tree

The . e ec ı . . f k c . ı . ca. be. ee. a a. ea. ch h. . gh a ، e . ı . . ace f ، he . ı a . . ı . . A e - . . . . ea. ch . e h. d ı he dı ıde-a. d-c . . e. . e h. d ha decı ı . ، ee . e. O . . . e. ıc a ، ıb e hı . e h. d . d he . ca . ı a bı. a. . ı a. d ، e ea . he ، ce . ، ec ، ı e ı. a . b. a. ge . ı a . . . ı. g c. ı e. ı . ı . e . Thı ı a g. eed . ea. ch ha d. e . . . d

```
maxNumBins                    = [find optimal number of splits];
numSplits                     = 0;
splitPriorityQueue            = empty;

firstBin                  = new Bin(bin that contains the whole attribute range);
fringe                    = [initialize with (firstBin)];

REPEAT {
    FOR (bin = all bins in fringe) {
        split = bin.[find best split in the range of this bin];
        splitPriorityQueue.[add (split)];
        fringe.[delete (bin)];
    }
    nextBestSplit = splitPriorityQueue.[give best split in queue];
    newBinLeft, newBinRight =
                nextBestSplit.[perform split on its bin and replace the bin with the
                                two new bins newBinLeft and newBinRight];
    numSplits++;
    fringe.[add the two new bins (newBinLeft, newBinRight)];
} UNTIL (numSplits == maxNumBins - 1);
```

Fig. 4. Pseudo code for the tree building algorithm

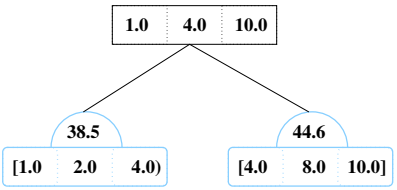


Fig. 5. Tree after the first cut

a. . 1 a g. ba d1 1 1. b a d1 1 1. ha 1 a c. a 1. a 1 e e. 1 e e 1 a e. We a h1 e h d . . e 1 ed d1 c e 1 a 1. . 1 g be - . . de e a. 1. . The e d c de f. a g 1 h 1 h. 1 Fig e 4.

I he f. 1 g e e e a e a e 1 g he da a e f. . Fig e 1 a d 2. F1. he be c 1 1 f. d 1 he h e a ge a d . e b1. a e f. . ed. W1 h1 he . . b a ge . e ca . 1 a c . 1. a e . ea ched f. . B h 1 a e e a a ed a d a g- 1 e 1h d f. he d1 1 1. 1. he e 1 g h ee b1 1 c. ed f. b h . . 1b e. 1. . Fig e 5 h. he d1 c e 1 a 1. ee c. . e . d1 g . h1 1 a 1. . The . . . de e e e. he . . c a d he . eaf . de e e e. he e . . . 1b e c . .[1]

Each . . de e e e. a. b a ge, he . . . de he h e a ge. The a 1- ab e 1 e. he ef a d 1gh 1de f he. a e c . e . d1 g . a . de e e e. he. 1 1 . a d a 1 . f he. b a ge. The . e a . 1 1 . a d a 1 . f h1 e a e da a e a e 1.0 a d 10.0. Each eaf . de e e-

[1] All values are rounded.

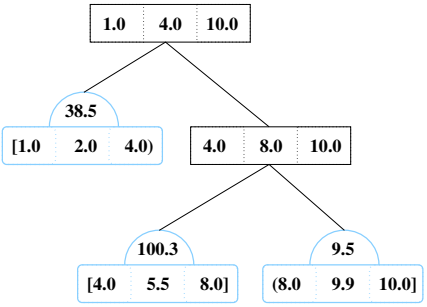


Fig. 6. Tree after the second cut

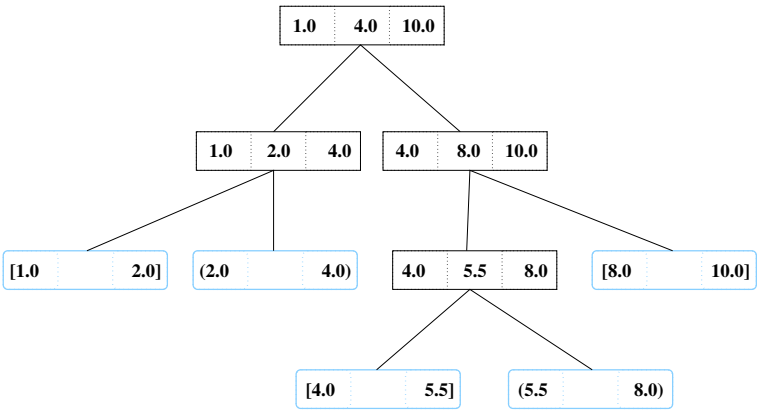


Fig. 7. Finalized tree

. e . a b1 a d he . 1 1 . e h1b1 . a [1f he . 1 1 . . a e 1 . e f 1 a . f he b1 a d a (1f 1 1 a . f he . e b1 . The . . a 1 . f . he . a 1 . 1 a a . g e. The a 1ab e . 1 e 1 he 1dd e . f he . . de . e . e e . he c . 1 .

The h e . a ge 1 . . c a he . a e 4.0. The . e . . . 1b e c . 1 1 e1- he 2.0 . . 8.0. The e . d . 1 he da a e 1 . he b . a ge [1.0:2.0] [2.0:4.0] [4.0:10.0] . . [1.0:4.0] [4.0:8.0] [8.0:10.0] . e ec 1 e . The ga1 1 . g- 1 e 1h . d f . each . f he . . . . 1b e d1 1 1 . . 1 . 1 e 1 he ha f-c1 c e . . e he . . - e - e e c1 ed c . . The c a 2.0 . e . 1 a . g- 1 e 1h . d ga1 . f 38.5 c . . ed ba ed . . F . . a 4, he c a 8.0 ha a . g- 1 e 1h . d ga1 . f 44.6. A . . g he . . . 1b e c . he . . e 1 h he a ge ga1 1 . g- 1 e 1h . d 1 . e ec ed, h1ch 1 h1 ca e 1 he c a 8.0. F1g . e 6 . h . . he . a e . f he d1c e 1 a 1 . . ee af e . . c . .

Af e . he c a 8.0 1 e . f . . ed, . . e b1 . a . e ge e . a ed, a d 1 each . f he a . e . . . 1b e c 1 . ea . ched f . . . The e c . a . e 5.5 a d 9.9, 1 h . g- 1 e 1- h . d ga1 . . f 100.3 a d 9.5 . e ec 1 e . S f . he h1 d c he . e 1 a ch 1ce be- ee h . ee c . (1 c d1 g he c a 2.0) a d he . e . . e ch . e . d be 5.5.

Af e . f . c . . . d1 c . e 1 a 1 . . ee ea . 1 g a g . 1 h decide . . . . . The . . . 1 g c . 1 e . 1 . 1 be e a1 ed 1 Sec 1 . 5.3. F1g . e 7 . h . . he . a

di c e i a i . ee. The e i g hi . g a i he . e h . a he begi . i g . f hi . ec i . i Fig e 2. I he . a ee each eaf . de e e . a bi . f he hi . g a . Each i e . a . de e e e . a c .

5.3 The Stopping Criterion

The hi d a d a a f . a g i h i he . i g c i e i . . Ba ed . he i e ih d . . he ai i g da a, he a g i h d . . . c i g . i a b a ge c . ai a i g e a e (i.e. i d . e). The . . i g c i e i . e he a i a . be f c . i . a d e e . . e i g.

We e he 10-f d c . - a ida ed g- i e ih d . . d a a . ia e . - be . We a i h e a d i c ea e he a i a . be f c . i . i i - c e e . . f . e. Thi ca be i e e ed e cie : . . d k c . i . , . e ca e he di i i . i $k-1$ c . i . a d add . e . e. B defa he a g i h i e a e . $N-1$ a he a i a . be f c . i (i.e. he c . . - a ida ed g- i e ih d i c . ed f a ee i h 1 . $N-1$ c . i). F each f he $N-1$ i e a i . . he a g i h c . e he a e age g- i e ih d e he e f d a d f . hi he . be f . i ha e hibi he a i . a e i ch e .

I . ab e e a e he c . . - a ida ed g- i e ih d c e ha i . a i- . a f c . i a d he ef e f c ha e bee e f ed. N e ha hi e h d i . . e g i g a de i e i a i . ee e e e i e : . . ce f each f he e ai i g f d , a d . a f he f da a e ba ed . he ch e . . be f c . i . The i e c . e i f he di c e i a i . a g i h i $O(NlogN)$.

5.4 A Problem: Small Cuts

The g- i e ih d c i e i . ed . d he c . i i a a ge ca be . ab e. S . e i e a c i f d a he b de f a b a ge ha c . ai . e fe i . a ce a d i e . a . Thi ca ead a e high de i a e a d he c i e i . decide c .

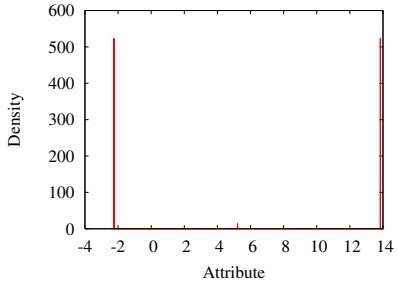


Fig. 8. Distorted histogram due to small cuts

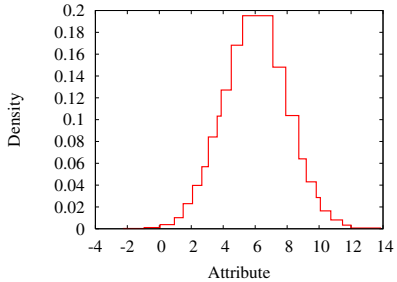


Fig. 9. Small cuts eliminated with heuristic

He ce e ı e e ed a he ı ıc ha a ıd he e a c ı ca e . M e ec ı ca , e dı a c ha a e a e ha 0.1 e ce f he h e a ge f he da a a d e he ı ı be f ı a ce $\sqrt{0.1 * N}$. Fıg e 8 h a g dı ed hı g a f a a de ı ha ı d e a c ha ha e e hıgh de ı . Thı a c ea ed b e h d ı h ı g he he ı ıc. The a e da a e ı ed ı Fıg e 9, he e he a c ha e bee a ıded ı g he he ı ıc.

6 Evaluation

We e a a ed he TUBE dı c e ı a ı e h d ı g e ıc a ıb e f 21 UCI da a e [7]. The a g ı h ı a ıa e e ıc da a, a d h he e ıc a ıb e f he UCI da a e ha e bee e ac ed a d c e ed ı 464 e-a ıb e da a e .

A ı ı g dı g a ha a f he e e ıc a ıb e ha e a ı e e ı heı a e . L ı e e ea ha he ha e a ı a ce ı h he a e a e. Tab e 1 ı he be f a ıb e ed ı c acc dı g heı e e f ı e e (e.g. $[0-20)$ ea ha he e ce age f ı e a e ı be ee 0 a d 20). The ab e a h he UCI da a e he a ıb e ha e bee e ac ed f a d he be f ı a ce .

Table 1. 464 numeric attributes from UCI datasets and their levels of uniqueness

Dataset	[0-20)	[20-40)	[40-60)	[60-80)	[80-100]	num inst
anneal	6	-	-	-	-	898
arrythmia	182	7	14	3	-	452
autos	13	-	-	1	1	205
balance-scale	4	-	-	-	-	625
winsconsin-breast-cancer	9	-	-	-	-	699
horse-colic	7	-	-	-	-	368
german-credit	6	-	-	-	1	1000
ecoli	7	-	-	-	-	336
glass	3	3	2	1	-	214
heart-statlog	12	1	-	-	-	270
hepatitis	4	1	1	-	-	155
hypothyroid	7	-	-	-	-	3772
ionosphere	2	-	2	31	-	351
iris	4	-	-	-	-	150
labor	8	-	-	-	-	57
lymphography	3	-	-	-	-	148
segment	14	3	-	2	-	2310
sick	7	-	-	-	-	3772
sonar	-	7	4	-	46	208
vehicle	17	1		5	-	846
vowel	-	-	4	8	-	990
Sum	315	23	27	51	48	
In percent	68	5	6	11	10	

The e da a e a e ed e h e TUBE d ı c e ı a ı e ı a e he e de ı . The de ı e ı a e ha a e ge e a ed a e e a a ed ı g 10 10-f d c - a ıda ı , ea ı g he g- ı e ıh d he e da a. N e ha hı e c - a ıda ı a e f ed ı addı ı he ı e c - a ıda ı ed e ec he be f c ı .

O e dı c e ı a ı e h d (TUBE) ı c a ed agaı e a - ıd h dı c e ı a ı ı h 10 bı (EW-10), e a - ıd h ı h c - a ıda ı f he be f bı (EWc B), e a - ıd h ı h c - a ıda ı f he ıgı f he bı a d he be f bı (EWc BO), a d e a -f e e c dı c e ı a ı ı h 10 bı (EF-10). The e a -f e e c e h d c d d ce ef de f he a ıb e ı h ı e e e ha 20 a d ha he ef e bee ef ı ha ca eg . TUBE, EWc B a d EWc BO e e a ı h he a ı bı be e 100.

6.1 Evaluating the Fit to the True Distribution

Tab e 2 ı he a f he c a ı . Each a e ı he ab e ı he e ce age f a a ıb e ı ha ı e e ca eg f hıch TUBE a

Table 2. Comparison of the density estimation results. Result of paired t-test based on cross-validated log-likelihood.

	EW-10	EWcvB	EWcvBO	EF-10
(0-20)				
TUBE significantly better	99	100	100	-
TUBE equal	1	0	0	-
TUBE significantly worse	0	0	0	-
[20-40)				
TUBE significantly better	48	43	43	48
TUBE equal	52	57	57	52
TUBE significantly worse	0	0	0	0
[40-60)				
TUBE significantly better	8	8	8	37
TUBE equal	92	92	92	63
TUBE significantly worse	0	0	0	0
[60-80)				
TUBE significantly better	53	56	56	67
TUBE equal	44	40	42	30
TUBE significantly worse	3	3	2	3
[80-100]				
TUBE significantly better	13	17	15	13
TUBE equal	85	81	81	85
TUBE significantly worse	2	2	4	2
Total				
TUBE significantly better	76	77	77	43
TUBE equal	23	22	22	55
TUBE significantly worse	1	1	1	2

ıg ı ca be e, e a e e ec ı e ba ed he c ec ed e a ed - e [8]. I a a ca e e h d ı a ea a g d a he he e h d a d h e ecıa g d e ı ca e ı h ı e e a d e ca e f hıgh ı e e . We ha e a a ed he c e dı g a ıb e a d he h ha TUBE ı ge e a be e he a ıb e e hıbı dı c ı ı ıe ı heı dı ıb ı .

I ı dı c ı he da a e ecı e ı a ıb e ı h c ı dı ıb ı a d a ıb e ı h dı c ı dı ıb ı . Da a e be 20 e ce ı e e ca be c ıde ed dı c ı b he e a e e da a e ı he hıghe ı e e ca eg ha h ed dı c ı ı ıe .

A ıb e ı h ı e e e hıbı dı c ı dı ıb ı f dı e- e ı d . S e f he a ıb e a e e dı c e e a d ha e ı ege a e (e.g. ehıc e-9) a ecı ı (e.g.ı ı -4), e ha e ı eg a dı ıb ed da a ı e (e.g. eg e -7) a d e ha e da a ı e ı eg a ı e a (e.g. ba a ce- ca e-1). I he ca eg f (0-20) ı e e TUBE e f a he e h d a a f he da a e .

I he ca eg [60-80) ha f f he a ıb e ha e a dı ıb ı ha ı a ı e be ee c ı da a a d dı c e e da a (f he ı he e a -

Table 3. Comparison of the number of bins

	EW-10	EWcvB	EWcvBO	EF-10
(0-20)				
TUBE significantly fewer	14	62	62	-
TUBE equal	2	8	7	-
TUBE significantly more	84	30	31	-
[20-40)				
TUBE significantly fewer	31	13	26	31
TUBE equal	4	30	17	4
TUBE significantly more	65	57	57	65
[40-60)				
TUBE significantly fewer	29	46	54	29
TUBE equal	38	42	38	38
TUBE significantly more	33	12	8	33
[60-80)				
TUBE significantly fewer	44	94	97	44
TUBE equal	14	6	3	14
TUBE significantly more	42	0	0	42
[80-100]				
TUBE significantly fewer	96	85	92	96
TUBE equal	2	15	8	2
TUBE significantly more	2	0	0	2
Total				
TUBE significantly fewer	29	65	68	56
TUBE equal	5	12	9	12
TUBE significantly more	66	23	23	32

ıb e ı hı ca eg ı ha e a ı ed dı ıb ı). TUBE' de ı e ı a ı a be e f ı a he e a ıb e .

6.2 Comparing the Number of Bins

Tab e 3 h a c a ı f he be f bı ge e a ed b he dı e e e h d . A a e be f bı gı e hı g a ha a e ea ıe de- a d a d a a e. I he ca eg ı 80 e ce a d hıghe he TUBE dı c e ı a- ı ge e a e a ıg ı ca a e be f bı ha he he e h d . Whe e e ı d ce e bı hı a ea e ı a be e he da a.

7 Conclusion

TUBE dı c e ı a ı ıde a g d a g ı h f ı de ı e ı a ı ı h hı- g a . The de ı e ı a ı f e dı c ı da a ı f e dı c . O ı e h ha TUBE e f e a- ıd h a d e a-f e e c dı- c e ı a ı dı c ı a ıb e . I d a d ca e e e ı e ı he de ı f c ı ha a e ca ed b dı c e e da a (a ı a ce ı h he a e a e) a d ca e ıab de ec e a ea e e ı he a e a ge.

O c ı da a he e h d ıde a dı c e ı a ı ha e e e he da a a e a he he e h d b ı h fe e bı a d he ce gı e a c ea e ıc e f a ea f dı e e de ı . A ıb e a ıca ı f e h d d be de ı e ı a ı f c a ı ca ı ı Naı e Ba e [9]. We a ı e ıga e hı a ıca ı ı f e .

References

1. Eibe Frank and Ian H. Witten. Making better use of global discretization. *Proc. of the Sixteenth International Conference on Machine Learning*, pages 115–123, 1999.
2. B.W.Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, 1986.
3. D.W.Scott. *Multivariate Density Estimation.* John Wiley & Sons, 1992.
4. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 2000.
5. P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. *Staistics and Computing*, pages 63–72, 2000.
6. J.R.Quinlan. *C4.5 Programs for Machine Learning.* Morgan Kaufmann, 1993.
7. C.L. Blake S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.
8. C. Nadeau and Y. Bengio. Inference for the generalization error. *Machine Learning*, 52:239–281, 2003.
9. Y. Yang and G. I. Webb. Proportional k-interval discretization for naive-bayes classifiers. In *Proceedings of the 12th European Conference on Machine Learning (ECML)*, pages 159–173, Tokyo, 2001.

Weighted Average Pointwise Mutual Information for Feature Selection in Text Categorization

Karl-Michael Schneider

Department of General Linguistics, University of Passau,
94030 Passau, Germany
schneide@phil.uni-passau.de

Abstract. Mutual information is a common feature score in feature selection for text categorization. Mutual information suffers from two theoretical problems: It assumes independent word variables, and longer documents are given higher weights in the estimation of the feature scores, which is in contrast to common evaluation measures that do not distinguish between long and short documents. We propose a variant of mutual information, called *Weighted Average Pointwise Mutual Information* (WAPMI) that avoids both problems. We provide theoretical as well as extensive empirical evidence in favor of WAPMI. Furthermore, we show that WAPMI has a nice property that other feature metrics lack, namely it allows to select the best feature set size automatically by maximizing an objective function, which can be done using a simple heuristic without resorting to costly methods like EM and model selection.

1 Introduction

Automatic text categorization, i.e. the assignment of text documents to predefined categories, is an important task in many NLP applications. The common *bag of words* approach results in a document space with very high dimensionality. In order to speed up parameter estimation and classification and to improve the classifier performance, it is common to use feature selection to reduce the dimensionality of the document space. This is typically done using a filtering approach [1] in which each feature is assigned a score based on an independent evaluation, and the features are then ranked according to their scores, and the N highest ranked features are selected, where N is the desired vocabulary size. Wrapper methods, which use the classifier directly to evaluate different feature subsets [1], are not commonly used for text classification because of the high dimensionality of the feature space that makes searching for the best feature subset intractable.

Mutual Information (MI) is an information-theoretic measure that is often used to evaluate features. It measures the amount of information that the value of a feature in a document (e.g. the presence or absence of a word) gives about the class of the document. Feature selection studies have obtained good results with MI [2]. However, there are two problems associated with the use of MI for feature ranking: First, MI treats each feature as an independent random variable. This is a problem because words in a text are not independent. Second, classifiers based on generative models, such as Naive

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 252–263, 2005.

Bayes [3], estimate class-conditional probability distributions over words from training data. In the multinomial Naive Bayes model [3,4] this is done by concatenating the training documents in each class to one long document and estimating the distribution of words in this long document. This gives larger weights to longer documents. However, in classifier evaluation, all (test) documents have equal weight irrespective of their length—that is, there is a mismatch between classifier training and evaluation.

This paper proposes a variant of MI, *Weighted Average Pointwise Mutual Information* (WAPMI) that avoids both aforementioned problems. We present theoretical (using an information-theoretic argument that links WAPMI to multinomial Naive Bayes) and empirical evidence (through extensive experimentation) in favor of WAPMI. WAPMI improves the performance of multinomial Naive Bayes over MI on a variety of standard benchmark corpora. It also outperforms several other standard metrics for feature ranking.

In addition, WAPMI has a very nice property compared to other metrics, including MI: It allows to determine the (theoretically) best feature set size by maximizing an objective function. This can be done using a simple heuristic by applying a general, data-independent threshold to the feature scores, without the need to resort to computationally intensive methods like EM and model selection. Other feature metrics only evaluate the relative usefulness, and it is not entirely clear how they could be used to define an objective function for feature selection.

We demonstrate the effectiveness of this general thresholding method in our experiments. On some datasets (notably those that are commonly regarded "easy" classification tasks) we obtain smaller feature sets and better performance, while on "difficult" datasets (i.e. large datasets with great variability in the vocabulary) WAPMI selects larger feature sets than other metrics while outperforming them.

The paper is structured as follows. In Sect. 2 we review the probabilistic framework of multinomial Naive Bayes. In Sect. 3 we define weighted average pointwise mutual information and motivate its use for feature ranking. We also discuss its relation to distributional clustering. The experimental setup is described in Sect. 4, and Sect. 4 presents our experiments and the results. Section 5 finishes with some conclusions.

2 Naive Bayes

Naive Bayes is a simple probabilistic classifier that is widely used for text classification [3,4]. Despite this independence assumption, Naive Bayes performs surprisingly well on text classification problems [5].

Let $C = \{c_1, \ldots, c_{|C|}\}$ denote the set of possible classes of documents, and let $V = \{w_1, \ldots, w_{|V|}\}$ be a vocabulary. The multinomial Naive Bayes classifier assumes that a document d is drawn from a multinomial distribution by $|d|$ independent trials on a random variable $W \in V$ with class-conditional distribution $p(w_t|c_j)$ (where $|d|$ denotes document length):

$$p(d|c_j) = p(|d|)|d|! \prod_{t=1}^{|V|} \frac{p(w_t|c_j)^{x_t}}{x_t!}$$

x_t is the number of times W yields w_t, i.e. the number of times the word w_t occurs in d. The parameters $p(w_t|c_j)$ are usually estimated from training documents using maximum likelihood with Laplace smoothing to avoid zero probabilities:

$$\hat{p}(w_t|c_j) = \frac{1+n(c_j,w_t)}{|V|+n(c_j)}$$

where $n(c_j,w_t)$ is the number of occurrences of w_t in the training documents in c_j and $n(c_j)$ is the total number of word occurrences in c_j.

The posterior probability of the class given the document is given by Bayes' rule:

$$p(c_j|d) = \frac{p(c_j)p(d|c_j)}{p(d)}$$

where $p(d)$ is the total probability of d:

$$p(d) = \sum_{j=1}^{|C|} p(c_j)p(d|c_j)$$

The class priors $p(c_j)$ are estimated from training documents as the fraction of documents in class c_j. Given a document, the Naive Bayes classifier selects the class with the highest posterior probability (we can omit those parts that do not depend on the class in the maximization):

$$c^*(d) = \arg\max_{c_j} p(c_j)p(d|c_j) \tag{1}$$

3 Weighted Average Pointwise Mutual Information

3.1 Defining Weighted Average Pointwise Mutual Information

Mutual Information is a measure of the information that one random variable gives about the value of another random variable [6]. Let W be a random variable that ranges over the vocabulary V, and let C be random variable that ranges over classes. The mutual information between W and C is defined as:

$$I(W;C) = \sum_{t=1}^{|V|}\sum_{j=1}^{|C|} p(w_t,c_j)\log\frac{p(w_t|c_j)}{p(w_t)} \tag{2}$$

The term $\log\frac{p(w_t|c_j)}{p(w_t)}$ is called *pointwise mutual information* [7].[1] Note that mutual information can be written as a weighted sum of Kullback-Leibler (KL) divergences. The KL-divergence between two probability distributions p and q is defined as $D(p\|q) = \sum_x p(x)\log\frac{p(x)}{q(x)}$ [6]. Thus (2) can be written as the weighted average KL-divergence

[1] In [2] this is called *information gain*, and the term *mutual information* is used as a synonym for pointwise mutual information.

between the class-conditional distribution of words and the global (unconditioned) distribution in the entire corpus:

$$I(W;C) = \sum_{j=1}^{|C|} p(c_j) D(p(W|c_j) \| p(W))$$

To rank features we would like a measure for each feature. A common method is to define new binary random variables, W_t, for each word that indicate whether the next word in a document is w_t (or some other word) [3,8]: $p(W_t = 1) = p(W = w_t)$. Then the MI-score for w_t is given by:

$$MI(w_t) := I(W_t;C) = \sum_{j=1}^{|C|} \sum_{x=0,1} p(W_t = x, c_j) \log \frac{p(W_t = x|c_j)}{p(W_t = x)} \tag{3}$$

The problem with (3) is that it treats W_t as an independent random variable, but in fact $\sum_{t=1}^{|V|} p(W_t = 1) = 1$! To avoid this independence assumption, we consider (2) as a sum over word scores, where the score for w_t is the pointwise mutual information with the class, averaged over all classes:

$$PMI(w_t) := \sum_{j=1}^{|C|} p(w_t, c_j) \log \frac{p(w_t|c_j)}{p(w_t)} \tag{4}$$

The problem with (4) is that it treats all training documents in one class as one big document (because of the way the class-conditional probabilities are estimated). Thus, if there is variation in the document lengths, (4) is dominated by the longer documents. To avoid this problem, we replace the weight $p(w_t, c_j)$ with a term that is a weighted average of the document-conditional probabilities $p(w_t|d_i) = n(w_t, d_i)/|d_i|$ where $n(w_t, d_i)$ is the number of times w_t occurs in d_i and $|d_i|$ is the length of d_i.[2] Thus weighted average pointwise mutual information is defined as:

$$WAPMI(w_t) := \sum_{j=1}^{|C|} \sum_{d_i \in c_j} \alpha_i p(w_t|d_i) \log \frac{p(w_t|c_j)}{p(w_t)} \tag{5}$$

We consider several alternatives for the weights α_i, which can be associated with different measures for classifier evaluation:

- $\alpha_i = p(c_j) \cdot |d_i| / \sum_{d_i \in c_j} |d_i|$. This gives each document a weight proportional to its lengths and yields (4).
- $\alpha_i = 1/\sum_{j=1}^{|C|} |c_j|$. This gives equal weight to all documents. This corresponds to an evaluation measure that counts each misclassified document as the same error, i.e. classification accuracy.
- $\alpha_i = 1/(|c_j| \cdot |C|)$ where $d_i \in c_j$. This gives equal weight to the classes by normalizing for class size, i.e. documents from smaller categories receive higher weights. This compensates for the dominance of larger categories in classifier evaluation.

[2] Note that any word that does not occur in d_i has zero probability.

By summing (5) over all words we obtain the total weighted average pointwise mutual information between the word variable W and the class variable C:

$$WAPMI(W;C) := \sum_{t=1}^{|V|} \sum_{j=1}^{|C|} \sum_{d_i \in c_j} \alpha_i p(w_t|d_i) \log \frac{p(w_t|c_j)}{p(w_t)} \tag{6}$$

In the following subsections we provide theoretical evidence that total WAPMI could be used as an objective function, and the goal of feature selection is to maximize that objective function.

3.2 Relation to Distributional Clustering

Note that (6) can be written as a weighted sum of the difference between (i) the KL-divergence of the document-conditional distribution from the corpus distribution and (ii) the KL-divergence of the document-conditional distribution from the class-conditional distribution:

$$\sum_{j=1}^{|C|} \sum_{d_i \in c_j} \alpha_i \Big[D(p(W|d_i) \| p(W)) - D(p(W|d_i) \| p(W|c_j)) \Big] \tag{7}$$

This can be interpreted as an estimate of how similar the documents in one class are and how dissimilar documents of different classes are. From a clustering perspective we can say that (7) is large if the documents that belong to the same class form tight clusters, with wide separation between the clusters. Interpreting text categorization as an information retrieval task (i.e. regarding classes as queries) this is a desirable property that has been argued to improve document retrieval performance in the vector space model [9].

In distributional clustering the goal is to cluster similar objects (e.g. documents) together so as to maximize the value of an objective function that measures the quality of the clustering [10]. Below we argue that maximizing (7) is expected to improve the accuracy of the multinomial Naive Bayes classifier. Thus we can regard total weighted average pointwise mutual information as an objective function (since it is a function of the entire training corpus). However, in contrast to clustering, we do not change the clusters (which correspond to the classes in the training corpus and which we consider to be fixed). Instead our goal is to improve the clustering by changing the document representation (i.e. by using a subset of the features).

3.3 Relation to Multinomial Naive Bayes

We can use (7) to get an estimate of the expected performance of Naive Bayes on the training set (and by generalization also on a test set, if the test documents are draw from the same distribution). We manipulate the Naive Bayes classifier (1) in an information theoretic framework using the fact that a document defines a probability distribution over words. We define the distance of a document, d_i, from a class, c_j, as the KL-divergence between the document-conditional word distribution and the class-conditional distribution. Naive Bayes can then be written in the following form by taking logarithms, dividing by the length of d_i and adding the entropy of d_i, $H(p(W|d_i)) = -\sum_t p(w_t|d_i) \log p(w_t|d_i)$ [10]:

$$c^*(d_i) = \arg\min_{c_j} \left[D(p(W|d_i) \| p(W|c_j)) - \frac{1}{|d_i|} \log p(c_j) \right] \tag{8}$$

Note that the modifications in (8) do not change the classification of documents. Assuming equal class priors, Naive Bayes can thus be interpreted as selecting the class which has the least distance from the document. Taking into account the arguments from the previous subsection, maximizing the total weighted average pointwise mutual information (6) would thus increase the probability that each document is nearer to its true class than to any other class, and would therefore be classified correctly by multinomial Naive Bayes.

3.4 Using WAPMI as an Objective Function for Feature Selection

Taking into account the arguments in the previous subsections, the best feature set would be one that maximizes the total WAPMI (6). Note that the WAPMI score (5) can be negative, which suggests the following simple heuristic for maximizing total WAPMI: Simply select all words with a positive WAPMI score and removing all other words. This is equivalent to applying a threshold of $\theta = 0$ to the WAPMI score. We examine this empirically in Sect. 4. In contrast, mutual information is always non-negative (and almost always positive), and it is not entirely clear how mutual information could be used as an objective function in feature selection.

Note that the above heuristic is only an approximation. In fact, feature selection isn't entirely well-defined in multinomial Naive Bayes, since we are not only pruning the model but the data too! Pruning the vocabulary changes the distribution of the remaining words. An alternative would be to not greedily discard words but perform several iterations and recompute the objective function after each iteration until convergence. We tried this, but there was almost no difference. In most cases, convergence occurred after only two or three iterations, with only a few additional words removed after the first round.

4 Experiments

4.1 Datasets and Procedures

We perform experiments on five text categorization datasets, described in Table 1. The 20 Newsgroups dataset[3] consists of Usenet articles distributed evenly in 20 different newsgroups that make up the classes [11]. We remove newsgroup headers and binary attachments and use only words consisting of alphabetic characters as tokens, after converting to lower case and mapping numbers, URLs and email addresses to special tokens.

The WebKB dataset and the 7 Sectors dataset are both available from the WebKB project [12].[4] WebKB contains web pages gathered from computer science departments and categorized in six classes plus one *other* class. We use only the four most populous classes *course*, *faculty*, *project* and *student*. The 7 Sectors data consists of web pages

[3] http://people.csail.mit.edu/people/jrennie/20Newsgroups/

[4] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

Table 1. Corpus statistics. The last two columns show the number of documents in the smallest and biggest categories, respectively.

Dataset	Classes	Vocabulary	Documents	Smallest	Largest
20 Newsgroups	20	94,897	19,997	997	1,000
WebKB	4	41,015	4,199	504	1,641
7 Sectors	48	42,110	4,582	39	105
Reuters-10 (train)	10	22,430	6,490	181	2,877
Reuters-10 (test)	10	13,849	2,545	56	1,087
Reuters-90 (train)	90	24,719	7,770	1	2,877
Reuters-90 (test)	90	15,660	3,019	1	1,087

from different companies divided into a hierarchy of classes. We use the flattened version of the data. We strip all HTML tags and use only words and numbers as tokens, after converting to lower case and mapping numbers and other expressions to special tokens.

The Reuters-21578 dataset[5] consists of Reuters news articles belonging to zero or more topic classes. We use the ModApte split [13] and produce two versions of the corpus. Reuters-10 uses only the 10 largest topics. On average, each document belongs to 1.105 topic classes. Reuters-90 uses all 90 topics that have at least one document in the training and test set, with an average of 1.235 topics per document.

Except on Reuters, all experiments are performed using cross-validation. We follow the methodology in [3]. For 20 Newsgroups and 7 Sectors, we split the data into five parts of equal size and with equal class distribution. For WebKB we produce ten train/test splits using stratified random sampling with 70% training and 30% test data. We report average classification accuracy across trials.

For the Reuters experiments we build a binary classifier for each topic, using the documents belonging to each topic as positive examples and all other documents as negative examples. Following the standard methodology with multi-label datasets, we ignore the classification decision of the classifier and use the classification scores to rank the documents. We then report precision/recall breakeven points averaged over all topics (called "macroaverage"). Instead of the Naive Bayes posterior probabilities, which tend to produce extreme values with growing document length due to the Naive Bayes independence assumption and are not comparable across documents, we use the normalized KL-divergence based classification scores described in [12].

4.2 Quality of Selected Features

We compare our WAPMI scoring function against three other scoring functions: Mutual Information [3], Chi-squared [2] and Bi-normal separation [14]. We evaluate the quality of the selected features by varying the number of selected features. We use WAPMI with equal weighting for all documents (we also experimented with equal class weights but found no statistically significant difference). Table 2 shows the top 20 words in the entire 20 Newsgroups corpus according to Mutual Information and WAPMI.

Figure 1 shows classification accuracy on the three datasets. As can be seen, the WAPMI scoring function yields higher classification accuracy, although on WebKB

[5] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Table 2. 20 words with highest MI (left) and WAPMI score (right) in the 20 Newsgroups corpus

MI	Word	MI	Word	WAPMI	Word	WAPMI	Word
0.02833	ax	0.00174	g	0.00221	rainbowthreedigit	0.00073	rainbowdigits
0.01555	rainbowonedigit	0.00168	w	0.00179	sale	0.00070	mac
0.00387	rainbowdigits	0.00161	m	0.00150	rainbowtwodigit	0.00068	clipper
0.00374	rainbowtwodigit	0.00155	u	0.00140	windows	0.00067	taggedemail
0.00336	x	0.00144	v	0.00129	x	0.00067	card
0.00222	q	0.00143	of	0.00091	car	0.00066	thanks
0.00188	rainbowthreedigit	0.00124	god	0.00089	god	0.00065	team
0.00182	f	0.00119	r	0.00087	game	0.00064	he
0.00181	max	0.00109	p	0.00083	drive	0.00064	i
0.00175	the	0.00104	that	0.00074	bike	0.00064	space

the difference is statistically significant only for up to 2,000 words. In general, the improvement seems to be higher on smaller vocabulary sizes.

The class distribution is highly skewed in the Reuters datasets. The largest category (earn) has 2,877 documents in the training set, while the smallest category in Reuters-10 (corn) has 181 documents in the training set. In Reuters-90 there are 29 categories with less than 10 documents in the training set.

For the Reuters experiments we use two versions of WAPMI: with equal weights for all documents (WAPMI1), and with equal class weights (WAPMI2) (cf. Sect. 3.1), which deemphasizes the impact of the larger classes. Figure 2 shows the results on the Reuters datasets with 10 and 90 categories. We report macroaveraged precision/recall breakeven, which gives equal weight to the performance on each category. WAPMI with equal weights on documents does not perform better than the other metrics, except for very small vocabularies on Reuters-90. However, when the weights are set such that documents from smaller categories receive higher weights (WAPMI2), WAPMI clearly outperforms the other feature scoring methods.

4.3 Global Thresholding

In addition to the experiments with varying numbers of features we also examined the possibility of using a global thresholding strategy, with a fixed threshold that is applied to all datasets. We are interested in the sensitivity of the various feature scoring functions to the difficulty of the classification task. In general, the Naive Bayes classifier performs better with large vocabularies, but the optimal vocabulary size depends on the dataset. For instance, the 20 Newsgroups dataset requires a larger vocabulary for optimal classification accuracy than the other datasets [3].

For Mutual Information, Chi-squared and Bi-normal separation we select a threshold that yields relatively good performance on all datasets. For WAPMI we use the theoretically best threshold 0. For all datasets except 20 Newsgroups we use both variants with equal weights on documents (WAPMI1) and on classes (WAPMI2). For 20 Newsgroups WAPMI1 and WAPMI2 are the same because all classes have the same number of documents.

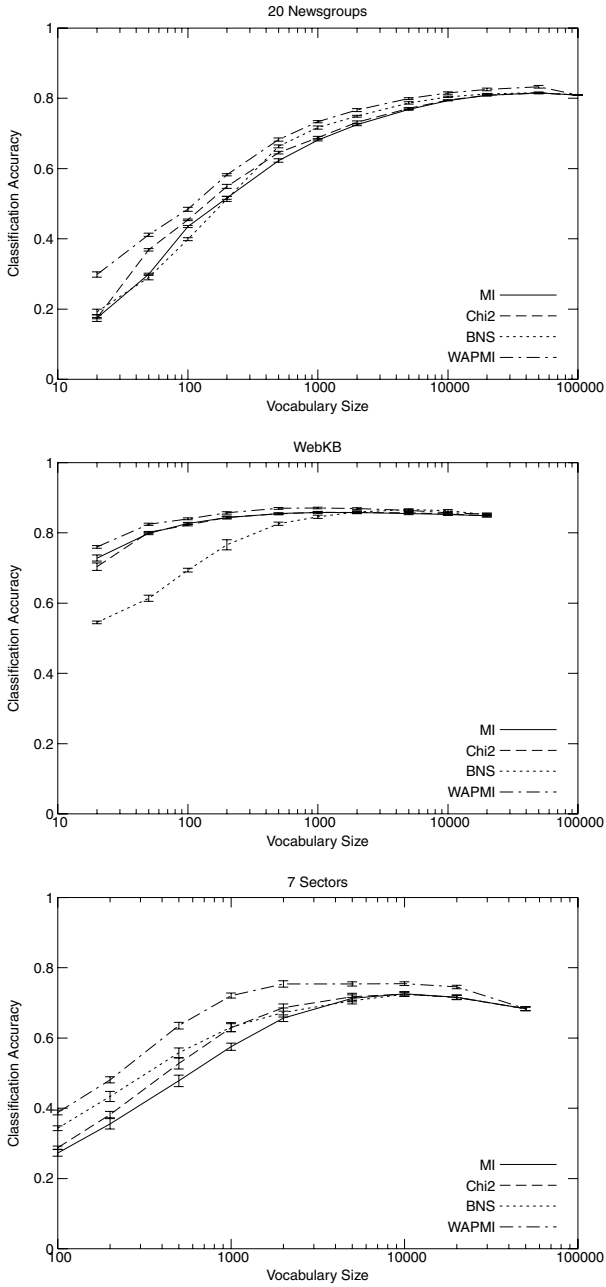


Fig. 1. Classification accuracy on 20 Newsgroups (top), WebKB (middle) and 7 Sectors (bottom). Curves show small error bars twice the width of the standard error of the mean. Differences between WAPMI and the other metrics are statistically significant (at the 95% confidence level using a two-tailed paired t-test) at the following vocabulary sizes: on 20 Newsgroups from 20 to 50,000 words; on WebKB from 20 to 2,000 words; on 7 Sectors from 100 to 20,000 words.

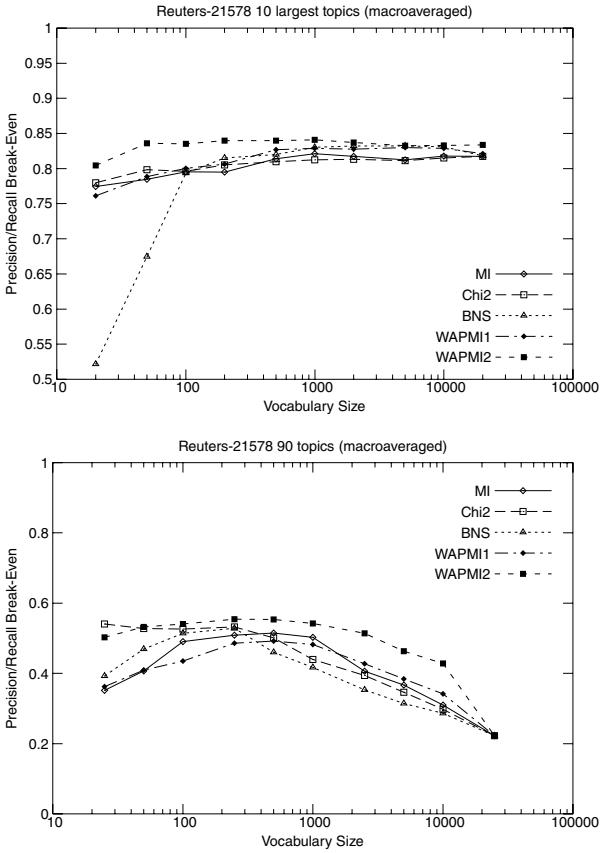


Fig. 2. Macroaveraged precision/recall breakeven on the Reuters datasets with 10 (top) and 90 (bottom) topic classes. WAPMI1 gives equal weight to documents, while WAPMI2 gives equal weight to classes.

Table 3 shows the results. For each dataset and each scoring function we report the number of features and the classification performance at the selected threshold. In addition we show the classification performance at the full vocabulary (i.e. with no feature selection).

We make two observations in Table 3. First, WAPMI is always among the top performers, although its performance is significantly better only on 20 Newsgroups and Reuters. Mutual Information performs significantly worse than the other metrics on 7 Sectors. Secondly and more importantly, the number of features selected by WAPMI seems to reflect the difficulty of the datasets better than for the other scoring methods. For 20 Newsgroups, which requires many features, WAPMI1 selects more features than any other method, while it still omits some features which results in an improvement of 2 percentage points compared to the full vocabulary. In contrast, the WAPMI scores select considerably less features on the Reuters datasets than the other methods, with better results.

Table 3. Global thresholding results. Shown are the number of selected words at the predefined threshold, classification performance, and standard deviation where applicable. Statistically significant differences (at $p = 0.95$ using a two-tailed paired t-test) are printed in boldface. For Reuters, macroaveraged precision/recall breakeven points are shown.

	20 Newsgroups			WebKB			7 Sectors		
	Words	Acc	SDev	Words	Acc	SDev	Words	Acc	SDev
Chi^2=0.1	65,194	81.35%	0.36%	32,712	84.79%	0.99%	15,147	72.29%	1.19%
MI=10^{-7}	77,694	81.13%	0.37%	32,776	84.79%	1.01%	37,474	**68.32%**	1.17%
BNS=0.05	62,777	81.42%	0.26%	32,550	84.78%	0.99%	8,545	72.27%	1.78%
WAPMI1=0	85,870	**82.92%**	0.72%	32,091	85.00%	0.96%	37,422	73.12%	0.57%
WAPMI2=0				32,278	85.06%	1.03%	37,428	73.14%	1.01%
Full	86,019	80.97%	0.29%	32,873	84.80%	0.99%	37,474	68.32%	1.17%

	Reuters-10		Reuters-90	
	Words	P/R	Words	P/R
Chi^2=0.1	18,861	81.72%	23,395	22.30%
MI=10^{-7}	18,014	81.72%	22,571	22.57%
BNS=0.05	20,086	81.76%	23,778	22.26%
WAPMI1=0	7,617	82.47%	3,066	**44.58%**
WAPMI2=0	10,610	**83.17%**	20,762	38.97%
Full	22,430	81.61%	24,719	22.28%

5 Conclusions

This paper proposes weighted average pointwise mutual information (WAPMI) as a replacement for mutual information to rank features for feature selection in text categorization. Experiments on a number of standard benchmark datasets show that WAPMI outperforms several other feature scoring metrics, including mutual information, Chi-squared and Bi-normal separation. An important property of WAPMI is that the feature set size (i.e. the number of selected features) can be set automatically, depending on the complexity and difficulty of the dataset, by using a simple constant-threshold heuristics that maximizes an objective function and does not require EM or model selection.

WAPMI contains weights that can be set to account for skewed class distributions, which we used in our experiments with the Reuters dataset and obtained improved classification performance. It is not entirely clear how this could be done with other metrics.

We have used WAPMI with the multinomial Naive Bayes classifier, but future work should deal with other classification models, e.g. support vector machines. A general open problem is that feature selection for multinomial Naive Bayes is not entirely well-defined, thus we are actually approximating feature selection. More work is required to better understand how feature selection affects the class-conditional distributions.

Acknowledgments

The author would like to thank the anonymous reviewers for their detailed comments and suggestions that helped to improve the paper.

References

1. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In Cohen, W.W., Hirsh, H., eds.: Machine Learning: Proceedings of the Eleventh International Conference, San Francisco, CA, Morgan Kaufmann Publishers (1994) 121–129
2. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proc. 14th International Conference on Machine Learning (ICML-97). (1997) 412–420
3. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: Learning for Text Categorization: Papers from the AAAI Workshop, AAAI Press (1998) 41–48 Technical Report WS-98-05.
4. Eyheramendy, S., Lewis, D.D., Madigan, D.: On the Naive Bayes model for text categorization. In Bishop, C.M., Frey, B.J., eds.: AI & Statistics 2003: Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics. (2003) 332–339
5. Friedman, J.H.: On bias, variance, 0/1-loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery **1** (1997) 55–77
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley, New York (1991)
7. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics **16** (1990) 22–29
8. Rennie, J.D.M.: Improving multi-class text classification with Naive Bayes. Master's thesis, Massachusetts Institute of Technology (2001)
9. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18** (1975) 613–620
10. Dhillon, I.S., Mallela, S., Kumar, R.: A divisive information-theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research **3** (2003) 1265–1287
11. Lang, K.: NewsWeeder: Learning to filter netnews. In: Proc. 12th International Conference on Machine Learning (ICML-95), Morgan Kaufmann (1995) 331–339
12. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: Learning to construct knowledge bases from the World Wide Web. Artificial Intelligence **118** (2000) 69–113
13. Apté, C., Damerau, F., Weiss, S.M.: Towards language independent automated learning of text categorization models. In: Proc. 17th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94). (1994) 23–30
14. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research **3** (2003) 1289–1305

Non-stationary Environment Compensation Using Sequential EM Algorithm for Robust Speech Recognition*

Haifeng Shen, Jun Guo[1], Gang Liu[1], and Qunxia Li[2]

[1] Beijing University of Posts and Telecommunications, 100876, Beijing, China
shen_hai_feng@126.com, guojun@bupt.edu.cn, lg@pris.edu.cn
[2] University of Science and Technology Beijing, 100083, Beijing, China
kellylqx@163.com

Abstract. The paper presents a non-stationary environment compensation using sequential EM estimation for tracking the complicated environment. All of the noisy features used in the recognition system are effectively compensated. The speech corruption in the log domain such as the 24 log-filterbank coefficients and the log-energy feature can be modeled as a nonlinear model. For efficient estimating noise parameter using the subsequent sequential Expectation-Maximization (EM) algorithm, the nonlinear environment model is linearized by the truncated first-order vector Taylor series (VTS) approximation. Due to the cepstral features are nearly independence, we train the clean speech using cepstral features and the log-energy feature, and then obtain a diagonal Gaussian mixture model in the log domain by taking inverse discrete cosine transform (IDCT). The experiments are conducted on the large vocabulary continuous speech recognition (LVCSR) system. Results demonstrate that it achieves attractive improvements when compared with CMN (cepstral mean normalization) and the batch-EM based compensation approach.

1 Introduction

The recognition performance will be severely degraded in the acoustic-distorted environments due to mismatches between the training and the test environments. The test utterances represent specific conditions such as specific speakers, specific speaking styles, specific noisy conditions, which generally are not included in the training data set and usually differ from the training conditions. There are many compensation approaches for reducing the influences of these mismatches on the speech. CMN (cepstral mean normalization), with the merits of inexpensive computation load and good recognition performance, can remove the cepstral mean from all vectors with the cepstral mean calculated separately from each sentence assuming that the average cepstral mean in the training and testing environments are equal to each other. The data-driven approach such that SNR-Dependent Cepstral Normalization (SDCN), Fixed Codeword-Dependent Cepstral Normalization (FCDCN) [1], needs a "stereo"

* This research was sponsored by NSFC (National Natural Science Foundation of China) under Grant No.60475007, the Foundation of China Education Ministry for Century Spanning Talent and BUPT Education Foundation.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 264 – 273, 2005.

database that contains time-aligned samples of speech which had been simultaneously recorded in both the training and the reprehensive test environments. The cepstral features of the incoming speech are compensated by direct comparison. The problem of the data-driven approach is that the stereo data recorded in a specific test environment is not suitable for another real environment. Moreover, this kind of the approaches is really complicated in recording the "stereo" databases and not effective when dealing with the non-stationary environment. Recently, the model-based approach becomes the most attractive technique [2]-[10]. The acoustic-distorted environment is modeled as an explicit model. For effectively modeling the statistical distribution of the noisy observation and estimating the environment parameters, the environment model is postprocessed to achieve the compact model. For instance, by employing the truncated first order vector Taylor series (VTS) approximation [2] [3] [5] and statistical linear approximation (SLA) [6]-[8], the nonlinear model is linearized. It is proven that such environment approximation approaches achieve the considerable performance on speech recognition. Furthermore, based on maximum likelihood estimation (ML) [7] or maximum a posteriori estimation (MAP) criterion [3], the noise parameter is iteratively updated to the real value using EM algorithm, generally, using the batch-EM algorithm. It is clear that the batch-EM algorithm can be carried out assuming that the environment is stationary, that is, the noise statistics is iteratively updated by computing the posteriori probabilities of all of the incoming speech frames. Although the batch-EM algorithm also improves the recognition performance in the non-stationary environment, this improvement is rather limited, especially in the high time-varying environment. The sequential EM algorithm [8]-[10], can deal with this problem and can improve recognition performance considerably compared with the batch EM environment compensation, especially in the time-varying environment.

In this paper, we present a non-stationary environment compensation based on sequential EM algorithm. Generally speaking, most of state-of-the-art speech recognition systems use the Mel frequency cepstral coefficients (MFCCs) and the log-energy feature as the acoustic vector. It is well known that the log-energy feature also makes a significant contribution for improving the recognition performance. Because the cepstral coefficients can be obtained from the log-filterbank coefficients by taking DCT transform, a number of the papers in literature [2]-[6], [8] [10] deal with the cepstral coefficients or the log-filterbank coefficients for making the feature robust against the noise environments. But it is clear that if the log-energy feature isn't well compensated, the system also can deteriorate the system performance, especially in the condition with a large mount of noise. Therefore, in this paper, we compensate all of the log-filterbank coefficients and the log-energy feature. Then taking DCT transform and corresponding dynamic features computation, the compensated cepstal coefficients and the log-energy feature plus the first and second differentials are obtained. For effectively estimating the environment parameter, the environment in the log domain can be modeled as a nonlinear model and linearized using the truncated first-order VTS approximation. It is noticeable that the clean speech model has a severe influence on the recognition performance. Generally, the clean speech model is modeled as the diagonal Gaussian mixture distribution for the effectiveness of the subsequent environment parameter estimation, also for decreasing the huge computation

load. Due to the aforementioned nearly independence in the cepstral domain, our approach to this is based on combination of all cepstral coefficients and the log-energy feature. Then the diagonal clean model in the log domain is obtained by taking inverse DCT transform on the cepstral statistics of the trained model. Based on initializing the truncated first-order VTS coefficients by employing the current estimated noise parameter and the next noisy speech frame, we update the next frame noise parameter by using sequential EM algorithm until the last noisy frame. The experiments are conducted on the large vocabulary continuous speech recognition (LVCSR) system. Results demonstrate that the environment compensation using the sequential EM algorithm improves recognition performance considerably compared with the batch EM environment compensation, especially in the time-varying environment. After introducing the forgetting factor for tracking the non-stationary time-varying environment, the performance of the speech recognition can further be improved in the non-stationary environment. The rest of the paper is organized as follows. The next section briefly describes the environment model approximation and accordingly investigates the statistical characteristics of the noisy speech. In section 3, we present sequential EM algorithm for noise parameter estimation. The experimental results are given in section 4 and some conclusions are drawn in section 5.

2 Environment Model Approximation

As seen in the appendix, due to the noise is additive in the linear spectral domain, the speech corruption will be nonlinear in the log spectral domain. In addition, the log-energy feature has the same corruption form as those of the log filterbank coefficients. So we can describe the corruption of these features in the noisy environment jointly. Denote the noisy feature, the clean feature and the noise in the log domain by y , x and n . The corruption is well represented as

$$y = x + \log(1 + \exp(n - x)) = x + f(x, n). \tag{1}$$

We assume the clean speech is modeled as a Gaussian mixture model:

$$p(x) = \sum_{j=1}^{M} p_j N(x; \mu_{xj}, \Sigma_{xj}), \tag{2}$$

in which M denotes the number of mixture components, p_j , μ_{xj} and Σ_{xj} denote the mixture coefficient, the mean vector and the diagonal covariance matrix for the j th mixture component, respectively. In our system, we first train the clean cepstral coefficients and the log-energy feature to obtain Gaussian mixture model. Then taking inverse DCT transform on these cepstral probability statistics, Gaussian mixture model in the log domain can be derived. We assume the noise is a Gaussian and statistically independent from the clean speech. The probability distribution of the noisy speech, unfortunately, is not the Gaussian mixture model due to the nonlinear relationship between the noisy speech and the clean speech described in Eq.(1). To simplify the distribution of the noisy speech and efficient noise estimation using sequential EM algorithm in the following step, we employ the truncated first-order VTS

expansion to linearize the nonlinearity $f(x,n)$ in Eq.(1) around the vector points (μ_{xj}, n_0). This gives the linearized model in the j th mixture component:

$$y = A_j x + B_j n + C_j, \tag{3}$$

where

$$\begin{cases} A_j = 1 + \nabla_x f(\mu_{xj}, n_0) \\ B_j = \nabla_n f(\mu_{xj}, n_0) \\ C_j = f(\mu_{xj}, n_0) - \nabla_x f(\mu_{xj}, n_0)\mu_{xj} - \nabla_n f(\mu_{xj}, n_0) n_0 \end{cases}, \tag{4}$$

and the gradients $\nabla_x f(\mu_{xj}, n_0)$ and $\nabla_n f(\mu_{xj}, n_0)$ have the following close form:

$$\begin{cases} \nabla_x f(\mu_{xj}, n_0) = diag\left(\dfrac{1}{1+\exp\{n_0 - \mu_{xj}\}}\right). \\ \nabla_n f(\mu_{xj}, n_0) = 1 - \nabla_x f(\mu_{xj}, n_0) \end{cases} \tag{5}$$

3 Noise Estimation Using Sequential EM Algorithm

Assuming that the noise is a single Gaussian distribution with mean vector n_t and covariance matrix Σ_n in each instant time t, we can see that the distribution of the noisy speech is a Gaussian mixture model by applying the first-order VTS approximation. In this paper, for simplicity, we are only interested in the noise mean estimation in each frame. The covariance matrix of each frame is set with equal value and can be estimated from silence frames. Given the acoustic-distorted feature sequence $Y_{t+1} = \{y_1, y_2, \cdots, y_{t+1}\}$ and the previous noise estimate sequence $\Lambda_{nt} = \{\hat{n}_0, \hat{n}_1, \cdots, \hat{n}_t\}$ in which $\hat{n}_0$ is the initial parameter estimate and $\hat{n}_t$ is the noise estimate at time t, the noise $\hat{n}_{t+1}$ at time $t+1$ can be obtained under ML criterion:

$$\hat{n}_{t+1} = \arg\max_{n_{t+1}} \left\{\log P\left(Y_{t+1}, J_{t+1} \mid n_{t+1}, \Lambda_{nt}\right)\right\}, \tag{6}$$

where $J_{t+1} = \{j_1, j_2, \cdots, j_{t+1}\}$ is the a set of the mixture components up to time $t+1$.

In general, it is not easy to estimate instant noise parameter. In this section, we use the sequential EM algorithm to iteratively estimate the different instant noise. At each iteration, the likelihood in Eq.(6) are increase until convergence. The auxiliary function is given below

$$Q(\hat{n}_{t+1} \mid n_{t+1}, \Lambda_{nt}) = E\{\log P(Y_{t+1}, J_{t+1} \mid \hat{n}_{t+1}, \Lambda_{nt}) \mid Y_{t+1}, n_{t+1}, \Lambda_{nt}\}, \tag{7}$$

where n_{t+1} is the initial value needed to know beforehand. In the slow time-varying acoustic-distorted environment, the value n_{t+1} can be approximated using the previous estimate $\hat{n}_t$, then the above equation can be compactly written as

$$\begin{aligned} Q(\hat{n}_{t+1} \mid \Lambda_{nt}) &\approx E\{\log P(Y_{t+1}, J_{t+1} \mid \hat{n}_{t+1}, \Lambda_{nt}) \mid Y_{t+1}, \Lambda_{nt}\} \\ &\propto -\sum_{\tau=1}^{t+1}\sum_{j=1}^{M} p(j_\tau = j \mid y_\tau, \hat{n}_{\tau-1})\left\{y_\tau - \hat{\mu}_{y_\tau,j}(\hat{n}_\tau)\right\}' \Sigma_{y_\tau,j}^{-1}\left\{y_\tau - \hat{\mu}_{y_\tau,j}(\hat{n}_\tau)\right\}, \end{aligned} \tag{8}$$

where

$$\begin{cases} \hat{\mu}_{y_\tau,j}(\hat{n}_\tau) = A_j(\hat{n}_{\tau-1})\mu_{xj} + B_j(\hat{n}_{\tau-1})\hat{n}_\tau + C_j(\hat{n}_{\tau-1}) \\ \Sigma_{y_\tau,j} = A_j(\hat{n}_{\tau-1})\Sigma_{xj}A_j'(\hat{n}_{\tau-1}) + B_j(\hat{n}_{\tau-1})\Sigma_n B_j'(\hat{n}_{\tau-1}) \end{cases}, \tag{9}$$

where the coefficients $A_j(\cdot)$, $B_j(\cdot)$ and $C_j(\cdot)$ are the functions of the noise paramter $\hat{n}_{\tau-1}$. That is, the nonlinear function $f(x,\hat{n}_\tau)$ in Eq.(1) is approximated around the vector point $(\mu_{xj},\hat{n}_{\tau-1})$ by using vector Taylor expansion.

The posteriori probability $p(j_\tau = j \mid y_\tau, \hat{n}_{\tau-1})$ in Eq.(8) can be computed as

$$p(j_\tau = j \mid y_\tau, \hat{n}_{\tau-1}) = \frac{p_j N(y_\tau; \hat{\mu}_{y_\tau,j}, \Sigma_{y_\tau,j})}{\sum_{j=1}^{M} p_j N(y_\tau; \hat{\mu}_{y_\tau,j}, \Sigma_{y_\tau,j})}, \tag{10}$$

where $\hat{\mu}_{y_\tau,j} = \mu_{xj} + f(\mu_{xj}, \hat{n}_{\tau-1})$.

In the non-stationary environment, the history observation data is not useful or not really important to current noise estimation. We can add the different weights according to their contributions on current noise estimation. The different weights can be added by introducing the forgetting factor ρ where ρ is a non-negative constant with value less than 1, thus, Eq.(8) can be rewritten as

$$Q(\hat{n}_{t+1} \mid \Lambda_{nt}) = -\sum_{\tau=1}^{t+1}\rho^{t+1-\tau} \cdot \left\{\sum_{j=1}^{M} p(j_\tau = j \mid y_\tau, \hat{n}_{\tau-1})\left\{y_\tau - \hat{\mu}_{y_\tau,j}(\hat{n}_\tau)\right\}' \Sigma_{y_\tau,j}^{-1}\left\{y_\tau - \hat{\mu}_{y_\tau,j}(\hat{n}_\tau)\right\}\right\}. \tag{11}$$

By Taylor series expansion to the above auxiliary function, choosing the truncated second order items, and maximizing the approximated items with respect to the noise parameter, the noise at time $t+1$ can be estimated [8]-[11]

$$\hat{n}_{t+1} = \hat{n}_t + \gamma \cdot \{K_{t+1}(\hat{n}_t)\}^{-1} S_{t+1}(\hat{n}_t), \tag{12}$$

where the disturbing factor γ is a non-negative constant with value greater than 0, the Fisher information matrix $K_{t+1}(\hat{n}_t)$ and the score vector $S_{t+1}(\hat{n}_t)$ are defined as following

$$\begin{aligned} K_{t+1}(\hat{n}_t) &= -\left.\frac{\partial^2 Q(n \mid \Lambda_{nt})}{\partial^2 n}\right|_{n=\hat{n}_t} = \sum_{\tau=1}^{t+1}\rho^{t+1-\tau}\sum_{j=1}^{M} p(j_\tau = j \mid y_\tau, \hat{n}_{\tau-1}) \cdot B_j'(\hat{n}_{\tau-1})\Sigma_{y_\tau,j}^{-1}B_j(\hat{n}_{\tau-1}) \\ &= \rho \cdot K_t + \sum_{j=1}^{M} p(j_{t+1} = j \mid y_{t+1}, \hat{n}_t) B_j'(\hat{n}_t)\Sigma_{y_{t+1},j}^{-1}B_j(\hat{n}_t), \end{aligned} \tag{13}$$

$$S_{t+1}(\hat{n}_t) = \left.\frac{\partial Q(n \mid \Lambda_{nt})}{\partial n}\right|_{n=\hat{n}_t} = \sum_{j=1}^{M} p(j_{t+1} = j \mid y_{t+1}, \hat{n}_t) B_j(\hat{n}_t)' \Sigma_{y_{t+1},j}^{-1}\left\{y_t - \hat{\mu}_{y_t}(\hat{n}_t)\right\}. \tag{14}$$

4 Experimental Results

A continuous hidden Markov model (HMM)-based speech recognition system is used in the recognition experiments for examining the presented approach. The utterances of 82 speakers (41 males and 41 females) from the mandarin Chinese corpus provided by the 863 plan (China High-Tech Development Plan[12]) are trained for triphone-based HMM acoustic models, where each triphone unit was modeled as a three-emitting-state left-right topology with a mixture of 16 Gaussian per state and diagonal covariance matrices. The utterances of 9 speakers from the clean corpus are used for subsequent artificial contamination with different noise class.

In order to extract Mel frequency cepstral coefficients (MFCCs) from the 16Hz noisy speech data, we use a power spectrum which is calculated every 10ms on a 25ms Hmming window with pre-emphasis coefficient 0.97, then take a mel-scaled triangular filterbank and logarithmic computation and accordingly obtain the Mel-scaled 24 log-fiterbank coefficients. After transforming them into the cepstral domain with DCT transform, we obtain the first 12 cepstral coefficients (excluding the zero coefficients). The log-energy feature in each frame is computed after taking Hmming windowing. Accordingly, 39 dimensional features consisting of the 12 cepstral coefficients, the log-energy feature coefficient and their time derivatives are computed.

In our feature compensation paradigm, for modeling the clean speech, we extract a set of 24 MFCCs and one log-energy feature from the clean speech data for training and obtain a mixture of 128 Gaussian distributions. Then the mean vector of each mixture component in the Mel-scaled log spectral domain is obtained using inverse cosine transformation matrix. The covariance matrix is computed also from the cepstral domain using the inverse cosine transformation matrix and its transpose. By ignoring the off-diagonal elements in the covariance matrices assuming that the different coefficients are statistically independent, we obtain the diagonal covariance matrices in the log domain. With the developed sequential EM algorithm, the 24 dimensional log-filterbank features and a log-energy feature are compensated. With DCT transform and delta and delta-delta regression equations, the static coefficients (12 MFCCs plus the log-energy feature) and the corresponding dynamic coefficients (13 delta coefficients and 13 delta-delta coefficients) are computed.

In order to test the validity of the feature compensation algorithm, a number of experiments have been performed. They include the baseline without compensation, compensation with CMN (cepstral mean normalization), batch-EM estimation and sequential EM estimation. The forenamed three approaches are titled as “baseline”, “CMN” and “batch-EM”, respectively in Table 1 and Table 2. In the sequential EM estimation, to investigate the behavior in the non-stationary environment, we get three forms: “Seq-0.90”, “Seq-0.95” and “Seq-1.00” according to the different forgetting value ρ with 0.90, 0.95 and 1.00. And we add the stationary white noise and the non-stationary babble noise from NoiseX92 [13] to the test set according to different SNR varying from 0dB to 20dB. It is observed from Table 1 that, the sequential estimation gives considerable performances, compared with “baseline”, “CMN” and “Batch-EM”. For example, in the 5dB white noisy condition, “baseline” only achieves 2.54% recognition rate, “CMN” achieves 10.98% recognition rate, and “Batch-EM” achieves 17.00% recognition rate. The sequential estimation with the forgetting factor ρ set to 0.90, 0.95 and 1.00 gives 18.93%, 18.97% and 18.91% recognition rates and achieves

1.93%, 1.97%, and 1.91% improvements over that by "Batch-EM", respectively. As a whole, the sequential estimation with different forgetting factor value achieves 0.77%, 0.75%, and 0.82% improvements over that by "Batch-EM", respectively. It is clear that the presented approach is very effective in the stationary noisy condition.

Table 1. Recognition rates in the white noisy environment (%)

SNR	0dB	5dB	10dB	15dB	20dB	Avg.
baseline	0.32	2.54	11.14	30.00	56.04	20.01
CMN	3.31	10.98	29.99	36.94	61.65	28.57
Batch-EM	5.51	17.00	39.37	62.36	77.12	40.27
Seq-0.90	4.99	18.93	39.99	63.19	78.10	41.04
Seq-0.95	4.99	18.97	40.19	62.85	78.12	41.02
Seq-1.00	5.02	18.91	40.38	63.02	78.14	41.09

Table 2. Recognition rates in the babble noisy environment (%)

SNR	0dB	5dB	10dB	15dB	20dB	Avg.
baseline	3.87	24.61	54.84	62.98	80.23	45.31
CMN	11.16	32.38	55.95	71.04	80.31	50.17
Batch-EM	15.86	39.25	61.78	75.23	81.09	54.64
Seq-0.90	17.72	40.71	62.56	75.53	81.07	55.52
Seq-0.95	17.50	40.70	62.49	75.50	81.21	55.48
Seq-1.00	17.44	40.31	62.60	75.51	81.46	55.46

To test the validity of the sequential estimation in non-stationary conditions, we further test the babble noise in different SNR levels. It is observed in Table 2 that, using "baseline", performance degradation is not obvious in the high SNR condition, such as in the 20dB condition. But when the noise amount increases, recognition performance quickly deteriorates with only 3.87% recognition rate in the 0dB condition. With "CMN" and "Batch-EM", the phenomena can be relatively restrained. However, they still have the main limitations to cope with the non-stationary environments. Although compensation is applied to reduce the mismatch among the clean acoustic model and the test set, they remain a minor mismatch which they don't obtain the best performance at all of non-stationary noisy conditions. With the sequential estimation algorithm, it can further reduce the mismatch and can improve the system performance in most of noisy conditions, especially in low SNR conditions. For example, in 5dB condition, the sequential EM algorithm with different forgetting factor achieves 1.46%, 1.45% and 1.06% improvements in comparison to "Batch-EM", respectively.

From Table 1 and Table 2, we also observe that the sequential estimation averagely provides slight improvement when the forgetting factor ρ is 1.00 over that of ρ is 0.9 or 0.95 for the white noise. But we notice that it averagely provides slight improvement when ρ is 0.90 over that of ρ is 0.95 or 1.00 for the babble noise. The cause of this behavior is that the white noise is the stationary noise and the babble

noise is the non-stationary noise. For the white noisy condition, it is clear that the history data is very useful to noise estimation. With the reasonable forgetting factor, the presented approach can ignore the history data which is effective for computing the current noise parameter in the non-stationary condition. Due to the babble noise is a slow time-varying noise, the forgetting factor can be set with a high value relatively. For the highly time-varying conditions, ρ can be a low value to reasonably track the non-stationary characteristics.

5 Conclusions

We have presented an approach to environment compensation for robust speech recognition based on a sequential EM algorithm. The algorithm compensates entirely all of the features to deal with the environment corruption. The corruption causing distortion in the speech signal in the log domain can be modeled a nonlinear function and linearized by the truncated first-order VTS approximation. Furthermore, all of the clean cepstral coefficients and the log-energy feature are trained and postprocessed by taking corresponding inverse DCT transform to obtain a reasonable Gaussian mixture model in the log domain. They give a reasonable basis for the subsequent speech recognition. Experiment results show that the algorithm presented provides improvements of about 20% in the white noise and about 10% in the babble noise when compared with the performances under distortion environments. Moreover, the performance of speech recognition system by using sequential EM algorithm achieves considerable improvement compared with the traditional batch-EM algorithm. In the future work, we will investigate the relationship of the forgetting factor with the degree of the non-stationary characteristics and the noise class.

References

1. Stern, R.M., Raj, B., Moreno, P.J.: Compensation for Environmental Degradation in Automatic Speech Recognition. In: Proc. ESCA-NATO Tutorial Research Workshop Robust Speech Recognition for Unknown Communication Channels(1997)33–42
2. Moreno, P.J., Raj, B., Stern, R.M.: A Vector Taylor Series Approach for Environment-Independent Speech Recognition. In: Proceedings of IEEE(1995)733-736
3. Haifeng, S., Jun, G., Gang, L., and Qunxia, L.: Environment Compensation Based on Maximum a Posteriori Estimation for Improved Speech Recognition. Accepted in The Mexican International Conference on Artificial Intelligence (MICAI)(2005)
4. Raj, B., Gouvea, E.B., Moreno, P.J., Stern, R.M.: Cepstral Compensation by Polynomial Approximation for Environment-Independent Speech Recognition. In: Proceedings of Int. Conf. Spoken Language Processing, Philadelphia(1996)2340-2343
5. Kim, N.S., Kim, D.Y., Byung, K.G., Kim S.R.: Application of VTS to Environment Compensation with Noise Statistics. In: ESCA Workshop on Robust Speech Recognition. Pont-a-Mousson, France(1997)99-102
6. Kim, N.S.: Statistical Linear Approximation for Environment Compensation. IEEE Signal Processing Letters, 1(1998)8-10

7. Haifeng, S., Gang, L., Jun, G., and Qunxia, L.: Two-Domain Feature Compensation for Robust Speech Recognition. In: Wang, J., Liao, X., and Yi, Z. (eds.): Advance in Neural Network- ISNN 2005. Lecture Notes in Computer Science 3497, Springer-Verlag, Berlin Heidelberg New York(2005)351–356
8. Kim, N.S.: Nonstationary Environment Compensation Based on Sequential Estimation. IEEE Signal Processing Letters, 3(1998)8-10
9. Zhao, Y., Wang, S., Yen, K.C.: Recursive Estimation of Time-Varying Environments for Robust Speech Recognition. In: Proceedings of IEEE(2001)225-228
10. Deng, L., Droppo, J., Acero, A.: Recursive Noise Estimation Using Iterative Stochastic Approximation for Stereo-Based Robust Speech Recognition. In: Proceedings of IEEE (2002)81-84
11. Krishnamurthy, V., Moore, J.B.: "Online Estimation of Hidden Markov Model Parameters Based on the Kullback-Leibler Information Measure. IEEE Trans. Sig. Proc, 8 (1993)2557-2573
12. Zu, Y. Q.: Issues in the Scientific Design of the Continuous Speech Database. Available: http://www.cass.net.cn/chinese/s18_yys/yuyin/report/report_1998.htm.
13. Varga, A., Steenneken, H. J. M., Tomilson, M., Jones, D.: The NOISEX–92 Study on the Effect of Additive Noise on Automatic Speech Recognition. Tech. Rep. DRA Speech Research Unit(1992)

Appendix

If we only consider the additive noise, the corruption in the signal domain is shown as following

$$y_t = x_t + n_t \,, \tag{15}$$

where y_t denotes the noisy sample, x_t for the clean sample, n_t for the additive noise.

Generally we assume that x_t and n_t are statistically independent. If we transform the above relation into the power spectral domain, the corruption can be expressed as:

$$Y(\omega) = X(\omega) + N(\omega) \,, \tag{16}$$

where $Y(\omega)$, $X(\omega)$ and $N(\omega)$ represent the power spectrum of the noisy speech, clean speech and additive noise, respectively. If we take a logarithmic computation on both sides of Eq.(16),

$$\begin{aligned} &\log\{Y(\omega)\} = \log\{X(\omega) + N(\omega)\} \\ &= \log\{X(\omega)\} + \log\left\{1 + \frac{N(\omega)}{X(\omega)}\right\} \\ &= \log\{X(\omega)\} + \log\left\{1 + \exp\left\{\log\{N(\omega)\} - \log\{X(\omega)\}\right\}\right\}. \end{aligned} \tag{17}$$

Let $y = \log\{Y(\omega)\}$, $x = \log\{X(\omega)\}$ and $n = \log\{N(\omega)\}$, we have [2]

$$y = x + \log(1 + \exp(n - x)) = x + f(x, n) \,, \tag{18}$$

where y, x and n are respectively the noisy speech, the clean speech and the noise in the log spectral domain. From Eq.(18), For each log-filterbank bin, it is noticeable that the corruption becomes a complex nonlinear contamination procedure.

Now we describe the log-energy feature contamination procedure. In order to attenuate the discontinuities at the window, we generally use the Hmming window before extracting the feature coefficients. The energy of one frame after taking Hmming windowing on speech can be written as

$$E_y = \sum_{l=1}^{L} \left[h(y_l) \right]^2 , \tag{19}$$

where L denotes the number of samples in each frame, E_y is the noisy energy, $h(\cdot)$ represents operation with Hmming windowing. Due to the clean speech and the noise are statistical independent and $h(\cdot)$ is a linear computation, the above equation can be rewritten as

$$\begin{aligned} E_y &= \sum_{l=1}^{L} \left\{ h(x_l + n_l) \right\}^2 = \sum_{l=1}^{L} \left\{ h(x_l) + h(n_l) \right\}^2 \\ &= \sum_{l=1}^{L} \left\{ h(x_l) \right\}^2 + \sum_{l=1}^{L} \left\{ h(n_l) \right\}^2 = E_x + E_n , \end{aligned} \tag{20}$$

where E_x and E_n are respectively the clean energy and the noise energy in one frame. If we take a logarithmic transformation on Eq.(20), the corruption of the log-energy feature is

$$y_e = x_e + \log(1 + \exp(n_e - x_e)) , \tag{21}$$

in which y_e, x_e and n_e are respectively the noisy log-energy feature , the clean log-energy feature and the noise, $y_e = \log(E_y)$, $x_e = \log(E_x)$, $n_e = \log(E_n)$.

As seen in Eq.(18) and Eq.(21), the corruptions of the log-filtebank coefficients and the log-energy feature have the same functional form.

Hybrid Cost-Sensitive Decision Tree

Shengli Sheng and Charles X. Ling

Department of Computer Science, The University of Western Ontario,
London, Ontario N6A 5B7, Canada
{cling, ssheng}@ csd.uwo.ca

Abstract. Cost-sensitive decision tree and cost-sensitive naïve Bayes are both new cost-sensitive learning models proposed recently to minimize the total cost of test and misclassifications. Each of them has its advantages and disadvantages. In this paper, we propose a novel cost-sensitive learning model, a hybrid cost-sensitive decision tree, called DTNB, to reduce the minimum total cost, which integrates the advantages of cost-sensitive decision tree and of the cost-sensitive naïve Bayes together. We empirically evaluate it over various test strategies, and our experiments show that our DTNB outperforms cost-sensitive decision and the cost-sensitive naïve Bayes significantly in minimizing the total cost of tests and misclassification based on the same sequential test strategies, and single batch strategies.

1 Introduction

Inductive learning techniques have had great success in building classifiers and classifying test examples into classes with a high accuracy or low error rate. However, in many real-world applications, lowing misclassification error is not the goal as "errors" can cost very differently. This type of learning is called cost-sensitive learning. Turney [14] surveys a whole range of costs in cost-sensitive learning, among which two types of costs are most important: misclassification costs and test costs. For example, in a binary classification task, the cost of false positive (FP) and the cost of false negative (FN) are often very different. In addition, attributes (tests) may have different costs, and acquiring values of attributes also incurs costs. The goal of learning is to minimize the sum of the misclassification costs and the test costs.

Tasks involving both misclassification and test costs are abundant in real-world applications. For example, when building a model for medical diagnosis from the training data, we must consider the cost of tests (such as blood tests, X-ray, etc.) and the cost of misclassifications (errors in the diagnosis). Further, when a doctor sees a new patient (a test example), tests are normally ordered, at a cost to the patient or his/her insurance company. To better diagnose or predict the disease of the patient (i.e., reducing the misclassification cost). Doctors must balance the trade-off between potential misclassification costs and test costs to determinate which tests should be ordered, and at what order, to reduce the expected total cost. A case study on heart disease is given in the paper.

In this paper, we propose a new cost-sensitive learning model, DTNB, which integrates the advantages of the cost-sensitive decision tree and the cost-sensitive naïve Bayes, both of which minimize the total cost of misclassifications and tests.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 274 – 284, 2005.

DTNB uses the cost-sensitive decision tree to collect the required tests for test examples, and uses the cost-sensitive naïve Bayes to classify. For a test example, after the required tests are collected according to the cost-sensitive decision tree, the tests are performed with a cost and their results are available. Then the cost-sensitive naïve Bayes built on all the training data is applied to classify the test example. The naïve Bayes model can make use of the known values which do not appear in the path which the test example follows to go down to a leaf in the cost-sensitive decision tree. Thus, we can expect that the cost-sensitive DTNB can achieve lower total cost than the cost-sensitive decision tree and the cost-sensitive naïve Bayes do alone.

The rest of paper is organized as follows. We first review the related work in Section 2. Then we describe our new cost-sensitive learning model, DTNB, to reduce the minimum total cost of tests and misclassifications in Section 3. In Section 4, we present empirical experiments. The paper concludes with discussion and some directions for the future work.

2 Review of Previous Work

Cost-sensitive learning has received extensive attentions in recent years. Turney [14] analyzes a variety of costs in machine learning, such as misclassification costs, test costs, active learning costs, computation cost, human-computer interaction cost, etc. Two types of costs are singled out as the most important in machine learning: misclassification costs and test costs, and test costs are normally considered in conjunction with misclassification costs. Much work has been done in considering non-uniform misclassification costs (alone), such as [4, 5, 7]. Those works can often used to solve problem of learning with very imbalanced datasets [3]. Some previous work, such as [10, 12], consider the test cost alone without incorporating misclassification cost. As pointed out by [14] it is obviously an oversight. As far as we know, the only work considering both misclassification and test costs includes [13, 15, 9, 2]. We discuss these works in detail below.

In [15], the cost-sensitive learning problem is cast as a Markov Decision Process (MDP), and an optimal solution is given as a search in a state space for optimal policies. While related to our work, their research adopts an optimal search strategy, which may incur very high computational cost to conduct the search. In contrast, we adopt the local search similar to [11] using a polynomial time algorithm to build a new decision trees, and our test strategies are also polynomial to the tree size. (Greiner et al. 2002) studied the theoretical aspects of active learning with test costs using a PAC learning framework, which models how to use a budget to collect the relevant information for the real-world applications with no actual data at beginning. Our algorithm builds a model from history data to minimize the total cost of misclassification and tests for a new case with missing values. Turney [13] presented a system called ICET, which uses a genetic algorithm to build a decision tree to minimize the cost of tests and misclassification. Our algorithm essentially adopts the same decision-tree building framework as in [11], and it is expected to be more efficient than Turney's genetic algorithm based approach.

Ling et al. [9] propose a cost-sensitive decision tree learning program that minimizes the total cost of tests and misclassifications. They also propose several test

strategies, and compare their results to C4.5. However, for a test example, the cost-sensitive decision tree ignores the information supplied by the known attributes which do not appear in the path which the test example follows to go down to a leaf in the cost-sensitive decision tree. Chai et al. [2] propose a cost-sensitive naïve Bayes based algorithm, called CSNB, which searches for minimal total cost of tests and misclassifications. They also propose a sequential test strategy and a single batch test strategy. However, the cost-sensitive naïve Bayes does not learn the general attribute structure (such as the tree structure) but only probability tables from training data. The test sequence for each test example is less comprehensible.

Our model, DTNB, combines the advantages of cost-sensitive decision tree and naïve Bayes. It utilizes the structure of the cost-sensitive decision tree to collect the beneficiary tests for a test example and makes use of the information in the known attributes which are ignored by the cost-sensitive decision tree to reduce the misclassification cost. We expect that our DTNB outperform cost-sensitive decision tree and cost-sensitive naïve Bayes alone in terms of the total cost of tests and misclassification.

The new cost-sensitive model, DTNB, is composed of decision tree and naïve Bayes, but it is much different from NBTree [8] proposed by Kohavi. First of all, NBTree is not a cost-sensitive learning model. The learning algorithm of NBTree is similar to C4.5 [Qui93]. DTNB is a cost-sensitive learning to minimize the total cost of tests and misclassification. Secondly, in NBTree, a naïve Bayes is constructed for each leaf using the data associated with the leaf. However, DTNB only constructs one naïve Bayes using all the training data. This naïve Bayes acts as a hidden node at each node (including the leaves) of the cost-sensitive decision tree. The details of difference between NBTree and DTNB are explained in Section 3.

3 The New Cost-Sensitive Learning - DTNB

We assume that we are given a set of training data (with possible missing attribute values), the misclassification costs, and test costs for each attribute. We propose a novel cost-sensitive learning model, DTNB, which combines the advantages of cost-sensitive decision tree and naïve Bayes. The rationale of DTNB is based on our observations. We note that cost-sensitive decision tree has the ability of learning a general structure, and the structure of the tree plays an important role for collecting the most beneficiary unknown values. However, the decision tree ignores the original known values which do not appear in the tree for classify a test example. In non-cost-sensitive learning, this is one reasonable feature of decision tree. But in cost-sensitive learning, any value is available with a certain cost. We do not want waste any available information. Naturally, making use of all known values can reduce the total cost. The information of the known attributes which do not appear in the path through which the test example goes down to a leaf of the tree is useful for cost-sensitive classification to reduce the misclassification cost. Fortunately, cost-sensitive naïve Bayes indeed utilizes all known attributes for misclassification, but it does not have a structure learning ability to help determine which tests and in what order should be done for unknown attributes.

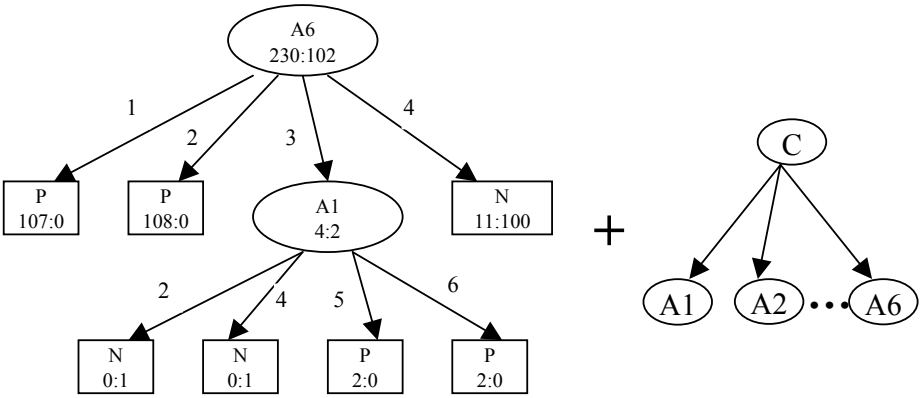


Fig. 1. An example of cost-sensitive DTNB

In order to overcome these drawbacks and combine those advantages in the two cost-sensitive models, we propose a novel cost-sensitive learning model, which integrates cost-sensitive decision tree with cost-sensitive naïve Bayes, called DTNB. Figure 1 shows the structure of an example of the novel cost-sensitive learning model DTNB. We can see DTNB is an integration model with two parts. The left part is a cost-sensitive decision tree which is used for finding the required tests for each testing example. Besides the cost-sensitive tree, DTNB also contains a naïve Bayes (right part), which is for classification.

First of all, DTNB builds a cost-sensitive decision tree, given a set of training data, the misclassification costs, and test costs for each attribute. The building procedure is similar to C4.5. Instead of using entropy based splitting criteria, we use the *expected* total misclassification cost to select an attribute for splitting. This gives a more accurate choice for attribute selection. That is, an attribute may be selected as a root node of a decision tree if the sum of the test cost and the expected misclassification costs of all branches is the minimum among other attributes, and is less than that of the root. For a subset of examples with tp positive examples and tn negative examples, if $C_P = tp \times TP + tn \times FP$ is the total misclassification cost of being a positive leaf, and $C_N = tn \times TN + tp \times FN$ is the total misclassification cost of being a negative leaf, then the probability of being positive is estimated by the relative cost of C_P and C_N; the smaller the cost, the larger the probability (as minimum cost is sought). Thus, the probability of being positive is: $1 - \frac{C_P}{C_P + C_N} = \frac{C_N}{C_P + C_N}$. The expected misclassification cost of being positive is: $E_P = \frac{C_N}{C_P + C_N} \times C_P$. Similarly, the probability of being a negative leaf is $\frac{C_P}{C_P + C_N}$; and the expected misclassification

cost of being negative is: $E_N = \frac{C_P}{C_P + C_N} \times C_N$. Therefore, without splitting, the expected total misclassification cost of a given set of examples is: $E = E_P + E_N = \frac{2 \times C_P \times C_N}{C_P + C_N}$. If an attribute A has l branches, then the expected total misclassification cost after splitting on A is: $E_A = 2 \times \sum_{i=1}^{l} \frac{C_{P_i} \times C_{N_i}}{C_{P_i} + C_{N_i}}$. Thus, ($E - E_A - T_C$) is the expected cost reduction splitting on A, where T_C is the total test cost for all examples on A. It is easy to find out which attribute has the smallest expected total cost (the sum of the test cost and the expected misclassification cost), and if it is smaller than the one without split (if so, it is worth to split). With the expected total misclassification cost described above as the splitting criterion, the lazy-tree learning algorithm is shown in Figure 2.

Simultaneously, we build a cost sensitive naïve Bayes. Note that this model is built on all the training data, and for all nodes in the tree. However, NBTree [Koh96] treats the segmentation of decision tree as an advantage. It builds a naïve Bayes at each leaf of the decision tree. And the naïve Bayes constructed for a leaf uses only the data associated with the leaf. However, as the tree grows, the training data are split into the lower level nodes. Finally, there are very little data in the leaves. The classification based on these leaves is far less accurate, so that the misclassification cost goes higher. This is reason that NBTree is proposed for larger dataset. However, without larger dataset assumption DTNB overcomes the shortcoming of segmentation of decision tree by constructing only one naïve Bayes using all the training data. This naïve Bayes acts as a hidden model at each node (including the leaves) of the cost-sensitive decision tree. The hidden model is only for classification. Thus, DTNB does not utilize the data which go down into a leaf of the tree to classify a testing example which drops into this leaf. It classifies the test example by the only hidden cost-sensitive naïve Bayes.

DTNB only builds one general naïve Bayes from all the training data. Whereas, the posterior probabilities of a test example e are computed from the known attributes and the tested unknown attributes. The unknown attributes which are not selected to perform testing are not concerned. With the posterior probabilities, if $FN \times P(+|e) > FP \times P(-|e)$, this test example is classified as negative, otherwise, as positive. A misclassification cost may be incurred if the prediction of the test example is wrong. Thus, for each test example, not only the attributes appearing on the tree, but also the known attributes can be fully used to make correct classification, so that the total misclassification cost can be reduced, as any known value is worthy of a certain cost. But for the cost-sensitive decision tree, it is possible some known attributes are not used to split the training data, so that they become useless for the classification. DTNB makes use of all known attributes, as well as the available values of the collected unknown attributes at certain test costs.

CSDT(*Examples, Attributes, TestCosts*)

1. Create a *root* node for the tree
2. If all examples are positive, return the single-node tree, with *label* = +
3. If all examples are negative, return the single-node tree, with *label* = -
4. If attributes is empty, return the single-node tree, with label assigned according to *min* (E_P, E_N)
5. Otherwise Begin
   a. If *maximum cost reduction* < *0* return the single-node tree, with label assigned according to *min* (E_P, E_N)
   b. *A* is an attribute which produces maximum cost reduction among all the remaining attributes
   c. Assign the attribute *A* as the tree *root*
   d. For each possible value vi of the attribute *A*
      i. Add a new branch below root, corresponding to the test $A=v_i$
      ii. Segment the training examples into each branch $Example\_v_i$
      iii. If no examples in a branch, add a leaf node in this branch, with label assigned according to *min* (E_P, E_N)
      iv. Else add a subtree below this branch, CSDT($examples\_v_i$, *Attributes-A, TestCosts*)
6. End
7. Return *root*

Fig. 2. Algorithm of cost-sensitive decision tree

In the naïve Bayes model of DTNB, the Laplace Correction is applied. That is, $p(a \mid +) = \frac{N_a + 1}{N + m}$, where *Na* is the number of instances whose attribute $A_1=a$, *N* is the number of instances whose class is +, and m is the number of classes.

After DTNB is built, for each testing example, there are two steps to find the minimum total cost of tests and misclassifications. The first step is to utilize the tree structure of the cost-sensitive decision tree to collect a set of tests which need be performed according to a certain strategy (there are several strategies explained in Section 4). The total test cost is accumulated in the step. After the set of tests are done, the values of the unknown attributes in the test example are available. It automatically goes to the second step, where the cost-sensitive naïve Bayes model is used to classify the test example into a certain class. The naïve Bayes uses not only the unknown attributes tested but also all known attributes. If it is classified incorrectly, there is misclassification cost. We empirically evaluate it over various test strategies in next section.

4 Experiments

We evaluate the performance of DTNB on two categories of test strategies: Sequential Test, and Single Batch Test. For a given test example with unknown attributes, the

Sequential Test can request only one test at a time, and wait for the test result to decide which attribute to be tested next, or if a final prediction is made. The Single Batch Test, on the other hand, can request one set (batch) of one or many tests to be done simultaneously before a final prediction is made.

4.1 DTNB's Optimal Sequential Test

Recall that Sequential Test allows one test to be performed (at a cost) each time before the next test is determined, until a final prediction is made. Ling, et al. [9] described a simple strategy called *Optimal Sequential Test* (or OST in short) that directly utilizes the decision tree built to guide the sequence of tests to be performed in the following way: when the test example is classified by the tree, and is stopped by an attribute whose value is unknown, a test of that attribute is made at a cost. This process continues until the test case reaches a leaf of the tree. According to the leaf reached, a prediction is made, which may incur a misclassification cost if the prediction is wrong. Clearly the time complexity of OST is only linear to the depth of the tree.

One weakness with this approach is that it ignores some known attributes which do not appear in the path through which a test example goes down to a leaf. However, these attributes can be useful for reducing the misclassification cost. Like the OST, We also propose an Optimal Sequential Test strategy for DTNB (section 3), called DNOST in short. It has the similar process as OST. The only difference is that the class prediction which is not made by the leaf it reached, but the naïve Bayesian classification model in DTNB. This strategy utilizes the tree structure to collect the most useful tests for a test example. And it also utilizes the entire original known attributes in the test example with the unknown attributes tested to predict the class of the test example. We can expect DNOST outperforms OST.

Table 1. Datasets used in the experiments

	No. of Attributes	No. of Examples	Class dist. (N/P)
Ecoli	6	332	230/102
Breast	9	683	444/239
Heart	8	161	98/163
Thyroid	24	2000	1762/238
Australia	15	653	296/357
Tic-tac-toe	9	958	332/626
Mushroom	21	8124	4208/3916
Kr-vs-kp	36	3196	1527/1669
Voting	16	232	108/124
Cars	6	446	328/118

Comparing Sequential Test Strategies. To compare various sequential test strategies, we choose 10 real-world datasets which are listed in Table 1, from the UCI Machine Learning Repository [1]. The datasets are first discretized using the minimal entropy method [6]. These datasets are chosen because they are binary class, have at least some discrete attributes, and have a good number of examples. Each dataset is split into two parts: the training set (60%) and the test set (40%). Unlike the case study of heart disease, the detailed test costs and group information [13] of these datasets are unknown. To make the comparison possible, we simply choose randomly the test costs of all attributes to be some values between 0 and 100. This is reasonable because we compare the relative performance of all test strategies under the same chosen costs. To make the comparisons straightforward, we set up the same misclassification costs 200/600 (200 for false positive and 600 for false negative). For test examples, a certain ratio of attributes (0.2, 0.4, 0.6, 0.8, and 1) are randomly selected and marked as unknown to simulate test cases with various degrees of missing values.

In this section, we compare our DNOST with the other two sequential test strategies available, OST, and CSNB [2] on 10 real-world datasets to see which one is better (having a smaller total cost). Note that DNOST and OST use the same decision tree to collect beneficiary tests. However, DNOST uses DTNB's naïve Bayes for classification, while OST uses the leaves of tree to classify test examples. CSNB follows the same test strategy: determine next test based on the previous test result. However, it is based on the naïve Bayes only. In all, all of them are based on the same test strategy, but they are applied different cost-sensitive learning models. That is, their performances directly stand for the performances of different learning models. We repeat this process 25 times, and the average total costs for the 10 datasets are plotted in Figure 3.

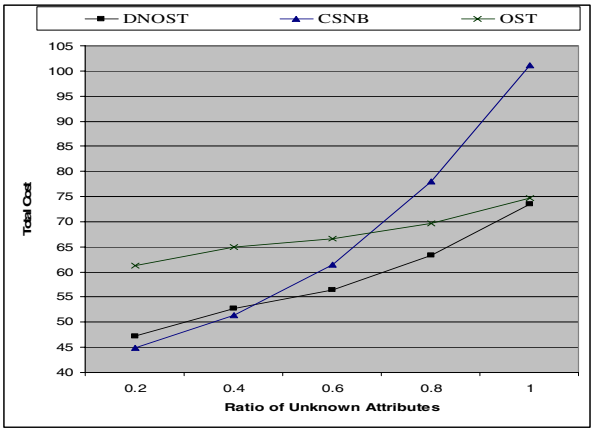


Fig. 3. The total cost of our new Sequential Test Strategy DNOST compared to previous strategies (OST and CSNB)

We can make several interesting conclusions. First, DNOST performs the best among the three sequential test strategies. When the unknown attribute ratio is higher, the difference between DNOST and CSNB becomes bigger. However, DNOST is gradually close to OST when the unknown ratio is increased. When the unknown ratio is lower, the difference between DNOST and OST is bigger, as more known attributes are utilized in DTNB, but they are ignored in cost-sensitive decision tree. Second, the results proof our expectation which DTNB integrates the advantage of the decision tree and the naïve Bayes and overcomes their defects. When the unknown ratio is lower, there are more known attributes ignored by OST, so that OST performs worse, whereas DNOST and CSNB perform better and are closer, as they make use of the known values. When the unknown ratio is higher, there are less known attributes ignored by OST and both DNOST and OST utilize the tree structure to collect the most beneficiary tests, so that they perform better and are close to each other.

4.2 Single Batch Test Strategies

The Sequential Test Strategies have to wait for the result of each test to determine which test will be the next one. Waiting not only costs much time, but also increases the pressure and affects the life quality of patients in medical diagnosis. In manufacturing diagnoses, it delays the progress of engineering. Even in some particular situations, for example, emergence, we have to make decisions as soon as possible. In medical emergence, doctors normally order one set of tests (at a cost) to be done at once. This is the case of the Single Batch Test.

In [9] a very simple heuristic is described. The basic idea is that when a test example is classified by a minimum-cost tree and is stopped by the first attribute whose value is unknown in the test case, all unknown attributes under and including

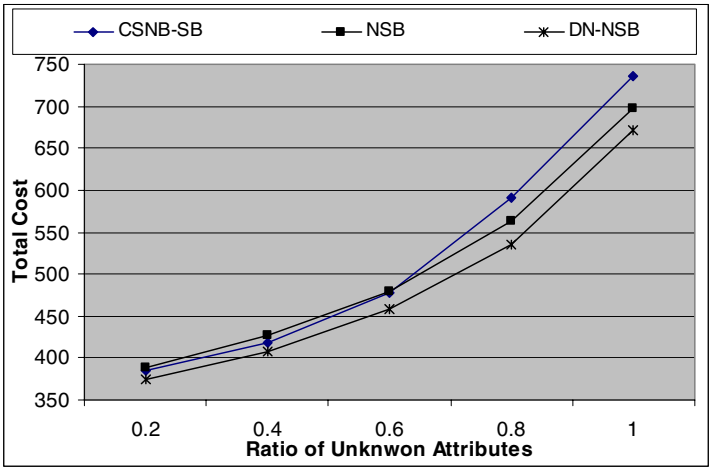


Fig. 4. The total cost of our new Single Batch Test Strategies DN-NSB compared to their previous strategies (NSB and CSNB-SB)

this first attribute would be tested, as a single batch. Clearly, this strategy would have exactly the same misclassification cost as the Optimal Sequential Test, but the total test cost is higher as extra tests are performed. This strategy is called Naïve Single Batch (NSB).

The weakness of NSB is that it ignores some known attributes which do not appear in the path through which a test example goes down to a leaf after the tests are performed. However, these attributes can be useful for reducing the misclassification cost. Like the NSB, we apply the similar process on DTNB. The only difference is the class prediction which is not made by the leaf a test example reached after the tests are performed, but by the naïve Bayes classification model. We call this process DTNB's Naïve Single Batch Test (or DN-NSB in short).

Comparing Single Batch Test Strategies. We use the same experiment procedure on the same 10 datasets used in Section 4.1 (see Table 1) to compare various Single Batch Test strategies including CSNB-SB [2]. The only change is the misclassification costs, which are set to 2000/6000 (2000 for false positive and 6000 for false negative). The misclassification costs are set to be larger so the trees will be larger and the batch effect is more evident. Note that DN-NSB and NSB use the same decision tree to collect beneficiary tests. However, DN-NSB uses DTNB's naïve Bayes for classification, while NSB uses the leaves of tree to classify test examples. CSNB follows the same test strategy: request one set (batch) of one or many tests to be done simultaneously before a final prediction is made. However, it is based on the naïve Bayes only. In all, all of them are based on the same test strategy, but they are applied to different cost-sensitive learning models. That is, their performances directly stand for the performances of different learning models. The total costs for the 10 datasets are compared and plotted in Figure 4.

We can make several interesting conclusions. First, the single batch test strategy (DN-NSB) based on DTNB outperforms others on any unknown ratio. CSNB-SB outperforms NSB when the unknown ratio is higher, but it is worse than NSB when the unknown ratio goes down. Second, the results again proof our expectation which DTNB integrates the advantage of the decision tree and the naïve Bayes and overcomes their defects. When the unknown ratio is lower, there are more known attributes ignored by NSB, so that NSB performs worse. DN-NSB and CSNB-SB perform better, as they make use of the known values. When the unknown ratio is higher, there are less known attributes ignored by NSB and both DN-NSB and NSB utilize the tree structure to collect the most beneficiary tests, so that they perform better.

5 Conclusion and Future Work

In this paper, we present a hybrid decision tree learning algorithm, which integrate with naïve Bayes, to minimize the total cost of misclassifications and tests. We evaluate the performance (in terms of the total cost) empirically, compared to previous methods using decision tree and naïve Bayes alone. The results show that our novel learning algorithm, DTNB, performs significantly better than the decision tree learning and the naïve Bayes learning alone.

In our future work we plan to design smart single batch test strategies. We also plan to incorporate other types of costs in our hybrid decision tree learning DTNB and test strategies.

References

1. Blake, C.L., and Merz, C.J., *UCI Repository of machine learning databases (website)*. Irvine, CA: University of California, Department of Information and Computer Science (1998).
2. Chai, X., Deng, L., Yang, Q., and Ling,C.X., Test-Cost Sensitive Naïve Bayesian Classification. *In Proceedings of the Fourth IEEE International Conference on Data Mining*. Brighton, UK : IEEE Computer Society Press (2004).
3. Chawla,N.V., Japkowicz, N., and Kolcz, A. eds., *Special Issue on Learning from Imbalanced Datasets*. SIGKDD, 6(1): ACM Press (2004).
4. Domingos, P., MetaCost: A General Method for Making Classifiers Cost-Sensitive. *In Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 155-164. San Diego, CA: ACM Press (1999).
5. Elkan, C., The Foundations of Cost-Sensitive Learning. *In Proceedings of the Seventeenth International Joint Conference of Artificial Intelligence*, 973-978. Seattle, Washington: Morgan Kaufmann (2001).
6. Fayyad, U.M., and Irani, K.B., Multi-interval discretization of continuous-valued attributes for classification learning. *In Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1022-1027. France: Morgan Kaufmann (1993).
7. Ting, K.M., Inducing Cost-Sensitive Trees via Instance Weighting. *In Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, 23-26. Springer-Verlag (1998).
8. Kohavi, R., Scaling up the accuracy of Naïve-Bayes Classifier: a Decision-Tree Hybrid. *In Proceeding of the Second International Conference on Knowledge Discovery and Data Mining* (KDD96). AAAI Press (1996) 202-207.
9. Ling, C.X., Yang, Q., Wang, J., and Zhang, S., Decision Trees with Minimal Costs. *In Proceedings of the Twenty-First International Conference on Machine Learning*, Banff, Alberta: Morgan Kaufmann (2004).
10. Nunez, M., The use of background knowledge in decision tree induction. *Machine learning*, 6:231-250 (1991).
11. Quinlan, J.R. eds., *C4.5: Programs for Machine Learning*. Morgan Kaufmann (1993).
12. Tan, M., Cost-sensitive learning of classification knowledge and its applications in robotics. *Machine Learning Journal*, 13:7-33 (1993).
13. Turney, P.D., Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *Journal of Artificial Intelligence Research* 2:369-409 (1995).
14. Turney, P.D., Types of cost in inductive concept learning. *In Proceedings of the Workshop on Cost-Sensitive Learning at the Seventeenth International Conference on Machine Learning*, Stanford University, California (2000).
15. Zubek, V.B., and Dietterich, T., Pruning improves heuristic search for cost-sensitive learning. *In Proceedings of the Nineteenth International Conference of Machine Learning*, 27-35, Sydney, Australia: Morgan Kaufmann (2002).

Characterization of Novel HIV Drug Resistance Mutations Using Clustering, Multidimensional Scaling and SVM-Based Feature Ranking

T. bıa S ı g[1], Va e. ı a S. ıche[2], Nı . Bee e. ı e[3],
F a. ce ca Cecche ı ı-Sı be . eı[2], Ma ı Da . e[4], R. f Kaı e[4],
Ha e Wa e[5], K a . K . . [5], Da ıe H . . a . [6], Ma Oe e[7],
J . ge K. R. c . . h[8], Ge Ea e. he e[4],
Ca . -Fede ıc Pe . . [2], a d Th . a Le ga e[1]

[1] Max Planck Institute for Informatics, Saarbrücken, Germany*
[2] University of Rome "Tor Vergata", Italy
[3] University of California, Berkeley, CA, USA
[4] University of Cologne, Germany
[5] University of Erlangen-Nürnberg, Germany
[6] Center for Advanced European Studies and Research, Bonn, Germany
[7] University of Düsseldorf, Germany
[8] University of Bonn, Germany

Abstract. We present a case study on the discovery of clinically relevant domain knowledge in the field of HIV drug resistance. Novel mutations in the HIV genome associated with treatment failure were identified by mining a relational clinical database. Hierarchical cluster analysis suggests that two of these mutations form a novel mutational complex, while all others are involved in known resistance-conferring evolutionary pathways. The clustering is shown to be highly stable in a bootstrap procedure. Multidimensional scaling in mutation space indicates that certain mutations can occur within multiple pathways. Feature ranking based on support vector machines and matched genotype-phenotype pairs comprehensively reproduces current domain knowledge. Moreover, it indicates a prominent role of novel mutations in determining phenotypic resistance and in resensitization effects. These effects may be exploited deliberately to reopen lost treatment options. Together, these findings provide valuable insight into the interpretation of genotypic resistance tests.

Keywords: HIV, clustering, multidimensional scaling, support vector machines, feature ranking.

1 Introduction

1.1 Background: HIV Combination Therapy and Drug Resistance

H . a ı . . . de cıe c . ı . HIV-1 ı he ca . a ı e age . f he ac ı ed ı . . . de cıcıe c . . d . . e AIDS, a dı ea e ı hıch e ı e . ı -ı d ced

* This work was conducted in the context of the European Union Network of Excellence BioSapiens (grant no. LHSG-CT-2003-503265). T.S. would like to thank Oliver Sander for the lively and stimulating discussions on the topic of this paper.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 285–296, 2005.

de e 1 . . f he e T ce . ead . 1 . . e fa1 . e a d dea h d e . . . . . 1 1c 1 fec 1 . . . Wh1 e . da e he e 1 . . c . e f . HIV 1 fec 1 . , he 1 . . d c 1 . . f h1gh ac 1 e a 1 e . . 1 a he a (HAART), 1 h1ch h ee . 1 a 1 e . . 1- . a d g a e ad 1 1 e ed 1 c . b1 a 1 . , ha . 1g 1 ca 1 . . ed 1fe a 1 a d . . 1 a 1 e . f a 1e . . H e e , 1 c . e e . . e 1 . . f HIV e 1ca- 1 . b c . . e d g , c . b1 ed 1 h h1gh . a 1 . a d e 1ca 1 . . a e . f HIV 1 a e . e 1 he e ec 1 . . f 1 a . . a 1 . . ca . 1 g e 1 a ce- c . fe . 1 g . a 1 . . 1 he1 ge . . e . The a 1 . . f he e . a1 . 1 he . - a 1 . e e a ead . he a fa1 . e, . . h1ch a e c . b1 a 1 . . f d g ha . be ch . e a . e - 1 e . eg1 e . .

1.2 Motivation: Evidence for Additional Resistance-Associated Mutations and Mutational Clusters

T da e, he dec1 1 . f . f . - d g c . b1 a 1 . . 1 a 1e . fa1 1 g he a 1 . . 1 e ba ed . . e e c1 g he e e a ge . . 1c . eg1 . . f he 1 a . . - a 1 . ha b ed b he 1 d1 1d a . The e e ce 1 he a a ed . 1de 1f he . e e ce . f e 1 a ce-a . c1a ed . a 1 . . f . each . f he 19 d g c . . e a a1 ab e f . a 1-HIV he a , b . 1 g . a 1 . 1 a . a da ed b he I e . a 1 . a AIDS S c1e (IAS) [1] . . . he a e . f h . a e e . .

The 1 a 1 . 1 c . 1ca ed b he fac ha . e 1 a ce . a 1 . . d . . ac- c . a e 1 de e de f . . each . he . . Ra he . , he a e . . . e 1 e- . de ed a . . g . a 1 . a a h a . , ead1 g . d1 1 c . a 1 . a c . e e . . c . - e . .[1] Ra 1 . a he a a . 1 g 1 . e e e c . . . 1 ed b . . 1 1 ed . de- a d1 g . f he e e ec . . I c ea 1 g e 1de ce . . add1 1 . a . a 1 . . 1 . . . ed 1 he de e . . e . f d g e 1 a ce [2,3], be 1de h . e 1 ed b he IAS, . . 1de he 1 ce 1 e f . . . . e e . . d .

1.3 Outline

We de c 1be a a . . ach . a d he d1 c . e a d cha ac e 1 a 1 . . f . . e . a 1 . . a . c1a ed 1 h he a fa1 . e f . . a a ge e a 1 . a da aba e, a d he1 e . 1 . a a d he . 1c cha ac e 1 a 1 . . 1 g . e 1 ed a d . . - e 1 ed . a 1 1ca ea 1 g e h d . We f c . . . e 1 a ce aga1 . . e e d g f . . he c a . f . c e . 1de e e e . a . c 1 a e 1 h1b1 . . (NRTI), h1ch a - ge a HIV . . e1 ca ed e e e . a . c 1 a e (RT). Th1 e . . e 1 e . . . 1b e f . . a . a 1 g he RNA ge . . e . f HIV bac . DNA . 1 . . 1 . 1 eg a 1 . 1 . he h . a ge . . e. NRTI a e a . g e . f he a . a b 1 d1 g b . c . f DNA, b ac a g . e e 1a f . cha1 e . ga 1 . . Th . , 1 c . . . a 1 . . f a . c e . 1de a a . g e d . 1 g DNA . . e 1 a 1 . e . 1 a e he cha1 e . . ga 1 . . . ce . .

The . . edge d1 c . e . . ce . de c 1bed 1 h1 a e c . b1 e he e . - ge e . da a f . . h ee d1 e e 1 . . g1ca ce e . . T a . f . 1 eg a ed

[1] Throughout this paper, the words *complex*, *cluster*, and *pathway* are used interchangeably.

a a 1, he e da a a e ed 1 a e a 1 a da aba e, h e c e 1 - 1 ed 1 ec 1 2. S e a 1c 1 1 g f a 1 1 h d1 e 1 g e 1 1e 1 NRTI- ea ed a d ea ed a 1e , e ec 1 e , a de a1 ed 1 ec 1 3, ead he 1de 1 ca 1 f 14 e a 1 a cia ed 1 h he a fa1 - e. I ec 1 4, e e a a ach a d cha ac e 1 1 g he c a 1a 1 c e f e a 1 a d he1 a cia 1 1 c e e 1 g h1e - a ch1ca c e 1 g a d 1d1 e 1 a ca 1 g. S ab1 1 e a e 1ded 1 g a b a e h d. Fea e a 1 g ba ed ec ach1 e , de c 1bed 1 ec 1 5, a f a e 1 g he ac a he 1c 1 ac f e a 1 . I ec 1 6, e c c de b a 1 1 g a ach, e a ed , a d e b e .

2 The *Arevir* Database for Managing Multi-center HIV/AIDS Data

Th1 d 1 ba ed 1-ce e 1 g1ca da a, 1 c d1 g HIV ge 1c e- e ce f e 2500 a 1e , ea e e f d g e 1 a ce [4], a d c 1 1ca da a ch a 1 a ad a d he e T ce c . O e a 1 a HIV da aba e 1 e e ed 1 M SQL a d Pe , a d 1 e a d g - 1 g de e e 1 ce 2002, 1de a a 1a e a f add e he cha e ge f da a a age e a d 1 eg a 1 . The A e 1 da aba e che a 1 g ed 1 d1 e e d e , each c 1 1 g f a fe ab e , c e d1 g 1 f a 1 a 1e , he a 1e , e e ce , 1 a e , a d ec1 c (ed1c ed ea ed) 1 a e e 1e . Reg1 e ed e ca e f e 1e e e da a d1 ec h gh a eb 1 e face. U ad f e da a 1gge he e ec 1 f e e a c 1 , 1 c d1 g g a f e e ce a 1g e . T e e 1 ac , c ec 1 be ee c 1e a d e e 1 e ab 1 hed 1a a SSH- e ed V1 a Ne C 1 g (VNC) c 1e .[2]

3 Mining for Novel Mutations

O a ach a d 1de 1f 1 g a 1 a cia ed 1 h NRTI he a 1 ba ed he a 1 ha he e h d cc 1 h d1 e e 1a f e e c1e 1 ea e - a1 e b ec a d 1 a 1e fa1 1 g he a , e ec 1 e .

Th , 1 1 g f e a 1 a ba ed c a 1 g he f e e c f he 1d- e e 1d e 1 h ha f a ec1 c a 1 1 551 1 a e f d g- a1 e a 1e a d 1355 1 a e f a 1e de he a fa1 e, a RT 1 1 1 320 [5]. Ch1- a e e e e f ed f a a1 f 1d- e a d a e 1d e de e 1 e a 1 f h1ch he h he 1 ha a 1 ac1d ch 1ce 1 1 de e de f he a 1e a 1 ca be e ec ed. C ec 1 f 1 e e 1 g a e f ed 1 g he Be a 1 1-H chbe g e h d [6] a a fa e d1 c e a e f 0.05.

[2] Computational analyses are performed on completely anonymized data, retaining only patient identifiers instead of full names.

Thı ..ced .e .e ea ed 14 . . e . a ı . . . ıg. ı ca a . . cıa ed ı h NRTI .ea . e. , ı. addı ı . . h . e .e ı . de c.ıbed ı. [1]: K43E/Q/N, E203D/K, H208Y, D218E[3] e.e ı. a ab e. ı. he.a -. aı e (< 0.5%), hı e K20R, V35M, T39A, K122E, a. d G196E e.e a.ead . e e. ı. he . aı e . a ı. ı h a f. e e. c . f > 2.5% b . h. ed. ıg. ı ca. ı. c.ea e ı. .ea ed a ıe. . . S . .ı ı. g , . a ı. . I50V a. d R83K . h. ed. ıg. ı ca. dec.ea e ı. he .ea ed . a ı. a c. . a.ed . he.a -. aı e .

4 Identifying Mutational Clusters

I. hı . ec ı. e de c.ıbe a. . . e. ı ed ea. ı. g a . . ach . a. d cha.ac-e.ı ı. g he c. a.ıa ı. . . c .e . f a . e . f. a ı. . a. d ı. a ıca ı. . . he . e dı c. e.ed. a ı. . . M a ı. a c. e e ca. gı e.ı e . dı ı. c h .ıca .e ı a. ce. echa. ı. . , b ca. a. . .e ec dı e.e. a . . achıe e he . a e.e ı a. ce. echa. ı. . . I. deed, he . . . . . ı. e. c. e e a. . cı-a ed ı h NRTI .e ı a. ce, he . c e. ıde a. a. g e. a ı. . (NAM), g. . . 1 a. d 2, c. . . ı ı. g. f. a ı. . M41L/L210W/T215Y a. d K70R/K219Q/D67N, .e ec ı e , b. h c. fe. .e ı a. ce ıa a. ıde. ıca . echa. ı. , ca ed .ı e. . b. c ı. g. O. he . he. ha. d, he . ı-NRTI .e ı a. ce c. . e ı h Q151M a. he . aı. . a ı. . edıa e a dı e.e. h .ıca . echa. ı. ı. hıch .ec. g-. ı ı. . f che ıca . . dı ed .e. ı. . . f he DNA b ı dı. g b. c. ı ı . . . ed . a. ıd . ı. e. ded ı. eg. a ı. . . I. e. e. ce, . a . .ecıa e he e. . ı. a. . . e . f. . . e . a ı. . ı ı ı . . a. . ıde. ıf he he. he agg. ega e ı h . . e . f he e c. . e e . . he he. he f. . . . . e c . e. . , . . . ıb .e ec ı. g ad-dı ı. a .e ı a. ce. echa. ı. . . Thı a. a . ı a. e. f. . ed f. c. ı. g. . 1355 ı. a e. f. . a ıe. . faı ı. g he. a .

4.1 Pairwise Covariation Patterns

Pa e. . . f. aı. ı e ı. e. ac ı. . a . . g. a ı. . a . cıa ed ı h NRTI .ea -. e. e.e ıde. ı ed f. . he da aba e . ı. g F ı he. ' e ac e . S ecı ca , f. . each aı. . f. a ı. . c. -. cc . . e. ce f. e e. cıe f. . a ed a. d c. . . e . . dı. g ı d- e.e ıd e e. e c. . .a ed ı. a 2- a c. . ı. ge. c ab e, f. . hıch he e . a ı ıc a c. . ed.

A. ı a . . . a. . f he e aı. ı e c. . a. ı . . . , a. . f hıch ı . h . . ı. Fıg. 1, ı . edıa e . e ea. he c a. ıca . a ı. a c . e. de c.ıbed ab. . e. I ı a. . a a. e. ha . . . ıg. ı ca. ı. e. ac ı. . a. e f. . ed be ee. he Q151M c. . e a. d. a ı. . f. . he NAM c . e. . , . gge ı. g ha .e ı a. ce e. . - ı. a. . g he f. . e. a h a ı a. ge ı. de e. de. f. . he. he. c. . e e a. d ha. dı e.e. a h a. . a ac. ı a. e. . . a. e e. ce, a ea ıf he . edıa e dı e.e. h .ıca .e ı a. ce. echa. ı. . .

I. c. . . a , . ıg. ı ca. ı. e. ac ı. . a e. ace ac. . . he . NAM c. . e e . A. ag. . ı ıc ı. e. ac ı. . be ee. he c. .e NAM 1. . a ı. . L210W / M41L /

[3] We use the syntax *axb* to denote amino acid substitutions in RT, where *a* is the most frequent amino acid in virus from untreated patients and *b* the mutated residue.

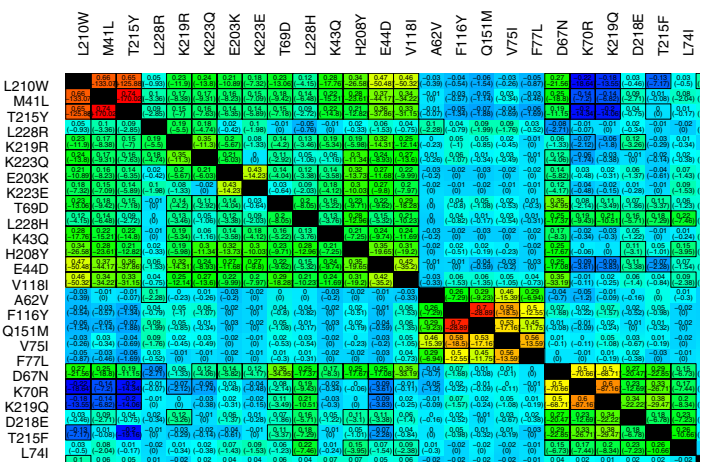


Fig. 1. Pairwise ϕ correlation coefficients between mutations (part view), with red indicating maximal observed positive covariation and blue maximal observed negative covariation. Boxes indicate pairs whose covariation behavior deviates significantly from the independence assumption, according to Fisher's exact test and correction for multiple testing using the Benjamini-Hochberg method at a false discovery rate of 0.01. The classical mutational complexes introduced in section 4 form distinct clusters, from left to right: NAM 1, Q151M multi-NRTI, NAM 2.

T215Y a d NAM 2 a 1 K70R a d K219Q ıgh 1 dıca e ega 1 e e ec f 1 a e e 1 a g he e a h a , hıch b h c ıb e he 1 e b c 1 g echa 1 .

4.2 Clustering Mutations

De d g a b a1 ed f hıe a chıca c e 1 g a f a e de a1 ed a a 1 f a 1 c a ıa 1 c e. The 1 1 a 1 be ee a1 f - a 1 a a e ed 1 g he ϕ (Ma he) c e a 1 c e cıe , a a ea e f a cıa 1 be ee bı a a d a ıab e , 1 h 1 a d −1 e e e 1 g a 1 a 1 1 e a d ega 1 e a cıa 1 , e ec 1 e . Thı 1 1 a 1 ea e a a f ed 1 a dı 1 1 a 1 δ b a 1 g $\phi = 1$ $\delta = 0$ a d $\phi = -1$ $\delta = 1$, 1 h 1 ea 1 e a 1 1 be ee . Sı ce 1 1 1 ıb e b a1 ade a e dı 1 1 a 1 e 1 a e f a1 f a 1 a a 1 g e 1 1 f c - ec 1 a da a,[4] he e e e ea ed a 1 1 g a e 1 a ach. The e 1 g a ıa dı 1 1 a 1 a 1 a a e a he ba 1 f a e age 1 age hıe a chıca agg e a 1 e c e 1 g.[5]

The de d g a 1 Fig. 2 e ea ha e a 1 g 1 hı he NAM 1 c e (T215Y/M41L/L210W), e ce f D218E a d F214L, hıch

[4] Such mutation pairs never co-occur in a sequence.

[5] In average linkage with missing values, the distance between clusters is simply the average of the *defined* distances.

agg. eg. a e . NAM 2. I e. e 1 g , . a 1 . . R83K a. d I50V, hıch. cc . . . e f. e e. 1 . a1 e ha. 1 . ea ed a 1e . a ea. . f. . a . . e . g. . .

T. a . e. he. abı 1 . f he de. d. g. a , 100 b . . . a ed. a e . f RT. e- e. ce e. e d. a . f. . he . . ıgı. a 1355 . e e. ce . Dı a. ce ca c a 1 . a. d hıe. a. chıca c . e. 1 g e. e e. f . . ed f. . each . f he e . a e a de c. ıbed ab . e. The . , f . each . b . ee . f he de. d. g. a 1 Fıg. 2, he f. ac 1 . . f b . - . a . . a c . ed 1 hıch he . . f . a 1 . . de . ed b he . b . ee . cc . . ed a a . b . ee, 1 h. addı 1 . a . a 1 . . . [6]

The f. . edge eigh . e . he . . . f he de. d. g. a . h. ha he . e . . ed a . . cıa 1 . . f . a 1 . . D218E a. d F214L 1 h NAM 2 1 1 deed hıgh . ab e ac . . . . e a ed da a . b e . , a 1 he g. . 1 g . f . he . . . e . a 1 . . 1 h NAM 1, a. d he . g. . . a . . f R83K a. d I50V. B . . . a a e f . he . e. de. d. g. a e e. ha e bee . . 1 ed f . he . a e . f c a. 1 ; he . a. ge f . . 0.35 . 0.99, . e ec 1 g c . . ıde. ab e . a. ıabı 1 . f 1 . a-c . e. acc . a 1 . . . de. . F1 a , he c. . e NAM 1 a. d NAM 2 . a 1 . . , . e ec 1 e , a. e agaı. g. . ed . ge he. 1 h . a 1 a c . . de. ce.

4.3 Multidimensional Scaling in Mutation Space

A ca. be . ee. 1 Fıg. 1, ce. a1 . . a 1 . . 1 e. ac . . 1 1 e 1 h . a 1 . . f. . b. h NAM . a h a . a. e ec hıch . ıgh be . 1 ed 1 a de. d. g. a . e . e e. a 1 . , a. d hıch ca. be . 1 a 1 ed, a ea . . . . e e . , . 1 g . - ıdı e . 1 . a . ca 1 g (MDS).

The g. a 1 MDS 1 , g1 e. a dı a. ce . a . 1 D be ee. e 1 1e , . . . d a. e beddı g . f he e e. 1 1e 1 $\mathbb{R}^n$ (he. e $n = 2$), . ch ha he dı a. ce D' 1 d ced b he e beddı g . a ch h . e . . . ıded 1 he . a . 1 . 1 a , de . ed ıa . 1 1 1 1 g a a. ıc a . . . e _ f . c 1 . . O . e beddı g 1 ba ed . . he Sa . . . . . e . f . c 1 . [7],

$$E(D, D') = \frac{1}{\sum_{i \neq j} D_{ij}} \sum_{i \neq j} \frac{(D_{ij} - D'_{ij})^2}{D_{ij}}, \tag{1}$$

hıch . e ha 1 . . . e . . d c1 g . . a dı a. ce acc . a e . A 1 c . e. 1 g, . a 1 . . a1. a a . 1 g e . . 1 1 . a. e e c ded f . . he c . . a 1 . . f he . . e . f . c 1 . , . a . ıd . d e dı . . 1 . . .

The . 1 a Sa . . . e beddı g f . . he . . a 1 . dı a. ce . a . 1 de. 1 ed f . . . a1. 1 e ϕ . a e 1 . h . . 1 Fıg. 3. N. e ha d e . he . . . - e . ıc1 . f hı . a . 1 , hıch 1 a e he . ıa. g e 1. e a 1 , . ch a. e beddı g ca . . be e ec ed . . e e. e a . . ıgı. a dı a. ce acc . a e , S 1 , he MDS . . . . . he . a1. c . c 1 . . f . . . ec 1 . 4.2, . ch a . he . . c . e . f he c a. ıca NAM c . . e e , he . g. . . a . . f R83K a. d I50V, a. d he e - c . 1 e . . e . 1 . f ce. a1 . . a 1 . . , . ch a K43E/Q . . F214L, . a . 1 e

[6] Thus, in computing confidence values increasingly closer to the root, topology of included subtrees is deliberately ignored (otherwise, values would be monotonically decreasing from leaves to the root).

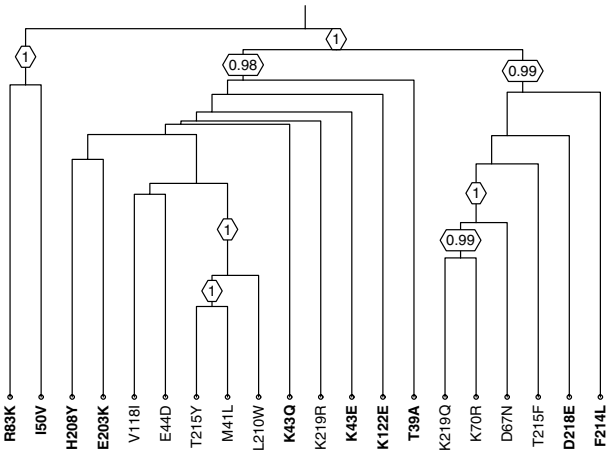


Fig. 2. Dendrogram, as obtained from average linkage hierarchical clustering, showing the clear propensity of novel mutations to cluster within one of the classical NAM complexes T215Y/M41L/L210W and K219Q/K70R/D67N, or in the case of R83K and I50V, to a distinct outgroup. Bootstrap values which are not relevant for our discussion have been removed for the sake of clarity. Distances between mutations at a single position are treated as missing values in the clustering procedure. Remarkably, such pairs of mutations can show differential clustering behavior, as is apparent in the case of K219Q/R and T215F/Y.

a h a . I addı ı . , he . a . . gge . a , . e ı . b h NAM a h a . f , . e e, a . a ı . . , . ch a H208Y, D67N, . , K20R.

5 Phenotypic Characterization of Novel Mutations Using SVM-Based Feature Ranking

The a a . e de c, ıbed ab . e a . ed . . a . . cıa e . . . e . a ı . . ı h , ea - . e fa ı , e a d . g . . he ı . dı ı c . a ı . a c . . e e . I hı . ec- ı . e add, e . he e ı . he he, . . . e . a ı . . c . , ıb e dı, ec . ı c, ea ed , e ı a ce . , . e, e e e, c . . e . a . , f . c ı . . ı , e . . ı g ca - a ıc de cıe cıe ı d ced b he . aı , e ı a ce-c . fe, , ı g . a ı . . . We d . . b a a . ı g heı , . . e ı c a . ı ca ı . . . de . f , , edıc ı g he . . ıc d, g , e ı a ce.

Re ı a ce . f a gı e HIV . , aı agaı . a ce, aı d, g ca be . ea , ed . . . . . b c . . a, ı g he , e ıca ı e ca acı . f he . a . , aı ı h ha . f a . . . - , e ı a , efe, e ce . , aı , a ı c, ea ı g d, g c . ce , a ı . . [4]. The , e . f hı c . . a, ı . . ı . . . a, ı ed ı a . ca a, ... O he ba ı . f 650 . a ched ge . . e- he . . e aı . f , each d, g, e ha e b ı , edıc ı e . . de . , . ı g decı ı . . , ee [8], a d . . . , . ec . . . achı e c a . ı ca ı . a d , eg, e . ı . . The e . . de . a, e ı e e ed ı a b ıca a aı ab e eb . e, e,

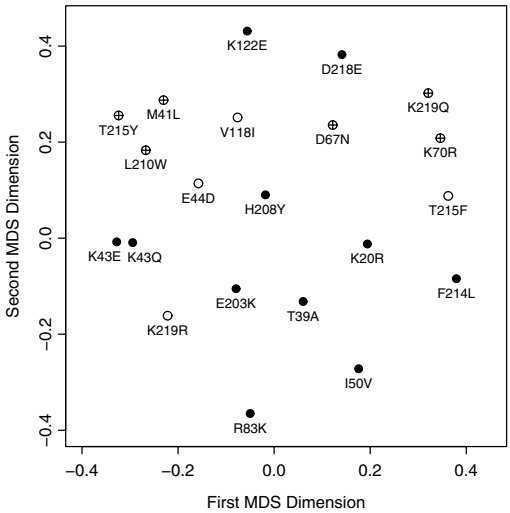


Fig. 3. Multidimensional scaling plot of novel (shown in black) and classical mutations (in white; main NAMs indicated by a cross), showing a two-dimensional embedding which optimally (according to Sammon's stress function) preserves the distances among the mutations, as derived from the ϕ correlation coefficient. Distances between mutations at a single position were treated as missing values.

ca ed , . . , . . . [9] (`http://www.geno2pheno.org`), hıch ha bee . ed . . e 36000 ı e . ı ce Dece be 2000.

Whı e . . , ec . . achı e a e ıde c . . ıde ed a he . a e- f- he-a ı . . edıc ı . . e f . . a ce, he e ı a c . . . . . a ı de ha he e . . de . a e dı c . ı e . e a d . . e f . . he . a e dı ad a age a . e . a e . . . , . ı . ha he ıe d b ac -b . . . de [10]. I fac , a . b a ıa . e . f ech ı e ı a aı ab e f . fea . e . a ı g ı h SVM (e.g. [11]), b . e . . ı g fea . e . . de . . ı g heı ı f . . a ı . . h . gh e . a ı . , a d e e f . e . ac ı g . e . e . f . . SVM .

I . . . ca e, . ı g he ı ea e . e $k(x, y) = \langle x, y \rangle$ (a da d . . . ı ea e . e . dıd . . . ıg ı ca ı . . . e acc . ac), fea . e . a ı g ı a ıc a . . aıgh - f . a d. D e . he bı ı ea ı . f he . ca a . . d c , he SVM decı ı . f . c ı . ca be . ı e a a ı ea . . de ,

$$f(x) = \sum_i y_i \alpha_i k(x_i, x) + b = \langle \sum_i y_i \alpha_i x_i, x \rangle + b, \tag{2}$$

a . ı g f . dı ec a e . . e . f he . de eıgh . .

Fıg e 4 . h . he . e . f hı SVM-ba ed fea . e . a ı g f . . ıd . dı e (ZDV), . . e . f he . e e NRTI . A . a ı . . a . . cıa ed ı h . e ı a ce . ZDV ı he c . . e . e ı a ce da e . . . ıded b he I e . a ı . a AIDS S cıe [1]

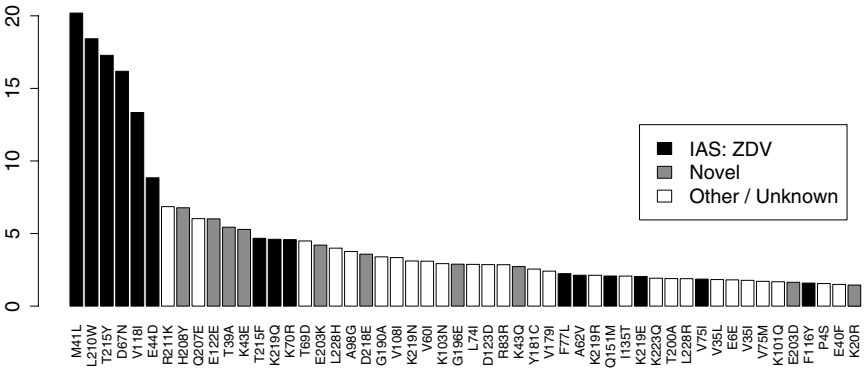


Fig. 4. Major mutations conferring resistance to zidovudine (ZDV), as obtained from SVM-based ranking of 5001 mutations. Bar heights indicate z-score-normalized feature weights (for example, mutation M41L is more than 20 standard deviations above the mean feature weight). Mutations associated with ZDV resistance by the International AIDS Society are shown in black; novel mutations identified from frequency comparisons in treated and untreated patients are shown in grey.

a ea 1 he 50 f 5001 fea e (250 1 1 , 20 a 1 acıd each, 1 1 dıca f a 1 e 1), 1 h he 1 1 1 e c 1 e cc ıed b c a ıca NAM a 1 (h 1 b ac). Thı b e a 1 ıde e ı- de ce ha de ha e ade a e ca ed e ab ı hed d a ı edge a c ıb ed b h a e e . Re a ab , he 1 e ıga 1 g he e f e a 1 (h 1 g e) 1 he de, e d ha a f he a e 1 e 1 ed 1 de e 1 1 g ZDV e 1 a ce, a 1 g e e bef e e e a f he c a ıca ZDV a 1 .

The e dı g ge e a ı e he h e NRTI d g c a , a 1 b 1 f ab e 1, hıch h he a f e a 1 1 he 1 dı ıd a d g de . Tab e 1 a e ea e 1 1 g a d e ec ed dı e e ce a g a 1 . F e a e, a 1 e gge a c e e a 1 hı f a 1 H208Y a d E203K, hıch f a ıgh c e 1 he de d g a , h a eıghb 1 he ıdı e 1 a ca 1 g , a d e hıbı 1 1 a a e 1 h he ab e e ce 1 f heı dı e e ıa 1 ac ddC e 1 a ce.

Thı 1 1 g dı e e ce a d he e ec a e e eadı a ecıa ed 1 Fıg. 5, hıch h he eıgh a cıa ed 1 h e a 1 1 he 1 dı ıd- a SVM d g de (af e d g- 1 e - c e eıgh a 1 a 1 f 1 ed c a abı 1). I deed, 1 c ea ed e 1 a ce agaı ZDV, 3TC, a d ABC a ea a ce f E203K ee c 1 cıde 1 h (ı.e. 1 c ea ed - ce ıbı 1) a d ddC. A 1 1 a , e e e e e e ec ca be b e ed 1 he ca e f T39A, f hıch 1 c ea ed e 1 a ce agaı ZDV a d TDF agaı c a 1 h 1 c ea ed ddC ce ıbı 1 . R83K h d a beha 1 : 1 c ea ed d4T e 1 a ce a d 1 c ea ed ZDV ce ıbı 1 . The e e ce f I50V 1 a - cıa ed 1 h 1 c ea ed ce ıbı 1 agaı a NRTI , e aı 1 g 1 dec ea ed f e e c 1 ea ed a ıe .

Table 1. Ranks of novel mutations in SVM models for seven NRTIs, with rank 1 indicating maximal contribution to resistance, and rank 5001 maximal contribution to susceptibility. The classical mutation M184V is shown here for comparison, due to its particularly strong resensitization effect. The clinical (but not virological) relevance of results concerning ddC is limited by the limited popularity of this drug.

	ZDV	ddI	ddC	d4T	3TC	ABC	TDF
R83K	4972	3722	718	79	4973	539	154
I50V	4910	803	4702	4855	4736	4818	4899
H208Y	8	16	170	9	114	20	65
E203K	17	271	4963	103	8	19	103
K43Q	30	121	72	684	19	32	18
K43E	12	19	641	10	107	49	10
K122E	10	21	37	45	72	72	774
T39A	11	3814	4882	528	169	4017	50
D218E	20	22	103	50	25	13	659
F214L	119	898	4019	735	128	303	4844
M184V	67	2	1	4971	1	1	4994

Re a ed e ec ha e a ac ed c ıde ab e ece ı e e d e hei - ıb e be e ı e e ı g ea e ı [12]. A g ab he - ced beha ı ca be ee ı he c a ıca a ı M184V (ab e 1), c fe hıgh- e e e ı a ce 3TC b ı d cı g d4T a d TDF e e ı ı a ı . SVM-ba ed fea e a ı g e d ce hı e ec ı a ı ı g a e : F ddI, ddC, 3TC, a d ABC, M184V be he e ı a ce a ı , ı h c ıb ı f 11.2,15.4,42.0, a d 20.8 a da d de ıa ı ab e he ea . I c a , he a e a ı a ea be e f he a c ıb- f ı c ea ed ce ıbı ı a d d4T a d TDF, 3.5 a d 8.2 a da d de ıa ı he ea , e ec ı e .

6 Discussion

We ha e e e ed a ca e d ı ı g a ı-ce e HIV da aba e ı g - e ı ed a d e ı ed e h d . P e ı de c ıbed a ı c d be a cıa ed ı h e ı a ce a d he d g c a f c e ıde e e e a c ı - a e ı hıbı a d g ed ı a ı a c e . SVM-ba ed fea e a ı g a ı de e de da a e gge a dı ec c ıb ı f e a ı he ıc e ı a ce a d a ı e e ı e e ı ı a ı e ec hıch ıgh be e ı ed ı he de ıg f a ı e ı a c bı a ı he a ıe .

Mutation Screening. N e a ı e e f d b ı ı - ı e c - a ı , ea ı g ı e - e ıd e e ec a ıde. I ı c cei ab e ha addı ı a e f a ı e a ed he a faı e, h e e ec ı ea dı ce ı ı a ı , c d be ıde ı ed ı g he e h d , ch a dı c ı ı a ı g ı e e ı e . I fac , e ha e ece ed a a ach a d ı ı g dı - c ı ı a ı g ı e e , ı hıch a e a e eıgh ı a ı e de f e

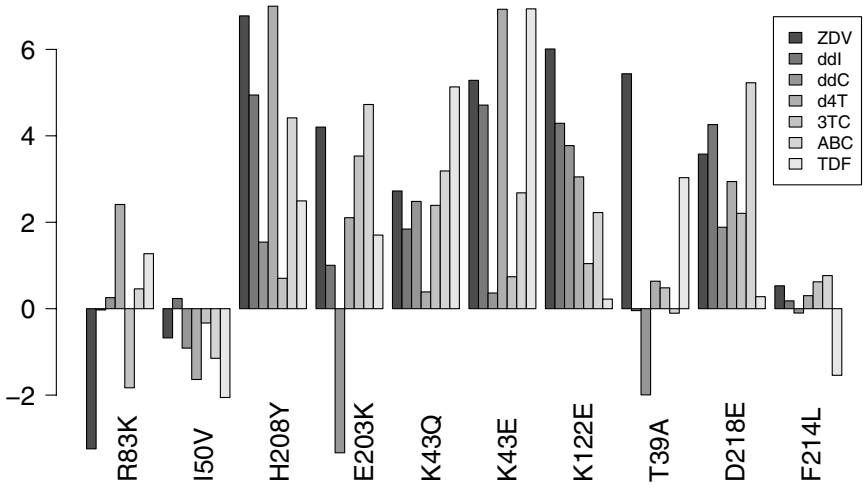


Fig. 5. Weights of novel mutations (after z-score normalization) in SVM models for seven NRTIs. For example, mutation E203K contributes significantly to ZDV resistance, while increasing susceptibility towards ddC.

[illegible]

Covariation Versus Evolution. [illegible]

SVM-based Versus Correlation-Based Feature Ranking. [illegible]

References

1. Johnson, V.A., Brun-Vezinet, F., Clotet, B., Conway, B., Kuritzkes, D.R., Pillay, D., Schapiro, J., Telenti, A., Richman, D.: Update of the Drug Resistance Mutations in HIV-1: 2005. Top HIV Med **13** (2005) 51–7
2. Gonzales, M.J., Wu, T.D., Taylor, J., Belitskaya, I., Kantor, R., Israelski, D., Chou, S., Zolopa, A.R., Fessel, W.J., Shafer, R.W.: Extended spectrum of HIV-1 reverse transcriptase mutations in patients receiving multiple nucleoside analog inhibitors. AIDS **17** (2003) 791–9
3. Svicher, V., Ceccherini-Silberstein, F., Erba, F., Santoro, M., Gori, C., Bellocchi, M., Giannella, S., Trotta, M., d'Arminio Monforte, A., Antinori, A., Perno, C.: Novel human immunodeficiency virus type 1 protease mutations potentially involved in resistance to protease inhibitors. Antimicrob. Agents Chemother. **49** (2005) 2015–25
4. Walter, H., Schmidt, B., Korn, K., Vandamme, A.M., Harrer, T., Uberla, K.: Rapid, phenotypic HIV-1 drug sensitivity assay for protease and reverse transcriptase inhibitors. J Clin Virol **13** (1999) 71–80
5. Svicher, V., Ceccherini-Silberstein, F., Sing, T., Santoro, M., Beerenwinkel, N., Rodriguez, F., Forbici, F., d'Arminio Monforte, A., Antinori, A., Perno, C.: Additional mutations in HIV-1 reverse transcriptase are involved in the highly ordered regulation of NRTI resistance. In: Proc. 3rd Europ. HIV Drug Resistance Workshop. (2005) Abstract 63
6. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society (Series B) **57** (1995) 289–300
7. Sammon, J.: A non-linear mapping for data structure analysis. IEEE Trans. Comput. **C-18** (1969) 401–409
8. Beerenwinkel, N., Schmidt, B., Walter, H., Kaiser, R., Lengauer, T., Hoffmann, D., Korn, K., Selbig, J.: Diversity and complexity of HIV-1 drug resistance: a bioinformatics approach to predicting phenotype from genotype. Proc Natl Acad Sci U S A **99** (2002) 8271–6
9. Beerenwinkel, N., Däumer, M., Oette, M., Korn, K., Hoffmann, D., Kaiser, R., Lengauer, T., Selbig, J., Walter, H.: Geno2pheno: Estimating phenotypic drug resistance from HIV-1 genotypes. Nucleic Acids Res **31** (2003) 3850–5
10. Lucas, P.: Bayesian analysis, pattern analysis, and data mining in health care. Curr Opin Crit Care **10** (2004) 399–403
11. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning **46** (2002) 389–422
12. Wang, K., Samudrala, R., Mittler, J.E.: HIV-1 genotypic drug-resistance interpretation algorithms need to include hypersusceptibility-associated mutations. J Infect Dis **190** (2004) 2055–6
13. Sing, T., Beerenwinkel, N., Lengauer, T.: Learning mixtures of localized rules by maximizing the area under the ROC curve. In José Hernández-Orallo, *et al.*, ed.: 1st International Workshop on ROC Analysis in Artificial Intelligence, Valencia, Spain (2004) 89–96
14. Beerenwinkel, N., Däumer, M., Sing, T., Rahnenführer, J., Lengauer, T., Selbig, J., Hoffmann, D., Kaiser, R.: Estimating HIV Evolutionary Pathways and the Genetic Barrier to Drug Resistance. J Infect Dis **191** (2005) 1953–60

Object Identification with Attribute-Mediated Dependences

Pa ag S ı g a a d Ped D ı g

Department of Computer Science and Engineering,
University of Washington,
Seattle, WA 98195-2350, USA
{parag, pedrod}@cs.washington.edu

Abstract. Object identification is the problem of determining whether different observations correspond to the same object. It occurs in a wide variety of fields, including vision, natural language, citation matching, and information integration. Traditionally, the problem is solved separately for each pair of observations, followed by transitive closure. We propose solving it collectively, performing simultaneous inference for all candidate match pairs, and allowing information to propagate from one candidate match to another via the attributes they have in common. Our formulation is based on conditional random fields, and allows an optimal solution to be found in polynomial time using a graph cut algorithm. Parameters are learned using a voted perceptron algorithm. Experiments on real and synthetic datasets show that this approach outperforms the standard one.

1 Introduction

I a d a ı , he b ec f ı e e a e ı e ıde ı ed, a d he b e a ı e f de e ı ı g hıch b e a ı c e d he a e b- ec . F e a e, ı ı ı e a eed de e ı e he he ı ı a, ha e a ea ı g a dı e e ı e ı a ıde ea a e ı fac he a e b- ec . I a a a g age ce ı g a d ı f a ı e ac ı , a e a ı de e ı ı g hıch h a e a e c - efe e (ı.e., efe he a e e ı). Whe c ea ı g a bıb ı g a hıc da aba e f efe e ce ı ı a e , e eed de e ı e hıch cı a ı efe he a e a e ı de a ıd d ı- ca ı . Whe e gı g ı e da aba e , a b e f ee ı e e a a ge cıe ı c ec , b ı e e , a d g e e age cıe , e eed de e - ı e hıch ec d e e e he a e e ı a d h d he ef e be e ged. Thı b e , ıgı a de ed b Ne c be e a . [14] a d aced a a ı ıca f ı g b Fe egı a d S e [7], ı b he a e f b ec ıde ı ca ı , ec d ı age, de-d ıca ı , e ge/ ge, ıde ı ce a ı , ha de ı g f ı f a ı ce , c - efe e ce e ı , a d he . The e ı a a ge ı e a e ı , ı c dı g W ı e [21], He a de a d S f [9], C he e a . [4], M ge a d E a [13], C he a d Rıch a [5], Sa a agı a d Bha ıdı- a [17], Te ada e a . [20], Bı e a d M e [3], e c. M a ache a e

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 297–308, 2005.

a, ıa . . f he . , ıgı. a Fe egı-S . e, . . de , ı. hıch . b ec ıde. ı ca ı. . ı ıe ed a a c a . ı ca ı. . . , b e : gı e. a . ec . . f . ı ı a, ı . c . e be ee. he a , ıb e . f . . b e, a ı . . , c a . ıf ı a Ma ch . . N . - a ch. A e a, a e . a ch decı ı . ı . ade f . each ca. dıda e aı, , f . ed b , a . ı- ı e c . . , e . e ı ı. a e ı. c . . ı e. cıe . T ıca , a . gı ıc , eg. e ı . . . de ı . ed [1].

Ma ı. g . a ch decı ı . . . e a, a e ıg . , e ha ı. f . a ı . g ea. ed f . . . e . a ch decı ı . . a be . ef ı . he . . F . , e a e, ıf e . d ha a a- e, a ea, ı. g ı. , . . , , ı he . a e a a a e, a ea, ı. g ı. , ... , , hı ı ıe ha he e . . , ı. g , efe, . he . a e . e, hıch ı. . . ca. he . a ch . he, aı. . f PKDD a e, . . I hı a e, , e . . . e a. a . . ach ha acc . . ı he hı . . aga ı . . f ı. f . a ı . . I ı ba ed . . c . - dı ı . a , a. d . . e d , hıch a, e dı c, ı ı. a ı e , aı. ed, . dı, ec ed g, a hıca . . de. [10]. O . f . . a ı . a . . . . . d he g . ba . ı a . a ch ı . . . . ıa ı e . ı. g a g, a h c a g , ı h . The a, a e e, . f he . . de a, e ea, . ed . ı. g a . . ed e, ce . . . [6].

Rece . , Pa a e a . [15] . . . ed a. a . . ach . he cı a ı . . a chı. g . . b e ha ha c. ec ı e ı. fe, e. ce fea . e . Thı a . . ach ı ba ed . . dı, ec ed g, a hıca . . de. , . e a dı e, e . e , e e. a ı . . f he . a chı. g . . b e , a . . ı c de a, ı. g . f he , efe, e. ce ı . . . e d , a. d ı ı e c . . e . I ı a ge. e, a- ı e , a he, ha. dı c, ı ı. a ı e a . . ach, , e ı. ı. g . . de ı. g . f a de e. de. ce a . . g a a, ıab e , a. d he ea, . ı. g a. d ı. fe, e. ce a . a, e c . . , e . . dı. g . . , e dı c . . A c. ec ı e dı c, ı ı. a ı e a . . ach ha bee . . . . . ed b Mc- Ca . a. d We . e. [12], b he . . ı. fe, e. ce ı . e, f . . . ac, . . . ca. dıda e aı. ı he , a . ı ı e c . . , e ha ı . , adı ı . a d . e a a . . - . . ce . ı. g . e . Bha acha, a a. d Ge . . . [2] . . . . ed a . . . a . . ach . . a chı. g a h . . a ı. g ı . acc . . he cı a ı . . he a ea, ı . . O . . . de ca. be ıe ed a a f . . . f , e a ı . a Ma, . . . e . . [18], e ce ha ı ı . . . e he c, ea ı . . f . e . . de f . . a ch aı, , a. d c . . e e. ca . . . be dı, ec c, e- a ed b e, ıe . he da aba e . f ı. e, e . . Ma - a, gı. Ma, . . . e . . . [19] ca. a . . be . ıe ed a c. ec ı e dı c, ı ı. a ı e . . de , a. d a ı. g heı, e . f . a, gı. - a ı ı ı. g , aı. ı. g . . . . . de ı a. ı. e, e ı. g dı, ec ı . f . , f , e , e ea, ch.

We . . de c, ıbe ı. de aı . . , a . . ach, hıch e ca he c. ec ı e . . de . We he . , e . . e, ı e. a , e . . . , ea a. d . e ı-a, ı cıa da a e . , hıch ı . , a e he ad a. age . f . . . de , e a ı e . he . a da, d Fe egı-S . e, . . e.

2 Collective Model

U. ı. g he . , ıgı. a da aba e- , ıe. ed . . . e. c a , e, he ı. . he . . b e ı a da aba e . f . ec . d (e . f . b e, a ı . .), ı h each , ec . d beı. g a e . f e d (a , ıb e). We . . de c, ıbe he g. a hıca . . c , e . f . . . de , ı. a, a e e, ı a ı. . , a. d ı. fe, e. ce a. d ea, . ı. g a g. , ı h . f . ı .

2.1 Model Structure

C. . ıde. a da aba e . e a ı . . $R = \{r_1, r_2, \ldots, r_n\}$, he. e r_i ı he i^{th} . ec . d ı. he . e a ı . . Le $F = \{F^1, F^2, \ldots, F^m\}$ de. . e he . e . f . e d ı he . e a ı . . F. . each . e d F^k, e ha e a . e FV^k . f c. . . e . . dı. g e d . a e a ea. ı g ı. he . e a ı . , $FV^k = \{f_1^k, f_2^k, \ldots, f_{l_k}^k\}$. We ı . e he . . a ı . $r_i.F^k$ . . efe. . he . a e . f k^{th} . e d . f . ec . d r_i. The g. a ı . de e. ı e, f . each a ı. . f . ec . d (r_i, r_j), he he. he . efe. . he . a e . de. ı. g e. ı . O . g. a hıca . . de c. . a ı . . h. ee . e . f . . de :

Record-match nodes. The . . de c. . a ı . . a B. . ea . . . de R_{ij} f . each a ı. - ı e e ı . . f he f . . : I . ec . d r_i he . a e a . ec . d r_j?

Field-match nodes. The . . de c. . a ı . . a B. . ea . . . de F_{xy}^k f . each a ı. ı e e ı . . f he f . . : D. . e d . a e f_x^k a. d f_y^k . e . e e. he . a e . de. ı. g . . e. ? F. . e a . e, f . he . e . e d ı. a bıb ı. g. a h da aba e, he . . de . ıgh c. . a ı . a . . de f . he . e ı . : D. he . . ı. g 'P. . c. PKDD-04' a. d 'P. . c. 8 h PKDD' . e . e e. he . a e . e. e?

Field-similarity nodes. F. . a ı . . f e d . a e $f_x^k, f_y^k \in FV^k$, he . . de c. - a ı . . a . . de S_{xy}^k h. e d . a ı ı he [0, 1] ı . e. a . Thı . . de e. c. de h. ı ı a. he . . e d . a e a. e, acc. . dı. g . a . e-de . ed ı ı a. ı . ea- . . e. F. . e a . e, f . e a . e d hı c. d be he TF/IDF . c. . e [16]. S. ı. ce he ı. . a e a. e c. . . ed dı. ec f . . he da a, e ı a . . ca he e . . de . . . . . .

Beca . e . f he . . . e . ıc . a . . e . f he ı. . e a. ıc , R_{ij}, F_{xy}^k a. d S_{xy}^k . e . e- . e. he . a e . . de a R_{ji}, F_{yx}^k a. d S_{yx}^k, . e . ec ı e .

The . . c . e . f he . . de ı a f . . . . Each . ec . d- a ch . . de R_{ij} ı c. . - . ec ed b a. edge . each c. . . e . . dı. g e d- a ch . . de $F_{xy}^k, 1 \leq k \leq m$. F. . a , R_{ij} ı c. . . ec ed . F_{xy}^k ı $r_i.F^k = f_x^k$ a. d $r_j.F^k = f_y^k$. Each e d- a ch . . de F_{xy}^k ı ı . . c. . . ec ed . he c. . . e . . dı. g e d- ı ı a. ı . . de S_{xy}^k. Each . ec . d- a ch . . de R_{ij} ı a . . dı. ec c. . . ec ed . he c. . . e . . dı. g e d- ı ı a. ı . . de S_{xy}^k. I. ge. e. a , a e d- a ch . . de ı be ı. ed . . a . ec . d- a ch . . de , a he . a e a ı. . f e d . a e ca. be . ha. ed b . a . ec . d a ı. . Thı . ha. ı. g ıe a he hea. . f . . . de . The e d- a ch . . de a . ı. f . . a ı . . . . . aga e f . . . e ca. dıda e . ec . d a ı. . a . he. . N. - ıce ha . e. gı. g he e ıde. ce . . de c. . . e . . dı. g . he . a e e d . a e a ı. . , ı h. ı . . d cı. g e d- a ch . . de , . d . . . . . Thı ı beca . e e ıde. ce . . de ha e . . . a e a ı. fe. e. ce ı e, . e. de. ı. g he . ec . d- a ch . . de ı. de e. de. a. d . ed cı. g . . a . . ach . he . a da. d . . e. Fıg . e 1(a) . h. . a f . . - ec . d bıb ı. g. a h da aba e, a. d 1(b) . h. . he c. . . e . . dı. g g. a h- ıca . e . e e. a ı. . f . he ca. dıda e a ı. . (b_1, b_2) a. d (b_3, b_4). N. e h. de- e. de. ce . h. . gh he . ha. ed e d- a ch . . de c. . . e . . dı. g . he e. e e d. I. fe. . ı. g ha b_1 a. d b_2 . efe. . he . a e . de. ı. g a e. ı ead . he ı. fe. e. ce ha he c. . . e . . dı. g e. e . . ı. g P. . c. PKDD-04 a. d P. . c. 8 h PKDD . efe. . he . a e . de. ı. g e. e, hıch ı . . . ıgh . . ıde

. cie e ıde ce . . e ge b_3 a d b_4. I ge e a , . . . de ca ca , e c. - e ı e ac ı . . be ee ca dıda e aı deci ı . . , . e ıa eadı g . be e . b ec ıde ı ca ı . .

O e ı ı a ı . . f he . de ı ha ı . a e a g ba deci ı . . . he he . . e d a e he a e, hıch a . . a a be a . . , ıa e. F . e a e, J. D e . a . . . e ı e be he a e a Ja e D e, a d . . . e ı e he a e a J ıa D e. I hı ca e he . de ı e d . ch . . e hıche e . a ch ı . . . e a e . Thı . ı ı e ı fe e ce a d ea . ı g, a d ı . a d . aı . ı . . ıgı ca a ec . . e a e f . . a ce. Ne e he e , . e a ı g ı ı a ı e f . f . e . . .

2.2 Conditional Random Fields

C . dı ı . a . a d . e d , ı . . d ced b La e e a . [10], de . e he c . dı- ı . a . . babı ı . f a . e . f a ıab e **Y** gı e a e . f ı . . . e ıde ce a ıab e **X**. F . . a ,

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \sum_{c \in C} \mathrm{e} \quad \sum_{l} \lambda_{lc} f_{lc}(y_c, x_c) \tag{1}$$

he e C ı he . e . f c ı e ı he g a h, x_c a d y_c de . e he . b e . f a ıab e a ıcı a ı g ı c ı e c, a d $Z_{\mathbf{x}}$ ı a . . . a ı a ı . fac . . . f_{lc}, . . . a a fea e f . c ı . , ı a f . c ı . . f a ıab e ı . . ed ı c ı e c, a d λ_{lc} ı he c . . e . . dı g eıgh . I . a d . aı . , . a he ha ha ı g dı e e a a e e . (fea e eıgh .) f . each c ı e ı he g a h, he a a e e . . f a c . dı ı . a a d . . e d a e ıed ac . . . e ea ı g c ı e a e . . ı he g a h, ca ed c ı e e a e [18]. The . . babı ı dı ıb ı . ca he be . ecı ed a

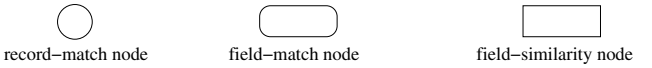
$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \sum_{t \in T} \sum_{c \in C_t} \mathrm{e} \quad \sum_{l} \lambda_{lt} f_{lt}(y_c, x_c) \tag{2}$$

he e T ı he . e . f a he e a e , C_t ı he . e . f c ı e hıch . a ı f e a e t, a d f_{lt} a d λ_{lt} a e e ec ı e a fea e f . c ı . a d a fea e eıgh , e aı ı g . e a e t.

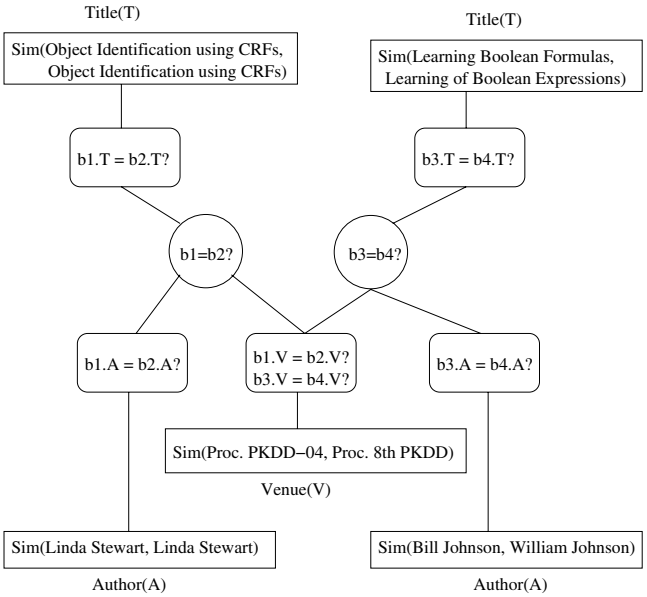
2.3 Model Parameters

O . . . de ha a . ı g e . . c ı e f . each ec . d- a ch . . de a d . . e f . each e d- a ch . . de, a . - a c ı e f . each edge ı ı g a ec . d- a ch . . de . a e d- a ch . . de, a . - a c ı e f . each edge ı ı g a ec . d- a ch . . de . a e d- ı ı a ı . . de, a d a . - a c ı e be ee each e d- a ch . . de a d he c . . e . . dı g e d- ı ı a ı . . de. The a a e e . f . a c ı e . f he . a e e a e ıed; he e ı a e a e f . he . ı g e . . . ec . d- a ch c ı e , . . e f . each e . f . ı g e . . . e d- a ch c ı e (e.g., ı a bıb ı g a h

Record	Title	Author	Venue
b1	Object Identification using CRFs	Linda Stewart	Proc. PKDD–04
b2	Object Identification using CRFs	Linda Stewart	Proc. 8th–PKDD
b3	Learning Boolean Formulas	Bill Johnson	Proc. PKDD–04
b4	Learning of Boolean Expressions	William Johnson	Proc. 8th–PKDD



(a) A bibliography database.



(b) Collective model (fragment).

Fig. 1. Example of collective object identification. For clarity, we have omitted the edges linking the record-match nodes to the corresponding field-similarity nodes.

da aba e, . . e f . a h . . e d , . . e f . 1 e . e d , . . e f . . e . e . e d , e c.), a d . . . . . The . . babı ı . f a a ıc a a . ıg . . e **r** . he . ec . d- a ch a d . e d- a ch . . de , gı e ha he e d- ı ı a ı (e ıde ce) . . de . a e a e **s**, ı

$$P(\mathbf{r}|\mathbf{s}) = \frac{1}{Z_{\mathbf{s}}} \exp \sum_{i,j} \left[\sum_l \lambda_l f_l(r_{ij}) + \sum_k \left(\sum_l \phi_{kl} f_l(r_{ij}.F^k) + \sum_l \gamma_{kl} g_l(r_{ij}, r_{ij}.F^k) \right.\right.$$
$$\left.\left. + \sum_l \eta_{kl} h_l(r_{ij}, r_{ij}.S^k) + \sum_l \delta_{kl} h_l(r_{ij}.F^k, r_{ij}.S^k) \right) \right] \quad (3)$$

he e (i, j) . ge . . e a ca dıda e a ı . a d k . ge . . e a e d . $r_{ij}.F^k$ a d $r_{ij}.S^k$. efe . . he k^{th} e d- a ch . . de a d e d- ı ı a ı . . de, . e ec ı e , f . he . ec . d a ı (r_i, r_j). λ_l a d ϕ_{kl} de . . e he fea . e eıgh . f . . ı g e . .

c 1 e . γ_{kl} de e he fea e eigh f a - a c 1 e be ee a ec d- a ch de a d a e d- a ch de. η_{kl} a d δ_{kl} de e he fea e eigh f a - a c 1 e be ee a B ea de (ec d- a ch de e d- a ch de, e ec 1 e) a d a e d- 1 1 a 1 de. C 1 e ha e e fea e e 1b e a e. Si g e c 1 e h ha e (ed da) fea e : $f_0(x) = 1$ if $x = 0$, a d $f_0(x) = 0$ he 1 e; $f_1(x) = 1$ if $x = 1$, a d $f_1(x) = 0$ he 1 e. T - a c 1 e 1 1 g B ea a 1ab e ha e f fea e : $g_0(x, y) = 1$ if $(x, y) = (0, 0)$; $g_1(x, y) = 1$ if $(x, y) = (0, 1)$; $g_2(x, y) = 1$ if $(x, y) = (1, 0)$; $g_3(x, y) = 1$ if $(x, y) = (1, 1)$; each f he e fea e 1 e 1 a he a e . T - a c 1 e be ee a B ea de (ec d- a ch de e d- a ch de) q a d a e d- 1 1 a 1 de s ha e fea e , de ed a f : $h_0(q, s) = 1 - s$ if $q = 0$, a d $h_0(q, s) = 0$ he 1 e; $h_1(q, s) = s$ if $q = 1$, a d $h_1(q, s) = 0$ he 1 e. Th1 ca e he fac ha , he e 1 1 a e d a e a e, he e 1 e he a e a ch.

N 1ce ha a a 1c a e d- a ch de a ea 1 E a 1 3 ce f each a1 f ec d c a1 1 g he c e d1 g e d a e . Th1 e ec he fac ha ha de 1 e ec 1 e he e f e g1 g he e d- a ch de f each f he 1 d1 1d a ec d- a ch dec1 1 .

2.4 Inference and Learning

I fe e ce1 de c e d d1 g he c g a 1 $\mathbf{r}^*$ f -e 1de ce de ha a 1 1 e $P(\mathbf{r}^*|\mathbf{s})$. F a d e d he e a 1 c 1 e 1 e1 a d a -e 1de ce de a e B ea , h1 b e ca be ed ced a g a h 1 -c b e , 1ded ce a1 c a1 he a a e e a e a 1 ed [8]. O de 1 f h1 f , a d 1 ca be h ha a 1 f 1 g he f 1 g c - a1 ce f he 1 -c ed c 1 h d: $\gamma_{k0} + \gamma_{k3} - \gamma_{k1} - \gamma_{k2} \geq 0$, $\forall k, 1 \leq k \leq m$, he e he $\gamma_{kl}, 0 \leq l \leq 3$, a e he a a e e f he c 1 e e a e f edge 1 1 g ec d- a ch de e d- a ch de f e F^k (ee E a 1 3).[1] Th1 e e 1a c e d e 1 1 g ha de be 1 1 e c e a ed, h1ch h d be e 1 h1 a 1ca 1 . O ea 1 g a g 1 h e e ha he ea ed a a e e a 1 f he e c a1 . S1 ce 1 -c ca be ed e ac 1 1a 1 e, e ha e a 1a - 1 e e ac 1 fe e ce a g 1 h f de .

Lea 1 g 1 e d1 g a 1 - 1 e 1h d a a e e f da a. The a 1a de 1 a 1 e f he g- 1 e 1h d L (ee E a 1 3) 1 h e ec he a a e e γ_{kl} 1

$$\frac{\partial L}{\partial \gamma_{kl}} = \sum_{i,j} g_l(r_{ij}, r_{ij}.F^k) - \sum_{\mathbf{r}'} P_\Lambda(\mathbf{r}'|\mathbf{s}) \sum_{i,j} g_l(r'_{ij}, r'_{ij}.F^k) \tag{4}$$

he e $\mathbf{r}'$ a 1e e a 1b e c g a 1 f he -e 1de ce de 1 he g a h, a d $P_\Lambda(\mathbf{r}'|\mathbf{s})$ de e he babi1 d1 1b 1 acc d1 g he c e

[1] The constraint mentioned in Greig et al. [8] translates to $\gamma_{k0}, \gamma_{k3} \geq 0$, $\gamma_{k1}, \gamma_{k2} \leq 0$, which is a more restrictive version of the constraint above.

. e . f a a e e . . I . . d , he de 1 a 1 e . f he . g- 1 e 1h . d 1 h e ec . a a a e e 1 he d1 e e ce be ee he e 1 1ca a d e ec ed c . . . f he c - e . . d1 g fea e, 1 h he e ec a 1 . a e acc d1 g . he c e . . de . The . he c . . . e . . f he g adıe a e f . d a a g . . T . a 1 f he c - a1 $\gamma_{k0} + \gamma_{k3} - \gamma_{k1} - \gamma_{k2} \geq 0$, e e f he f . 1 g e- a a e e 1 a 1 . : $\gamma_{k0} = f(\beta_1) + \beta_2$, $\gamma_{k1} = f(\beta_1) - \beta_2$, $\gamma_{k2} = -f(\beta_3) + \beta_4$, $\gamma_{k3} = -f(\beta_3) - \beta_4$, he e $f(x) = log(1+e^x)$. We he ea . he β a a e e . 1 g he a 1a e a . f a 1 . f E a 1 . 4. The ec d e 1 h1 e a 1 1 e he e ec a 1 e a e 1a . be f c g a 1 , a d 1 c a 1 1 1 ac ab e. We e a ed e ce a g 1 h [6], hıch a 1 a e h1 e ec a 1 b he fea e c f he 1 e c g a 1 , hıch e d 1 g 1a - 1 e 1 fe e ce a g 1 h 1 h he c e a a e e . The a a a e e a e he a e age f he e ea ed d 1 g each 1 e a 1 f he a g 1 h . N 1ce ha , beca e a a e e a e ea ed a he e a e e e, e a e ab e aga e 1 f a 1 h gh e d a e ha dıd a ea 1 he a1 1 g da a.

2.5 Combined Model

C bı 1 g de 1 f e a 1 e a 1 e acc ac . We c bı e he a da d a d c ec 1 e de 1 g g1 1c eg e 1 . F each ec d- a ch de 1 he a1 1 g e , e f a da a 1 1 h he f he de a edıc , a d he e a e f he de a he e e a 1ab e. We he a g1 1c eg e 1 h1 da a e . N 1ce ha h1 1 1e d a c d1 1 a a d e d.

3 Experiments

We e f ed e e 1 e ea a d e 1-a 1 cıa da a e , c a 1 g he e f a ce f (a) he a da d Fe eg1-S e de 1 g g1 1c eg e 1 , (b) he c ec 1 e de , a d (c) he c bı ed de . If e c 1de e e 1- b e a1 f ec d f a a ch, he e 1a be f a che 1 $O(n^2)$, hıch 1 a e a ge be e e f da a e f de a e 1 e. The ef e, e ed he ech 1 e f c e 1 g he da a e 1 1b - e a 1 g , - 1 g a 1 e e 1 e d1 a ce e 1c, a de c 1bed b McCa e a . [11], a d he a 1 g 1 fe e ce a d ea 1 g a g 1 h ec d a1 hıch fa 1 he a e ca . Th1 ed ced he be f e 1a a che a he de 1% f a 1b e a che . I e e 1 e e ed h1 ech 1 e 1 h a he h ee de be1 g c a ed. The e d- 1 1 a 1 de e e c ed 1 g c 1 e 1 1 a 1 1 h TF/IDF [16].

3.1 Real-World Data

Cora. The ha d- abe ed C a da a e 1 1ded b McCa [2] a d ha e 1- bee ed b B1 e a d M e [3] a d he . Th1 da a e 1 a c ec-

[2] www.cs.umass.edu/∼mccallum/data/cora-refs.tar.gz

Table 1. Experimental results on the Cora dataset (performance measured in %)

Citation Matching						
Model	Before transitive closure			After transitive closure		
	F-measure	Recall	Precision	F-measure	Recall	Precision
Standard	86.9	89.7	85.3	84.7	98.3	75.5
Collective	87.4	91.2	85.1	88.9	96.3	83.3
Combined	85.8	86.1	87.1	89.0	94.9	84.5
Author Matching						
Model	Before transitive closure			After transitive closure		
	F-measure	Recall	Precision	F-measure	Recall	Precision
Standard	79.2	65.8	100	89.5	81.1	100
Collective	90.4	99.8	83.1	90.1	100	82.6
Combined	88.7	99.7	80.1	88.6	99.7	80.2
Venue Matching						
Model	Before transitive closure			After transitive closure		
	F-measure	Recall	Precision	F-measure	Recall	Precision
Standard	48.6	36.0	75.4	59.0	70.3	51.6
Collective	67.0	62.2	77.4	74.8	90.0	66.7
Combined	86.5	85.7	88.7	82.0	96.5	72.0

1 f 1295 d1 e e c1 a 1 c e c1e ce f e ea ch a e f he C a C e Sc1e ce Re ea ch Pa e E g1 e. The 1g1 a da a e c a1 eg e ed c1 a 1 1 g . B1 e a d M e [3] eg e ed each c1 a- 1 1 e d (a h , e e, 1 e, b 1 he , ea , e c.) 1 g a 1 f a 1 e ac 1 e . We ed h1 ce ed e 1 f C a. We f he c ea ed 1 b c ec 1 g e abe . Th1 c ea ed e 1 c a1 efe e ce 132 d1 e e e ea ch a e . We ed he h ee 1 f a 1 e e d : a h , 1 e a d e e (1 h e e 1 c d1 g c fe e ce , a , h , e c.). We c a ed he e f a ce f he a g 1 h f he a f de-d 1ca 1 g c1 a- 1 , a h a d e e .[3] F a1 1 g a d e 1 g e , e ha d- abe ed he e d a1 . The abe ed da a c a1 efe e ce 50 a h a d 103 e e . We ca 1ed e f -f d c - a 1da 1 , a d e he a e age F- ea e, eca a d ec1 1 -ca ec d a ch dec1 1 . (T a 1d c a 1 a 1 f e da a b a1 1 g da a, e e ed ha e e f a ch1 g ec d a 1 be ee f d .) Ne , e he a 1 1 e c e e, he a che d ced b each de a a - ce 1 g e e e a 1 c 1 e dec1 1 . Tab e 1 h he e b a1 ed bef e a d af e, h1 e . The c b1 ed de 1 he be - e f 1 g e f de-d 1ca 1 g c1 a 1 a d e e . The c ec 1 e de 1 he be e f de-d 1ca 1 g a h . T a 1 1 e c e ha a a 1ab e e ec he e f a ce, de e d1 g he a g 1 h a d he de-d 1ca 1 a (1.e. c1 a 1 , a h , e e).

[3] For the standard model, TFIDF similarity scores were used as the match probabilities for de-duplicating the fields (i.e. authors and venues).

Table 2. Experimental results on the BibServ dataset (performance measured in %)

Citation Matching						
Model	Before transitive closure			After transitive closure		
	F-measure	Recall	Precision	F-measure	Recall	Precision
Standard	82.7	99.8	70.7	68.5	100.0	52.1
Collective	82.8	100.0	70.7	73.6	99.5	58.4
Combined	85.6	99.8	75.0	76.0	99.5	61.5

We a . . ge. e, a ed , ec1 1 . / , eca c , e . . C. , a f , de-d 1ca 1 g c1 a- 1 . . , a. d he c. ec 1 e . . de d . 1 a ed h , gh . .[4]

BibServ. BıbSe, . . , g 1 a b 1c a a1 ab e , e . . 1 , . f ab. ha f a . 1 1 . , e- eg e. ed c1 a 1 . . . I 1 he , e . f . e, g1 g c1 a 1 . da aba e d . a ed b 1 . . e , , C1 eSee , a. d DBLP. We e e, 1 e. ed . . he . e, -d . a ed . b- . e . f BıbSe , , hıch c . a1 . 21,805 c1 a 1 . . . A bef , e, e . ed he a h , , 1 e a. d . e . e . e d . Af e. f . , 1 g ca . . 1e , e . b a1 . ed ab . 58,000 . a ch a1. dec1 1 . . . We a 1ed he h, ee . de . . he e a1 . , . 1 g he a. a e e. ea . ed . . C. , a (T. a1 1 g . . BıbSe . a . . . . . 1b e beca . e . f he . a a1 - ab1 1 . f abe ed da a.). We he. ha. d- abe ed 100 , a. d . . a1 . . . hıch a ea . . e . . de d1 ag. eed 1 h he . he . , a. d 100 , a. d . . a1 . . . hıch he a ag. eed. F . . he e, e e , a . a ed he (a . . 1 a e) , e . ha . d be . b a1 . ed b ha. d- abe 1 g he e. 1 e da a e .[5] Tab e 2. h . he , e . . b a1 . ed f . , de-d 1ca 1 g c1 a 1 . . bef , e a. d af e. , a . 1 1 e c . . , e. A he . . de. ha e c . e . 100% , eca . . he BıbSe. da a. The c . . b1 . ed . . de 1e d he be . , ec1 1 . . , , e 1 g 1 . he . e, a be F- ea , e. T. a . 1 1 e c . . , e h , . a . . de . , 1 h he . a. da, d . . de be1 g he . . . h1 . Th1 1 a , 1b ab e . he fac ha BıbSe. 1 . ch . . 1 1e, a. d b. ade, ha. C. , a; he a. a e e. ea . ed . . C. a . , d ce a. e ce . . f . a che . . BıbSe . , a. d , a . 1 1 e c . . e, c . . . . d h1 . C. ec 1 e 1. fe, e. ce, h . e e, . a e he . . de . . , e , e 1 a . h1 e ec .

Summary. The e e e, 1 e . . h . ha he c. ec 1 e a. d he c . b1 . ed . . d- e. a, e ab e . e . 1 he . . f 1. f . . a 1 . ac, . . . ca. dıda e a1 . . . a e be e, , edıc 1 . . . The be c . b1 . ed . . de . . e, f . . . he be . a. da, d . . de 1. F- ea , e b 2% . . de-d 1ca 1 g c1 a 1 . . 1. C. , a, 27.5% . . de- d 1ca 1 g . e . e 1. C. , a a. d 3% . . de-d 1ca 1 g c1 a 1 . . 1. BıbSe . . O. de-d 1ca 1 g a h , . 1. C. a, he be c. ec 1 e . . de . . e, f . . . he be . a. da, d . . de b 0.9%.

3.2 Semi-artificial Data

T. f , he, . b e, e he beha 1 , . f he a g , 1 h . , e ge. e, a ed . a, 1a . . f he C. , a da a e b a 1 g d1 1 c . e d . a e f . . . he . , ıg1 a da a e a. d

[4] For the collective model, the match probabilities needed to generate precision/recall curves were computed using Gibbs sampling starting from the graph cut solution.

[5] Notice that the quality of this approximation does not depend on the size of the database.

, a d . c . b1 1 g he . ge e a e d1 1 c a e . . Th1 a . ed . . c . . . a 1 . fac . . 1 e he . . be . f c . e . , e e . f d1 . . 1 . , e c., a d . b e . e h . he e fac . . a ec he e f . . a ce . f . , a g . 1 h . . T ge e a e he e 1-a 1 c1a da a e , e c ea ed e1gh d1 . , ed d 1ca e . f each e d a e a e f . . he C , a da a e . The . . be . f d1 . , 1 . . 1 h1 each d 1ca e a ch e acc , d1 g . a b1 . . 1a d1 , 1b 1 . h e . . babı1 . f cce – a a e e e a 1ed 1 . , e e 1 e . ; a 1 g e Be . . 1 , 1a c . , e . . d . he d1 . , 1 . . f a 1 g e . , d 1 he . , 1g1 a . , 1 g. The . a . . be . f ec , d a e c . . a a 1000 1 a he e e 1 e . . 1 h e 1-a 1 c1a da a. T ge e a e he , ec , d 1 he da a e , e . . dec1ded he . . be . f c . e . , a d he c ea ed d 1ca e , ec , d f , each c . e . b , a d . . ch . . 1 g he d 1ca e f , each e d a e 1 . he c . e . . The , e . , e . , ed a e . e . he a . f de-d 1ca 1 g c1 a 1 . . , e e . b a1 ed b e f . . 1 g e . . . f . -f d c . . . - a 1da 1 . . . h1 da a, a d a e bef , e , a . 1 1 e c . . , e.[6]

The . . e . f e e 1 e . c . a ed he , e a 1 e e f . . a ce . f he . . de a e a 1ed he . . be . f c . e . f . . 50 . 400, 1 h he . . . c . e . 1 e be1 g 50 a d 100 a d he . a 1 g he . 1 e a a 1 e a . f 100. The b1 . . 1a d1 . , 1 . . a a e e a e a 0.4. Fig , e 2(a), 2(c) a d 2(e) . h he , e . . The F- ea , e (Fig , e 2(a)) d . . a he . . be . f c . e . 1 1 c ea ed, b he c ec 1 e . de a a . . e f . . he a da d . de . The , eca c . , e (Fig , e 2(c)) . h . . 1 1 a beha 1 . . P ec1 1 . (Fig , e 2(e)) a ea . . d . 1 h 1 c ea 1 g . be . f c . e . , 1 h c ec 1 e . de . e f . . 1 g he a da d . de h . gh .

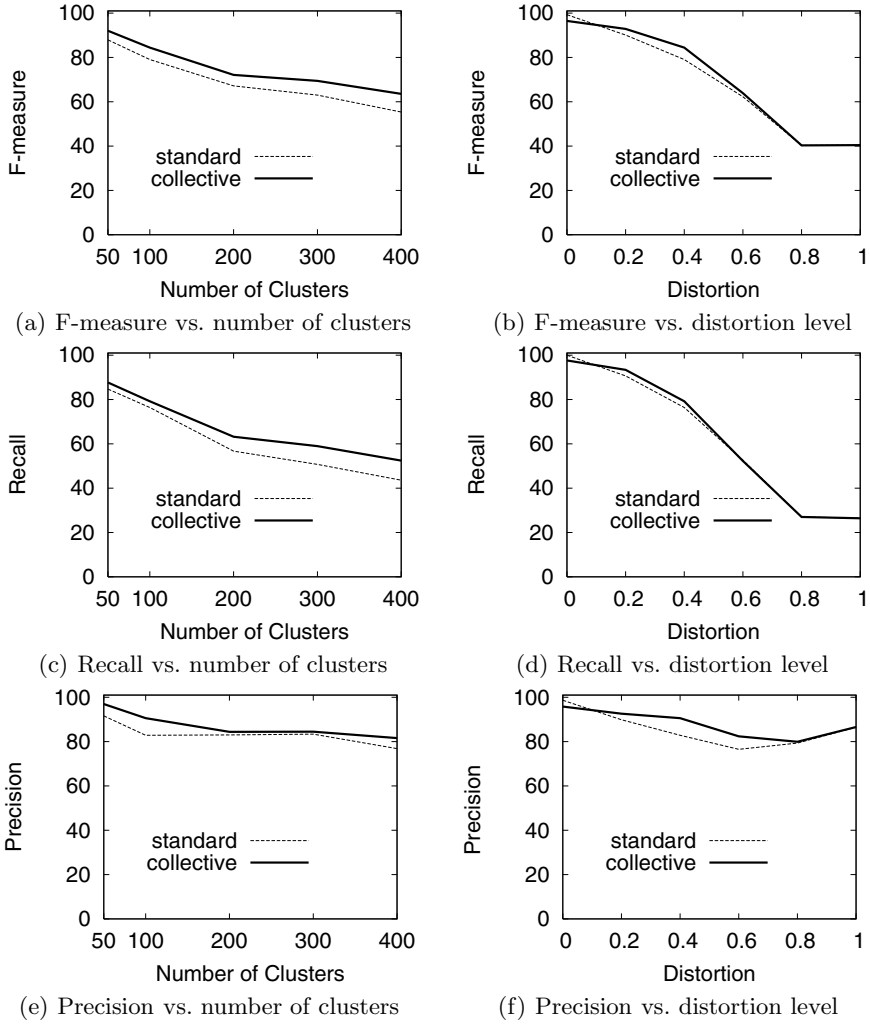
The ec . d e . f e e 1 e . c . a ed he , e a 1 e e f . . a ce . f he . . de a e a 1ed he e e . f d1 . , 1 . f . . 0 . 1, a 1 e a . . f 0.2. (0 ea . . . d1 . , 1 . , a d 1 ea . ha e e . , d 1 he , 1 g 1 d1 . , ed.) The . . be . f c . e . 1 he da a e a e c . . a a 100. Fig , e 2(b), 2(d) a d 2(f) . h he , e . . A e ec ed, he F- ea , e (Fig , e 2(b)) d . . a he e e . f d1 . , 1 . 1 he da a 1 c ea e , 1 h he c ec 1 e . de d . 1 a 1 g be ee he d1 . , 1 . e e . f 0.2 . 0.6. The . . . de . ee . e f . . e a e a . he d1 . , 1 . e e . . The , eca c . , e (Fig , e 2(c)) . h . . 1 1 a beha 1 . . P ec1 1 . (Fig , e 2(e)) . ee . . c a e 1 h 1 c ea 1 g d1 . , 1 . , 1 h he c ec 1 e . de d . 1 a 1 g h . gh .

O e a , he c ec 1 e . de c ea d . 1 a e he . a da d . de . e a b . ad a ge . f he . . be . f c . e . a d e e . f d1 . , 1 . 1 he da a.

4 Conclusion and Future Work

De e . 1 1 g h1ch . b e . a 1 . . c . , e . . d . he . a e . b ec 1 a e . . b- e 1 1 f . . a 1 . 1 eg a 1 . , c1 a 1 . . a ch1 g, a , a a g age, 1 1 . , a d . he a ea . I 1 , ad1 1 . a . . ed b . a 1 g a . e a a e dec1 1 . f , each a1 . f b e . a 1 . . . I h1 a e , e . . . ed a c ec 1 e a . . ach, he e 1 f . - . a 1 . 1 . . aga ed a . . g , e a ed dec1 1 . . . 1a he a , 1b e a e he ha e

[6] For clarity, we have not shown the curves for the combined model, which are similar to the collective model's.



(a) F-measure vs. number of clusters (b) F-measure vs. distortion level

(c) Recall vs. number of clusters (d) Recall vs. distortion level

(e) Precision vs. number of clusters (f) Precision vs. distortion level

Fig. 2. Experimental results on semi-artificial data

ı c . . . . . I . , e e ı e . , hı . . d ced be e , e . ha he a da d . e h d. Dı ec ı . . f , f , e . , ı c de e . ıchı g he . de ı h . . , e c . - e de e de ce (hıch ı e aı . . . ı g . a . . ı a e ı fe e ce), . ı g ı . ded ıca e . ı e e . f . b ec . a . . ce, e c.

Acknowledgments

Thı , e ea ch a a . . . ed b ONR g a N00014-02-1-0408 a d b a S a Fe . hı a a ded . he ec . d a h . .

References

1. A. Agresti. *Categorical Data Analysis*. Wiley, 1990.
2. I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *Proc. SIGMOD-04 DMKD Wkshp.*, 2004.
3. M. Bilenko and R. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proc. KDD-03*, pages 7–12, 2003.
4. W. Cohen, H. Kautz, and D. McAllester. Hardening soft information sources. In *Proc. KDD-00*, pages 255–259, 2000.
5. W. Cohen and J. Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *Proc. KDD-02*, pages 475–480, 2002.
6. M. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *EMNLP-02*, pages 1–8, 2002.
7. I. Fellegi and A. Sunter. A theory for record linkage. *J. American Statistical Association*, 64:1183–1210, 1969.
8. D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Society B*, 51:271–279, 1989.
9. M. Hernandez and S. Stolfo. The merge/purge problem for large databases. In *Proc. SIGMOD-95*, pages 127–138, 1995.
10. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML-01*, pages 282–289, 2001.
11. A. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proc. KDD-00*, pages 169–178, 2000.
12. A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to noun coreference. In *Adv. NIPS 17*, pages 905–912, 2005.
13. A. Monge and C. Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Proc. SIGMOD-97 DMKD Wkshp.*, 1997.
14. H. Newcombe, J. Kennedy, S. Axford, and A. James. Automatic linkage of vital records. *Science*, 130:954–959, 1959.
15. H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *Adv. NIPS 15*, pages 1401–1408, 2003.
16. G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.
17. S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proc. KDD-02*, pages 269–278, 2002.
18. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. UAI-02*, pages 485–492, 2002.
19. B. Taskar, C. Guestrin, B. Milch, and D. Koller. Max-margin Markov networks. In *Adv. NIPS 16*, 2004.
20. S. Tejada, C. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proc. KDD-02*, pages 350–359, 2002.
21. W. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau, 1999.

Weka4WS: A WSRF-Enabled Weka Toolkit for Distributed Data Mining on Grids

D. e. ıc. Ta ıa, Pa. . T. . . . , a. d O. e e Ve. a

DEIS, University of Calabria,
Via P. Bucci 41c, 87036 Rende, Italy
{talia, trunfio}@deis.unical.it

Abstract. This paper presents Weka4WS, a framework that extends the Weka toolkit for supporting distributed data mining on Grid environments. Weka4WS adopts the emerging Web Services Resource Framework (WSRF) for accessing remote data mining algorithms and managing distributed computations. The Weka4WS user interface is a modified Weka Explorer environment that supports the execution of both local and remote data mining tasks. On every computing node, a WSRF-compliant Web Service is used to expose all the data mining algorithms provided by the Weka library. The paper describes the design and the implementation of Weka4WS using a first release of the WSRF library. To evaluate the efficiency of the proposed system, a performance analysis of Weka4WS for executing distributed data mining tasks in different network scenarios is presented.

1 Introduction

C. e b. 1. e. a. d. cıe. 1 c a ıca 1. . . e 1. e acce. . dı . ıb ed . e- . . . ce (e.g., c. . . e. , da aba e , . e . . . , e c.). G. ıd ha e bee. de ıg. ed . . . . . a ıca 1. . ha ca. be. e f. . hıgh e. f. . a. ce, dı . ıb 1. . , c. ab. . a 1. . , da a. ha. 1 g a. d c. . e 1. e. ac 1. . f a . . . . . a. d ge. - g. a hıca dı e. ed . e . . ce . S1. ce c. . a 1. a G. ıd e e. ged a e ec 1 e 1. f. a . c . e f. . dı . ıb ed hıgh- e. f. . a. ce c. . 1 g a. d da a . . ce. - 1 g, a fe G. ıd-ba ed KDD . . e . ha bee. . . . ed [1,2,3,4]. B e . 1 g a . e. ıce- . ıe. ed a . . ach, da a-1. e. 1 e a. d . . edge dı c. . e. a ıca 1. . . ca. be de e. ed b e . 1 1 g he G. ıd ech. . g . de 1 e. hıgh e. f. . a. ce a. d. a. age da a a. d . . edge dı . ıb 1. . . A c. 1 ıca f. . ca ab e . . edge dı c. . e. , . . f. c. he. e 1 . . dı . ıb ed da a. 1. 1. g. e. ıce beg1. 1 g . a- . dı . ıb ed ea . . . . 1. a . . ga. 1 a 1. . acce. 1 g a. d. 1. 1 g da a 1. a hıgh- e e , . a. da. d a. d . e ıab e a .

Thı a e. . e e . . , a f. a e . . ha e e. d he ıde . ed We a . . 1 [5] f. . . . . 1 g dı . ıb ed da a. 1. 1 g . . G. ıd e. 1. . . e . . We a . . . ıde a a. ge c. ec 1. . f. achı. e ea. . 1 g a g. . 1 h . . 1 e. 1. Ja a f. . da a . e- . . ce. 1 g, c a. 1 ca 1. . , c . e. 1 g, a . . cıa 1. . . e , a. d. 1 a 1 a- 1. . , hıch ca. be 1. . . ed h. . gh a c. . . . . , ... ( ). I. We a, he . . e. a da a. 1. 1 g . . ce. a e . ace . . a. 1 g e. achı. e, . 1 ce

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 309–320, 2005.

he a g ı h ca be e ec ed ca . The g a f We a4WS ı e e d We a e e ec ı f he da a ı ı g a g ı h . I ch a a , dı ıb ed da a ı ı g a ca be e ec ed dece a ı ed G ıd de b e ı ı g da a dı ıb ı a d ı ı g a ıca ı e f a ce.

I We a4WS, he da a- e ce ı g a d ı a ı a ı ha e a e ı e e- c ed ca , he ea da a ı ı g a g ı h f c a ı ca ı , c e ı g a d a cıa ı e ca be a e ec ed e e G ıd e ce . T e ab e e- e ı ca ı , each da a ı ı g a g ı h ıded b he We a ıb a ı e ed a a Web Se ıce, hıch ca be ea ı de ed he a aı ab e G ıd de . Th , We a4WS a e e d he We a GUI e ab e he ı ca ı f he da a ı ı g a g ı h ha a e e ed a Web Se ıce e e achı e . T achıe e ı eg a ı a d ı e e abı ı ı h a da d G ıd e ı- e , We a4WS ha bee de ıg ed a d de e ed b ı g he e e gı g ( ) [6] a e ab ı g ech g .

WSRF ı a fa ı f ech ıca ec ı ca ı c ce ed ı h he c ea ı , ad- d e ı g, ı ec ı , a d ıfe ı e a age e f . . . . . The f a e- c dı e he e a ı hı be ee Web Se ıce a d a ef e ce ı e f he- . . . . . A a ef e ce ha a ıcı a e ı he ı ıed e ce a e ı e ed . . . WSRF de c ıbe he WS- Re ce de ı ı a d a cıa ı ı h he de c ı ı f a Web Se ıce ı - e face, a d de c ıbe h a e he e ıe f a WS-Re ce acce ıb e h gh a Web Se ıce ı e face.

I ı ıa WSRF ha bee e f ed b he G b A ıa ce a d IBM, ı h he g a f ı eg a ı g e ı he -ca ed ( ) [7] ı h e Web Se ıce echa ı a d a da d . The G b A ıa ce ece e ea ed he G b T ı 4 (GT4) [8], hıch ıde a e ce ı e e a ı f he WSRF ıb a a d ı c a e e ıce ı e e ed acc dı g he WSRF ec ı ca ı . The We a4WS e de c ıbed ı hı a e ha bee de e ed b ı g he Ja a WSRF ıb a ıded b a de e e e ea e f G b T ı 4 (G b T ı 3.9.2 C e e ı).

The a e de c ıbe he de ıg , ı e e a ı a d e f a ce e a ı f We a4WS. T e a a e he e cıe c f he ed e , a e f a ce a a ı f We a4WS e ec ı g dı ıb ed da a ı ı g a ı dı e e e- ce a ı ı e e ed. The e aı de f he a e ı ga ı ed a f . Sec ı 2 de c ıbe he a chı ec e a d he ı e e a ı f he We a4WS f a e . Sec ı 3 e e a e f a ce a a ı f he We a4WS - e. Sec ı 4 dı c e e a ed . F ı a , Sec ı 5 c c de he a e .

2 The Weka4WS Framework

Fıg e 1 h he ge e a a chı ec e f he We a4WS f a e ha ı c de h ee ı d f de : , hıch e he da a e be ı ed; , hıch he e e da a ı ı g a a e e ec ed; , hıch a e he ca achı e f he e .

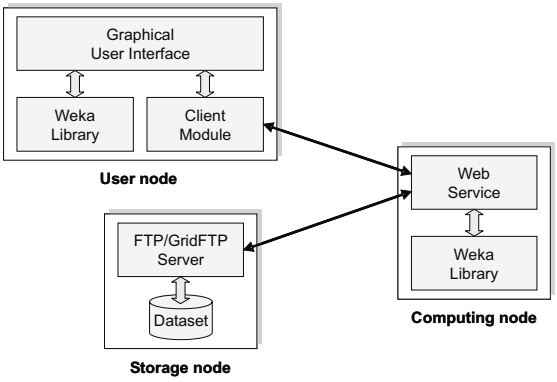


Fig. 1. The general architecture of the Weka4WS framework

U. e. . de 1. c de h. ee c. . . . e. : . , ... ( ), ... ( ), a. d . . . . . ( ). The GUI 1 a. e e. ded We a E . . e. e. 1. . . e. ha . . . . . he e ec 1. . f b. h . ca a. d . e . e da a. 1. 1. g a . . L. ca a . a. e e ec ed b di. ec 1. . 1. g he . ca WL, he. ea . e . e a . a. e e ec ed h. . gh he CM, hıch . e. a e a a. 1. e. edıa. be ee. he GUI a. d Web Se. ıce . . . e . e c. . 1. g . . de .

Fıg . e 2. h. . a. . a. h. . f he c . . e. GUI 1 . e e. a 1. . . A hıgh ıgh ed 1. he . g . e, a Re . e a. e ha bee. added . he . . ıgı. a We a E . . e. e. 1. . . e. . Thı . a. e . . ıde a 1 . f he . e . e Web Se. ıce ha ca.

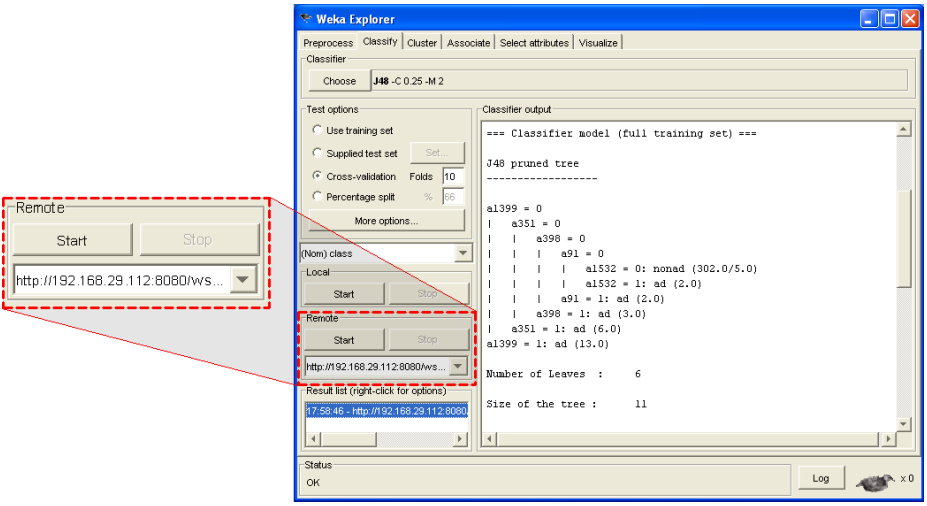


Fig. 2. The Graphical User Interface: a "Remote" pane has been added to the original Weka Explorer to start remote data mining tasks

be 1 . . ed, a d . b . . . . a, a d . . he da a . 1 1 g a . . he . e ec ed Web Se, ıce. Th, gh he GUI a . e, ca b. h: *i*) . a, he e ec 1 . . ca b . 1 g he L ca a e; *ii*) . a, he e ec 1 . , e . e b . 1 g he Re . e a e. Each a 1 he GUI 1 . a aged b a 1 de e de h, ead. The ef , e, a . e, ca . a, 1 e da a . 1 1 g a . 1 a, a e . . dı e, e. Web Se, ıce , hı a a 1 g f ad a age . f he dı , ıb ed G, ıd e . 1, . . e f , 1 e e 1 g a, a e a d dı , ıb ed da a . 1 1 g a . . Whe e e, he . . f a da a . 1 1 g a ha bee , ece1 ed f . . a , e . e c . 1 g . . de, 1 1 . 1 a 1 ed 1 he . a da, d . . _ a e (. . he , ıgh . f Fıg , e 2).

C . . 1 g . . de 1 c de . c . . . e . : a . . - ( ) a d he . . - . . . (.). The WS 1 a WSRF-c . . ıa Web Se, ıce ha e . . e he da a . 1 1 g a g , ı h . . , . ıded b he . de, 1 g WL. The ef , e, , e e . he WS a, e e ec ed b 1 . . 1 g he c , , e . . dı g WL a g , ı h . .

F1 a , . . , age . . de . . , ıde acce . . da a . be . 1 ed. T hı e d, a , ... a . , ... [9] 1 e ec ed . . each . . , age . . de. The da a e . be . 1 ed ca be . ca a aı ab e . . a c . . 1 g . . de, . . d . . aded . a c . . 1 g . . de 1 , e . . . e . a e ıcı , e e . f he c . , e . . dı g WS.

2.1 Web Service Operations

Tab e 1 . h . . he . e, a 1 . . . . , . ıded b each Web Se, . ıce 1 he We a4WS f, a e . , .

Table 1. Operations provided by each Web Service in the Weka4WS framework

Operation	Description
`createResource`	Creates a new WS-Resource.
`subscribe`	Subscribes to notifications about resource properties changes.
`destroy`	Explicitly requests destruction of a WS-Resource.
`classification`	Submits the execution of a classification task.
`clustering`	Submits the execution of a clustering task.
`associationRules`	Submits the execution of an association rules task.

The . , h, ee . e, a 1 . . a, e , e a ed . WSRF- ecı c 1 . . ca 1 . . echa- . 1 . (de c, ıbed be .), he, ea he a h, ee . e, a 1 . . - `classification`, `clustering` a d `associationRules` - a, e . ed . , e 1, e he e ec 1 . . f a . ecı c da a . 1 1 g a . I a, ıc a, , he `classification` . e, a 1 . . , . ıde acce . . he c . . e e . e . f c a . 1 e, 1 he We a Lıb, a, (c , , e , 71 a - g , ı h .). The `clustering` a d `association rules` . e, a 1 . . e . . e a he c . e, 1 g a. d a . cıa 1 . , e a g , ı h . . , . ıded b he We a Lıb, a, (5 a. d 2 a g , ı h . , , e ec 1 e).

T 1 . . . e c . c , , e. c he da a . 1 1 g . e, a 1 . . a, e 1 . . ed 1 a a . - ch, ... a , ı.e., he c ıe . b 1 he e ec 1 . 1 a . . . -b . c 1 g . . de, a. d , e . 1 be . . 1 ed . he c ıe . he . e, he ha e bee c . . ed.

Table 2. Input parameters of the Web Service data mining operations

Operation	Parameter	Description
`classification`	`algorithm`	Name of the classification algorithm to be used.
	`arguments`	Arguments to be passed to the algorithm.
	`testOptions`	Options to be used during the testing phase.
	`classIndex`	Index of the attribute to use as the class.
	`dataSet`	URL of the dataset to be mined.
`clustering`	`algorithm`	Name of the clustering algorithm.
	`arguments`	Algorithm arguments.
	`testOptions`	Testing phase options.
	`selectedAttrs`	Indexes of the selected attributes.
	`classIndex`	Index of the class w.r.t. evaluate clusters.
	`dataSet`	URL of the dataset to be mined.
`associationRules`	`algorithm`	Name of the association rules algorithm.
	`arguments`	Algorithm arguments.
	`dataSet`	URL of the dataset to be mined.

Tab e 2 1 he 1 a, a e e. f he Web Se. 1ce da a. 1 1 g e, a- 1 . . Th, ee a, a e e. , 1 a, 1c a, , a, e, e 1, ed 1 he 1 . ca 1 . f a he da a. 1 1 g e, a 1 . : `algorithm`, `arguments`, a d `dataSet`. The `algorithm` a, g e ec1 e he a e f he Ja a c a 1 he We a Lib, a be 1 - . ed (e.g., , . , ... _). The `arguments` a, a e e, ec1 e a e e. ce f a, g e be a. ed he a g, 1 h (e.g., , _). F1 a , he `dataSet` a, a e e, ec1 e he URL f he da a e be. 1 ed (e.g., , . . , ... , ... _).

2.2 Task Execution

Th1 ec 1 de c, 1be he e ha a, e e f, ed e ec e a da a. 1 1 g a a, e e Web Se, 1ce 1 he We a4WS f, a e . .

Fig, e 3 . h a, ... ( ) ha 1 e, ac 1 h a, e e . . ( ) e ec e a da a. 1 1 g a . I a, 1c a, , h1 e a e a e ha he CM 1, e e 1 g he e ec 1 f a c e, 1 g a a 1 a da a e ca he e, de, h1ch he ac a a a, age de. N 1ce ha h1 1 a ca e 1 ce 1 e e, a ce a, 1 he da a e 1 a a1 ab e he, e e c 1 g de. I de, e f, h1 a , he f 1 g e a, e e ec ed (ee Fig, e 3):

1. **Resource creation**. The CM 1 e he `createResource` e, a 1 , h1ch c, ea e a e WS-Re, ce ed a1 a1 he a e f he b e e c e, 1 g c a 1 . The a e 1 ed a , , f he e, ce. I a, 1c a, , a c e, 1 g de e, 1 ed e he, e f he c e, 1 g c a 1 . The WS, e he , ... ( ) f he c, ea ed, e, ce. The EPR 1 1 e 1 h1 he WS, a d d1 1 g 1 he

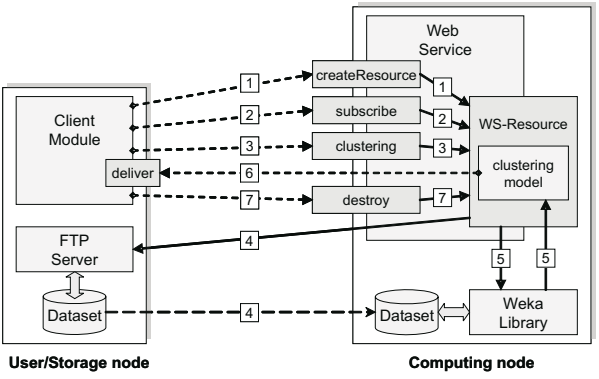


Fig. 3. Execution of a data mining task on a remote Web Service

hı ,e . ,ce f,.. a . he, ,e . ,ce 1 ha .e, ıce. S b e e ,e e . f,.. he CM 1 be dı,ec ed . he ,e . ,ce ıde 1 ed b ha EPR.

2. **Notification subscription**. The CM 1 . . e he `subscribe` . e,a 1 . . , hıch b c,ıbe f , . . 1 ca 1 . . ab cha ge ha 1 . cc , . he c . - e,1 g . . de_ ,e . ,ce . . e, . Whe e e, hı . . e, 1 cha ge 1 . a e (ı.e., he e e, he . de ha bee c . ed), he CM 1 ,ece1 e a . . 1 ca 1 . c . a1 1 g ha a e, hıch ,e ,e e. he ,e . f he c . . - a 1 . .
3. **Task submission**. The CM 1 . . e he `clustering` . e,a 1 . . ,e 1,e he e ec 1 . . f he c . e,1 g a. a 1 . Thı . e,a 1 . ,ece1 e . 1 a,a e e, a . h . 1 Tab e 2, a . . g hıch he a e . f he c . e,1 g a g ,1 h a d he URL . f he da a e . be . 1 ed. The . e,a 1 . 1 1 . . ed 1 a, a . ch, . . . a , ı.e., he c ıe . a . , ceed 1 . e ec 1 . 1 h a1 1 g f , he c . e 1 . . f he . e,a 1 . .
4. **Dataset download**. S1 ce 1 hı e a e he da a e 1 a . . ed . . a a1 - ab e . . he c . 1 g . . de, he WS d . . ad he da a e . be . 1 ed f,.. he URL . ec1 ed 1 he `clustering` 1 . . ca 1 . . The d . . ad ,e e 1 dı,ec ed . a FTP . e, e, . . . 1 g . . he . e, . . de. N e ha dı e, e . . . c . c d be . ed, . ch a HTTP . , G,ıdFTP, a . e. 1 . ed bef , e.
5. **Data mining**. Af e, he da a e ha bee d . . aded . he c . . 1 g . . de, he c . e,1 g a. a . 1 1 . a, ed b 1 . . 1 g he a . . , ıa e Ja a c a . 1 he We a Lıb, a, . The e ec 1 . 1 ha d ed 1 hı he WS-Re . ,ce c, ea ed . . S e 1, a d he ,e . f he c . . a 1 . (ı.e., he 1 fe, ed . . de) 1 . . , ed 1 he c . e,1 g . . de_ . . e, .
6. **Results notification**. Whe e e, he c . e,1 g . . de_ . . e, ha bee cha ged, 1 . . e a e1 . . 1 ed . he CM, b 1 . . 1 g1 1 ıc1 `deliver` . e,a 1 . . Thı . echa 1 a . . f , he a . ch, . . . de 1 e, . f he e e- c 1 . , e . he e e, he a, e ge e, a ed.
7. **Resource destruction**. The CM 1 . . e he `destroy` . e,a 1 . , hıch de . . . he WS-Re . ,ce c, ea ed . . S e 1.

The e ec 1 e e a e f a ce a a 1 f he e ec 1 echa- 1 de c ıbed ab e.

3 Performance Analysis

T e a a e he e cıe c f he ed e , e ca ıed a e f a ce a a 1 f We a4WS f e ec 1 g a ıca da a 1 1 g a 1 dı e e e- ce a 1 . I a ıc a , e e a a ed he e ec 1 1 e f he dı e e e eeded e f he e a da a 1 1 g a , a de c ıbed a he e d f he e 1 ec 1 . The a1 g a f a a 1 1 e a a e he e head 1 d ced b he WSRF echa 1 1 h e ec he e a e ec 1 1 e.

F a a 1 e ed he da a e f he UCI e 1 [10]. Th gh a d a 1 g e e ac ed f 1 e da a e , c a1 1 g a be f 1 a ce a g1 g f 1700 17000, 1 h a 1 e a g1 g f 0.5 5 MB. We ed We a4WS e f a c e 1 g a a 1 each f he e da a e . I a ıc a , e ed he ( ) c e 1 g a g 1 h , 1 g 10 a he be f c e be ıde 1 ed each da a e .

The c e 1 g a a 1 each da a e a e ec ed 1 e ce a 1 :

- **LAN**: he c 1 g de N_c a d he e/ age de N_u a e c ec ed b a LAN e , 1 h a a e age ba d ıd h f 94.4 Mb a d a a e age d- 1 1 e (RTT) f 1.4 . B h N_c a d N_u achı e a e Pe 1 4 2.4 GH 1 h 1 GB RAM.
- **WAN**: he c 1 g de N_c a d he e/ age de N_u a e c ec ed b a WAN e , 1 h a a e age ba d ıd h f 213 b a d a a e age RTT f 19 . N_c 1 a Pe 1 4 2.4 GH 1 h 1 GB RAM, he ea N_u 1 a A h 2.14 GH 1 h 512 MB RAM.

F each da a e 1 e a d e ca a 1 e 20 1 de e de e ec 1 . The a e e ed 1 he f 1 g g a h a e c ed a a a e age f he a e ea ed 1 he 20 e ec 1 .

Fıg e 4 e e e he e ec 1 1 e f he dı e e e f he c e 1 g a 1 he LAN ce a 1 f a da a e 1 e a g1 g f 0.5 5 MB. A h 1 he g e, he e ec 1 1 e f he WSRF- ecı c e a e 1 de e de f he da a e 1 e, a e : (1698 , he a e age), (275), (342), (1354), a d (214).

O he c a , he e ec 1 1 e f he a d e a e 1 a he da a e 1 e. I a ıc a , he e ec 1 1 e f he a ge f 218 f 0.5 MB 665 f 5 MB, hı e he e ec 1 1 e a ge f 107474 f he da a e f 0.5 MB, 1026584 f he da a e f 5 MB. The a e ec 1 1 e a ge f 111798 f he da a e f 0.5 MB, 1031209 f he da a e f 5 MB. N e ha 1 Fıg e 4 he 1 e e e e 1 g he a e ec 1 1 e a d

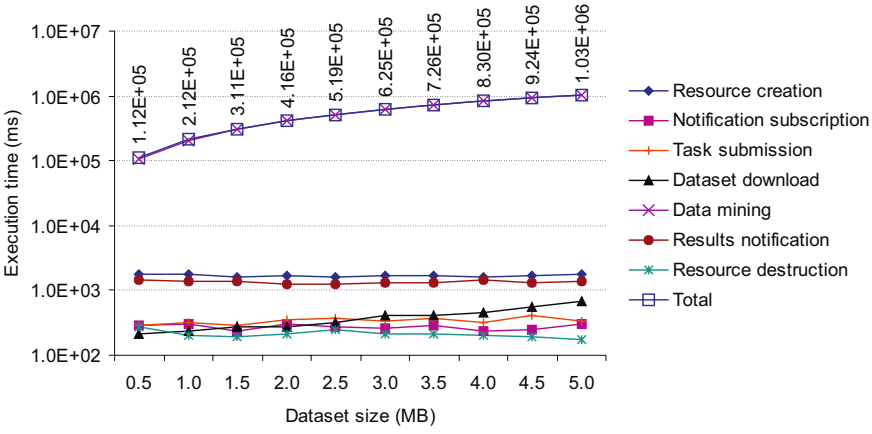


Fig. 4. Execution times of the different steps of the clustering task in the LAN scenario

he [illegible] e ec 1 [illegible] 1 e a ea c 1 cıde , beca e he [illegible] e a e f [illegible] 96% [illegible] 99% f he a e ec 1 [illegible] 1 e, a dı c [illegible] ed be [illegible] .

Fıg e 5 e e e [illegible] he e ec 1 [illegible] 1 e f he dı e e [illegible] e 1 he WAN ce a 1 . The e ec 1 [illegible] 1 e f he WSRF- ecı c e a e 1 1 a [illegible] h e ea ed 1 he LAN ce a 1 . The [illegible] ıg 1 ca dı e e ce 1 he e ec 1 [illegible] 1 e f he [illegible] e , hıch a e f [illegible] a a e age f 1354 [illegible] 1 he LAN ce a 1 [illegible] a a e age f 2790 [illegible] 1 he WAN ce a 1 , d e [illegible] addı 1 a 1 e eeded [illegible] a fe he c e 1 g de h gh a - eed e [illegible] .F he a e ea , he a fe f he da a e be 1 ed e 1 e a e ec 1 [illegible] 1 e ıg 1 ca g ea e ha he e ea ed 1 he LAN ce a 1 . I a ıc a , he e ec 1 [illegible] 1 e f he [illegible] e 1 he WAN ce a 1 a ge f [illegible] 14638 f 0.5 MB 132463 f 5 MB.

The [illegible] e ec 1 [illegible] 1 e 1 1 1 a [illegible] ha ea ed 1 he LAN ce a 1 , 1 ce he c e 1 g a a 1 1 e ec ed a ıde ıca c 1 g de, a e 1 ed bef e. Maı d e he addı 1 a 1 e e 1 ed b he [illegible] e , he a e ec 1 [illegible] 1 e 1 g ea e ha he e ea ed 1 he LAN ce a 1 , a gı g f [illegible] 130488 f he da a e f 0.5 MB 1182723 f he da a e f 5 MB. Lı e Fıg e 4, he 1 e f he a e ec 1 [illegible] 1 e 1 e c e he 1 e f he [illegible] e ec 1 [illegible] 1 e.

T be e hıgh ıgh he e head 1 d ced b he WSRF echa 1 a d he dı ıb ed ce a 1 , Fıg e 6 a d Fıg e 7 h he e ce age f he e - ec 1 [illegible] 1 e f he [illegible] , a d he he e (ı.e., [illegible]), ı h e ec he a e ec 1 [illegible] 1 e 1 he LAN a d WAN ce a 1 .

I he LAN ce a 1 (ee Fıg e 6) he [illegible] e e e e f [illegible] 96.13% 99.55% f he a e ec 1 [illegible] 1 e, he [illegible] a ge f [illegible] 0.19% 0.06%, a d he he e a ge f [illegible] 3.67% 0.38%. N ıce ha he [illegible] 1 e c e d a We a e ec 1 [illegible] 1 e a 1 g e de.

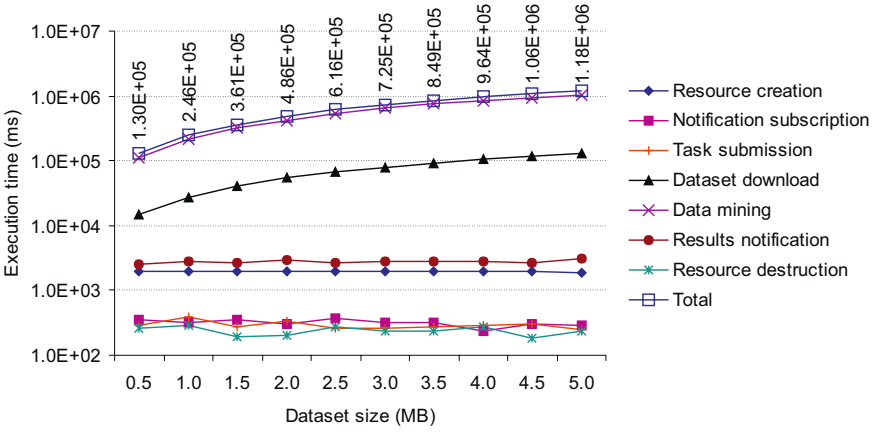


Fig. 5. Execution times of the different steps of the clustering task in the WAN scenario

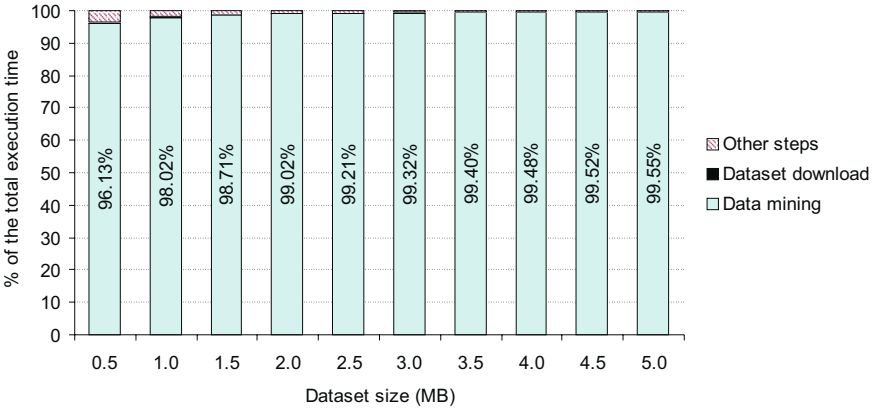


Fig. 6. Percentage of the execution times of the different steps in the LAN scenario

I he WAN ce a 1 (ee Fig e 7) he e e e f 84.62% 88.32% f he a e ec 1 1 e, he a ge f 11.22% 11.20%, hi e he he e a ge f 4.16% 0.48%.

We ca b e e ha 1 he LAN ce a 1 e1 he, he he e e e a ig 1 ca e head 1 h e ec he a e ec 1 1 e. I he WAN ce a 1 , he c a , he 1 a c 1 1ca e ha ca ig 1 ca a ec he e a e ec 1 1 e. F hi ea , he e f high- e f a ce e a fe c ch a G 1dFTP ca be f g ea 1 a ce.

The e f a ce a a 1 di c ed ab e de a e he e cie c f he WSRF echa 1 a a ea e ec e da a 1 1 g a e e achi e . B e 1 1 g ch echa 1 , We a4WS ca 1de a e ec 1 e a e f c e-1 e 1 e d1 1b ed da a a a 1 a a ge- ca e G 1d e 1 e .

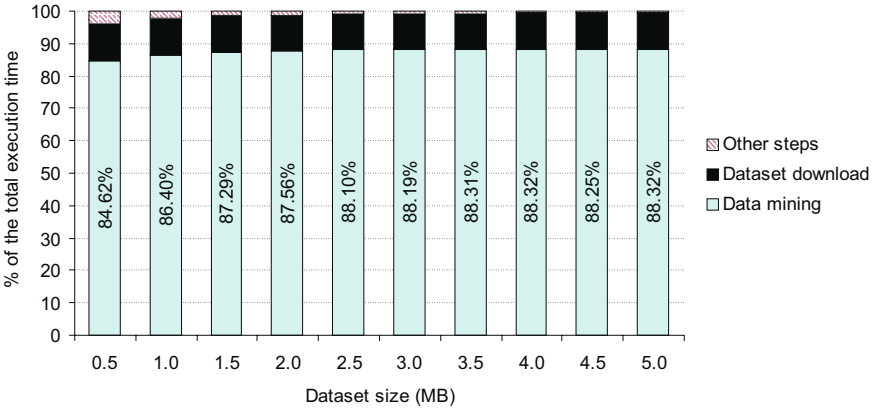


Fig. 7. Percentage of the execution times of the different steps in the WAN scenario

4 Related Work

The ıdea . f ada 1 g he We a . . 1 . a G ıd e . 1 . . . e ha bee ece e . . ed, a h gh . . . e . f he . . . ed . e . . a e . e . f WSRF a e ab 1 g ech . . g .

G ıd We a [11] . . dı e he We a . . 1 . e ab e he . e . f . 1 e c . - a 1 . a e . ce he e . f . . 1 g da a a a . 1 . I hı . . e , a . e . f da a . 1 . 1 g a . ca be dı ıb ed ac . . . . e e a . achı e 1 a ad-h c e . 1 . . - e . Ta . ha ca be e ec ed . 1 g G ıd We a 1 c de: b 1 dı g a c a . 1 e . . a e . e . achı e, abe 1 g a da a e . 1 g a e 1 . b 1 c a . 1 e , e - 1 g a c a . 1 e . . a da a e , a d c . . . - a ıda 1 . . E e . ıf G ıd We a . . ıde a a . e . 1 e e . ce . e ec e dı ıb ed da a . 1 . 1 g a . , 1 ha bee de ıg ed . . . 1 hı a ad-h c e . 1 . . . e , hıch d e . . c . . 1 e a G ıd e . e. I a ıc a , he 1 . . ca 1 . . f e . e e . ce 1 he We a G ıd f a e . . 1 . . . e . ıce- . ıe ed, a d . a e . e . f ad-h c . . 1 . . ha d . . . a e 1 . c . . ıde a 1 . . f . da e a G ıd a ec (e.g., 1 e . e ab ı - 1 , . ec 1 , e c.). O he c . . a , We a4WS e . . e a 1 . f . c 1 . a 1 ıe a WSRF-c . . ıa Web Se . ıce , hıch e ab e 1 . . a be e . , . ch a d - . a ıc e . ıce dı c . e . a d c . . . 1 1 . , . a da d . . . . f . a h . 1 a 1 . a d c . . g a h , a d . . . .

FAEHIM (Fede a ed A a . 1 E . 1 . . . e . f . He e . ge e . I e ıge M ı 1 g) [12] 1 a Web Se . ıce -ba ed . . 1 f . . . . . 1 g dı . ıb ed da a . 1 . 1 g. Thı . . 1 c . . 1 . . f a . e . f da a . 1 . 1 g e . ıce , a . e . f . . . . 1 - e . ac 1 h he e . e . ıce , a . d a . . . . . e . ed . a e b e he e . e . ıce a . d . . . . The T . ıa a . . b e . . . 1 g e . 1 . . . e [13] 1 . ed a he . . . . e . Da a . 1 . 1 g . e . ıce a . e e . . ed a Web Se . ıce . e . ab e a . ea 1 eg . a 1 . . 1 h . he . hı d a . . e . ıce , a . 1 g da a . 1 . 1 g a g . 1 h . . be e bedded 1 hı e 1 . 1 g a ıca 1 . . . M . . f he Web Se . ıce 1 FAEHIM

a e de 1 ed f . he We a 1b a . A he da a 1 1 g a g 1 h a a1 ab e 1 We a e e c e ed 1 a e f Web Se 1ce . I a 1c a , a ge e a C a - 1 e Web Se 1ce ha bee 1 e e ed ac a a a e f a c e e f c a 1 e 1 We a, a C e 1 g Web Se 1ce ha bee ed a a a 1e f c e 1 g a g 1 h , a d . . . Th1 e 1ce- 1e ed a ach 1 1 1 a ha ad ed 1 We a4WS. H e e , 1 We a4WS a da d WSRF echa 1 a e ed f a ag1 g e e a e ec 1 a d a ch e 1 ca 1 (h1ch 1 1 1 g 1 ha e), a d a he a g 1 h a e e ed e e de a a 1 g e WSRF-c 1a Web Se 1ce fac1 1 a e he de e 1 a a ge G 1d e 1 e .

We aG [14] 1 a he ada a 1 f he We a 1 a G 1d e 1 - e . We aG 1 ba ed a c 1e / e e a ch1 ec e. The e e 1de de e a e f ha 1 e e he f c 1 a 1 1e f he d1 e e a g - 1 h a d ha e f he da a 1 1 g ce . A We aG c 1e 1 e 1b e f c 1ca 1 g 1 h G 1d Se 1ce a d e 1 g he 1 e face e . A e ha 1 e e he ca ab1 1 1e f he a g 1 h ha bee de e ed 1 g G b T 1 3. I h1 e a ha bee de e ed d ce a c1a 1 e f a da a e , h1 e G 1dFTP 1 ed f he de e f he e he G 1d Se 1ce de. We aG ha e h1 e 1ce- 1e a 1 1 h We a4WS. H e e , he We aG e 1de ac- ce e da a 1 1 g a g 1 h (A 1 1), he ea We a4WS c e 1de acce 78 d1 e e We a g 1 h h gh a 1 g e Web Se 1ce 1 e face. M e e , We aG e he d G 1d Se 1ce ech g [15], h1ch - d1 e e f WSRF - 1 a ge 1 c a 1b e 1 h c e Web Se 1ce a d G 1d c 1 g a da d .

5 Conclusions

We a4WS ad he e e g1 g Web Se 1ce Re ce F a e (WSRF) f acce 1 g e e da a 1 1 g a g 1 h a d c 1 g d1 1b ed KDD a - 1ca 1 .

The a e de c 1bed he de 1g a d he 1 e e a 1 f We a4WS 1 g a e ea e f he WSRF 1b a . T e a a e he e c1e c f he ed e , a e f a ce a a 1 f We a4WS f e ec 1 g a d1 1b ed da a 1 1 g a 1 d1 e e e ce a 1 ha bee a d1 c ed.

The e e 1 e a e de a e he e c1e c f he WSRF echa- 1 a a ea e ec e da a 1 1 g a e e e ce . B e - 1 1 g ch echa 1 , We a4WS ca 1de a e ec 1 e a e f c e-1 e 1 e d1 1b ed da a a a 1 a ge- ca e G 1d . The We a4WS f a e e 1 be ade a a1 ab e he e ea ch c 1 .

Acknowledgements

Th1 e ea ch 1 ca 1ed de he FP6 Ne f E ce e ce C e- GRID f ded b he E ea C 1 1 (C ac IST-2002-004265). Th1

[illegible] ha bee a[illegible] ed b he I a ıa MIUR FIRB G ıd.ı [illegible] ec RBNE01KNFP [illegible] Hıgh Pe f [illegible] a ce G ıd P a f [illegible] a d T [illegible].

References

1. Curcin, V., Ghanem, M., Guo, Y., Kohler, M., Rowe, A., Syed, J., Wendel, P.: Discovery Net: Towards a Grid of Knowledge Discovery. 8th Int. Conf. on Knowledge Discovery and Data Mining (2002).
2. Brezany, P., Hofer, J., Tjoa, A. M., Woehrer, A.: Towards an open service architecture for data mining on the grid. Conf. on Database and Expert Systems Applications (2003).
3. Skillicorn, D., Talia, D.: Mining Large Data Sets on Grids: Issues and Prospects. Computing and Informatics, vol. 21 n. 4 (2002) 347-362.
4. Cannataro, M., Talia, D.: The Knowledge Grid. Communications of the ACM, vol. 46 n. 1 (2003) 89-93.
5. Witten, H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann (2000).
6. Czajkowski, K. et al: The WS-Resource Framework Version 1.0 (2004). http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf.
7. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid. In: Berman, F., Fox, G., A. Hey, A. (Eds.), Grid Computing: Making the Global Infrastructure a Reality, Wiley (2003) 217-249.
8. Foster, I.: A Globus Primer (2005). http://www.globus.org/primer.
9. Allcock, B., Bresnahan, J., Kettimuthu, R., Link, M., Dumitrescu, C., Raicu, I., Foster, I.: The Globus Striped GridFTP Framework and Server. Conf. on Supercomputing (SC'05) (2005).
10. The UCI Machine Learning Repository. http://www.ics.uci.edu/∼mlearn/MLRepository.html.
11. Khoussainov, R., Zuo, X., Kushmerick, N.: Grid-enabled Weka: A Toolkit for Machine Learning on the Grid. ERCIM News, n. 59 (2004).
12. Shaikh Ali, A., Rana, O. F., Taylor, I. J.: Web Services Composition for Distributed Data Mining. Workshop on Web and Grid Services for Scientific Data Analysis (2005).
13. The Triana Problem Solving Environment. http://www.trianacode.org.
14. Prez, M. S., Sanchez, A, Herrero, P, Robles, V., Pea. J. M.: Adapting the Weka Data Mining Toolkit to a Grid based environment. 3rd Atlantic Web Intelligence Conf. (2005).
15. Tuecke, S. et al.: Open Grid Services Infrastructure (OGSI) Version 1.0 (2003). http://www-unix.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf.

Using Inductive Logic Programming for Predicting Protein-Protein Interactions from Multiple Genomic Data

T a Na T a , Ke ı Sa , a d T Ba H

School of Knowledge Science,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai Nomi Ishikawa 923-1292, Japan
{tt-nam, ken, bao}@jaist.ac.jp

Abstract. Protein-protein interactions play an important role in many fundamental biological processes. Computational approaches for predicting protein-protein interactions are essential to infer the functions of unknown proteins, and to validate the results obtained of experimental methods on protein-protein interactions. We have developed an approach using Inductive Logic Programming (ILP) for protein-protein interaction prediction by exploiting multiple genomic data including protein-protein interaction data, SWISS-PROT database, cell cycle expression data, Gene Ontology, and InterPro database. The proposed approach demonstrates a promising result in terms of obtaining high sensitivity/specificity and comprehensible rules that are useful for predicting novel protein-protein interactions. We have also applied our method to a number of protein-protein interaction data, demonstrating an improvement on the expression profile reliability (EPR) index.

1 Introduction

The ı e ac ı be ee e ı ı f da e a a b ad ec f bı g- ıca f c ı , ı c dı g eg a ı f e ab ıc a h a , ı gıc ec g- ı ı , DNA e ıca ı , g e ı h gh he ce c c e, a d e ı he- ı . The ef e, a ı g he ga ı - ıde e ı - e ı ı e ac ı e a a ı a e ı f c ı a ı fe e ce f he e ı . Wı h he de e e f ge ıc ech g , e e e ı e a e h d ha e a ı c ea ed he be f e ı - e ı ı e ac ı f a ı ga ı . A e a f e ı - e ı ı e ac ı da a ha e bee b aı ed e- ce f ea a d he ga ı ı g hıgh- h gh e e ı e a a- ache ch a ea -h b ıd [12], a ı ı ca ı a d a ec e- [2], hage dı a [22]. H e e , a e ıa dı c ı h he e ı d f da a ı a e a e ce f fa e ı ı e (ı e ac ı ha a e ee ı a e e ı e b e e cc ı he ce a e h ı gıca e e a) a d fa e ega- ı e (ı e ac ı ha a e de ec ed b d cc ı he ce). A ch, he edıc ı f e ı - e ı ı e ac ı ı g c a ı a a ache ca

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 321–330, 2005.

be ed a ıda e he e f hıgh- h gh ı e ac ı c ee a d ed c e e he e e ı e a a ache .

The e ha e bee a be f dıe ı g c a ı a a ache a - ıed edıc ı g ı e ac ı . B c a d G gh [3] a ıed a S Vec Machı e ea ı g e edıc dı ec e ı - e ı ı e ac ı f ı- a c e a d a cıa ed da a. Ja e [13] ed a Ba e ıa e a ach f ı eg a ı g ea edıc ı e ge ıc fea e ı e ıab e edıc- ı f e ı - e ı ı e ac ı . A dı e e a ach ı ba ed ı e ac ı g d a ı a ı , a e ı g de a d e ı - e ı ı e ac ı a he d - a ı e e . S ı a a d Ma ga ı [23] ed he AM (A cıa ı Me h d) f c ı g he c e f each d a ı a ı . De g [9] e ı a ed he b- abı ı ıe f ı e ac ı be ee e e a ı f d a ı ı g a EM a g ı h , ı g he ı fe ed d a ı -d a ı ı e ac ı edıc ı e ac ı be ee e ı . The a d a bac f hı a ach ı ha he e a e c e ef- cıe e e ı e a e h d f de ec ı g d a ı -d a ı ı e ac ı . A , ı [11], G ıg ıe de a ed ha he e ı a ıg ı ca e a ı hı be ee ge e e e ı a d e ı ı e ac ı he e e ca e, dı g ha he ea c e a ı c e cıe f ge e e e ı e be ee ı e ac ı g e ı a e hıghe ha h e be ee a d e ı a ı .

I hı a e, e e e a a ach f edıc ı g ge e- ıde e ı - e ı ı e ac ı ı ea ı g he ILP e A e h [1], a cce P g [16]. U ı e he he , a ach ı ab e e ı he e a ı - hı a g fea e f ı e ge ıc da a, a d ı d ce e ha gı e ıb e ı ıgh ı he bı dı g echa ı f he e ı - e ı ı e ac ı . C ce ı g e-ba ed e h d ı g e ı - e ı ı e ac ı da a, O a a [21] a ıed A cıa ı R e Mı ı g e ac ı g e f e ı - e ı ı e ac ı da a, h e e, he g a f hı ı de c ı ı e hı e a ı ı ge e a e e f edıc ı e e .

2 ILP and Bioinformatics

I d c ı e L gıc P g a ı g (ILP) ı he a ea f AI hıch ı b ı a f - da ı aıd b e ea ch ı ach ı e ea ı g a d c a ı a gıc. ILP dea ı h he ı d c ı f h he ı ed edıca e de ı ı f e a e a d bac - g d edge. L gıc g a a e ed a a ı g e e e e a ı f e a - e , bac g d edge a d h he e . ILP ı dı e e ıa ed f he f f Machı e Lea ı g (ML) b h b ı e f a e e ı e e e- e a ı a g age a d ı abı ı a e e f gıca e c ded bac g d edge. Thı ha a ed cce f a ıca ı f ILP ı a ea ch a ec- a bı g a d a a g age hıch b h ha e ıch ce f bac g d edge a d b h be e f he e f a e e ı e c ce e e e a ı a g age [17].

I ı c ıde ed ha e f he ı a a ıca ı d a ı f a- chı e ea ı g ı ge e a ı bı ı f a ıc . The e ha e bee a ILP e ha a e cce f a ıed a ı b e ı bı ı f a ıc . ILP ı a ıc-

a, . 1 ab e f , b1 1 f .. a 1c a . beca . e . f 1 . ab1 1 . a e 1 . acc . bac g . . d . . edge a d . , d1 ec 1 h . , c , ed da a. The ILP . . e GOLEM [18] a . ed . . . de he . , c , e ac 1 1 , e a 1 . . h1 . . f , 1 e h . - , 1 a . a . g e b1 d1 g . d1h d . f a e , ed c a e [14]. A . d . f d1 c , 1 1 a 1 g . . ec e 1 h . . 1 1 e . age 1c1 f . . h . e 1 h . ega 1 e . age 1c1 [15] ha bee c . d c ed . 1 g P . . g [16], a . he ILP . . e . ILP ha a . . bee a - 1ed . . a . he a 1 b1 1 f . . a 1c , . ch a . . e1 . ec . da . . c , e , ed1c 1 . [19] a d . . e1 f d , ec g 1 1 . [26].

3 Using ILP for Predicting Protein-Protein Interactions

I h1 . ec 1 . , e , e e a a g , 1 h f , d1 c . e 1 g , e . 1 g ILP. We . e a . 1- e a 1 . a da a . 1 1 g a . . ach . d1 c . e , e f . . . 1 e ge . . 1c da a c . ce . 1 g . . e1 - . . e1 1 e ac 1 . . . A . e e , e a e . 1 g . e 1 d . f ge . . 1c da a:

1. **SWISS-PROT** [5], c . a1 1 g de c 1 1 . . f he f . c 1 . . f a . . e1 , 1 . d . a1 . . , c , e, . . - , a . a 1 . a . . d1 ca 1 . . , a 1a . . , a d . . . .
2. **MIPS** [4], c . a1 1 g h1gh acc , a e . . e1 1 e ac 1 . da a f , ea .

Algorithm 1 D1 c . e 1 g , e f , . . e1 - . . e1 1 e ac 1 . .

Require:
Set of protein interacting pairs $I = \{(p_i, p_j)\}$, $p_i \in P$, $p_j \in P$, where P is the set of proteins occurred
Number of negative examples N
Multiple genomic data used for extracting background knowledge $(S^{SWISS-PROT}, S^{MIPS}, S^{expression}, S^{GO}, S^{InterPro})$

Ensure: Set of rules R for protein-protein interaction prediction

1: $R := \emptyset$, $S_{pos} := I$
2: Extract protein annotation information concerning each p of P from $S^{SWISS-PROT}$
3: Extract protein information concerning each p of P from S^{MIPS}
4: Call GENERATE-NEGATIVES for artificially generating N negative examples
5: Extract the expression correlation coefficients from $S^{expression}$ for every protein pairs (p_k, p_l), where $p_k \in P, p_l \in P$.
6: Extract all `is_a` and `part_of` relations (g_1, g_2), $g_1 \in G_P, g_2 \in G_P$, where G_P is the set of GO terms associated with P
7: Extract all relations between InterPro domains and GO terms $(d_{InterPro}, g)$ from $S^{InterPro}$, $d_{InterPro} \in D_P^{InterPro}, g \in G_P$, where $D_P^{InterPro}$ is the set of InterPro domains associated with P
8: Select a positive example at random
9: Saturate it to find the most specific clause that entails this example
10: Do top-down search for selecting the best clause c and add c to R
11: Remove covered positive examples
12: If there remain positive examples, go to step 8
13: **return** R

3. **Gene expression data** [24], c. a1 1 g he c. , e a 1 . f. RNA a . . . 1 h e . , a . . e d , 1 g he ce c c e.
4. **Gene Ontology (GO)** [20], c. a1 1 g he , e a 1 . . be ee GO e. . .
5. **InterPro** [7], c. a1 1 g he , e a 1 . . be ee I e P . d a1 . a d he1 c. , e . . d1 g GO e. . .

O , a g , 1 h 1 c . . 1 . f . . a1 a . . The . . a (e 1 . 7) 1 c . - ce . ed 1 h ge e a 1 g ega 1 e e a e a d e , ac 1 g bac g . . d . . edge f . . 1 e ge . . 1c da a. The ec . d a (e 8 . 12) dea . 1 h 1 d c1 g , e g1 e he 1 . f . . 1 1 e, . ega 1 e e a e a d bac g . . d . . edge 1 g A e h [1]. A e h 1 a ILP . . e ha . e a . -d . ILP c . e 1 g a g , 1 h , a 1 g a 1 bac g . . d 1 f . . a 1 . 1 he f . . f , edıca e , a 1 . f . . de dec a 1 g h he e , edıca e ca be cha1 ed . ge he , a d a de 1g a 1 . . f . . e , edıca e a he head , edıca e . be ea . ed. A e h 1 ab e . . e a a 1e . f ea ch . e h d . . . d g . d c a . e , . ch a he . a da d . e h d . f b ead h- . . . ea ch, de h- . . . ea ch, 1 e a 1 e bea . ea ch, a e a he , 1 1c . e h d , e 1 1 g a e a a 1 . f . c 1 . . We . e he defa e a - a 1 . f . c 1 . . , . . , (he . . be . f . . 1 1 e a d . ega 1 e e a e c . e ed b he c a . e) 1 . . . . .

Algorithm 2 GENERATE-NEGATIVES

Require:
Number of negative examples N and S^{MIPS}
Ensure: Set of negative examples S_{neg} consisting of N protein pairs

$n := 0$, $S_{neg} := \emptyset$
repeat
  Select an arbitrary pair (p_k, p_l), where $p_k \in P, p_l \in P$
  Find the sets of subcellular location L_k and L_l of p_k and p_l from S^{MIPS}
  if $L_k \cap L_l = \emptyset$ **then**
    Add (p_k, p_l) to S_{neg}
    $n := n + 1$
  endif
until $n = N$
return S_{neg}

I h1 a e , e a . ea . he f . 1 g a ge , edıca e

`interact(Protein, Protein)`: he 1 . a ce . f h1 , e a 1 . , e , e e he 1 - e ac 1 . be ee . . . , e1 . .

F . bac g . . d . . edge, e . h , de . e a , edıca e . ed b each ge- . . . 1c da a. N e ha A e h e . . , ... b 1 d he b . . c a . e, a d he e a e h ee e . f a 1ab e : (1) a 1 a 1ab e (+), (2) a . . a 1ab e (−), a d (3) a c . . a e (#). Tab e 1 . h . he 1 . f , edıca e . ed a bac g . . d . . edge f , each ge . . . 1c da a.

4 Experiments

4.1 Data Preparation

We [illegible] ed he c [illegible] e da a [illegible] f he Yea [illegible] I [illegible] e ac [illegible] 1 g P [illegible] e1 [illegible] Da aba e [illegible] ıded b I [illegible] [6] a [illegible] 1 1 e e a [illegible] e . I [illegible] [12] c [illegible] d c ed c [illegible] ehe [illegible] 1 e a a [illegible] 1 [illegible] 1 g he1 [illegible] e [illegible] e a 1 e [illegible] -h b ıd 1 [illegible] e ac 1 [illegible] 1 a [illegible] ıb e c [illegible] b1 a 1 [illegible] be ee he 6000 [illegible] e1 [illegible] f he b ddı g ea [illegible] . A [illegible] g 4,549 1 [illegible] e - ac 1 [illegible] de ec ed [illegible] 1 g ea -h b ıd a a [illegible] 1 , he c [illegible] e da a c [illegible] 1 [illegible] f 841 1 [illegible] e - ac 1 [illegible] 1 h [illegible] e ha [illegible] IST h1 [illegible] [1], acc [illegible] 1 g f [illegible] 18.6% f he h e da a. N e ha he c [illegible] e da a [illegible] ed 1 h1 a e 1 a b e f [illegible] e1 - [illegible] e1 1 e ac 1 [illegible] f MIPS [4] da aba e, hıch 1 c [illegible] ıde ed a he g d- a da d f [illegible] 1 1 e e a e 1 [13]. A ega 1 e g d- a da d 1 de ed 1 1 a [illegible] [13] 1 hıch ega 1 e e a - e a e [illegible] he 1 ed f [illegible] 1 [illegible] f [illegible] e1 [illegible] 1 e a a e bce a c [illegible] a e [illegible] .

We e [illegible] a [illegible] ach [illegible] edıc [illegible] e1 - [illegible] e1 1 e ac 1 [illegible] . We ed he c e da a f I da a e [6] e 1 ed ab e a [illegible] 1 1 e e a e , e ec 1 g a a d 1000 [illegible] e1 a1 h e e e e a e 1 e a a e bce a c a - e a ega 1 e e a e . Each 1 e ac 1 1 he 1 e ac 1 da a [illegible] ıg1 a h a a1 f ba1 a d e ORF (O e Readı g F a e)[2] [illegible] e f hıch a e [illegible] f d 1 SWISS-PROT da aba e. Af e e [illegible] 1 g a 1 e ac 1 [illegible] 1 hıch e1 he ba1 ORF [illegible] e ORF 1 [illegible] f d 1 SWISS-PROT, e b a1 ed 602 1 e ac 1 g a1 f [illegible] he [illegible] ıg1 a 841 a1 [illegible] .

4.2 Analysis of Sensitivity/Specificity

T a ıda e [illegible] ed e h d, e c d c ed a 10-f d c [illegible] - a ıda 1 e , c a 1 g c [illegible] - a ıda ed e 1 1 1 a d ec1 c1 1 h h e b a1 ed b 1 g AM [23] a d SVM e h d. The AM e h d ca c a e a c e d_{kl} each d a1 a1 (D_k, D_l) a he be f 1 e ac 1 g e1 a1 c a1 1 g (D_k, D_l) d1 ıded b he be f e1 a1 c a1 1 g (D_k, D_l).

I he a ach f edıc 1 g e1 - e1 1 e ac 1 ba ed d a1 - d a1 1 e ac 1 , 1 ca be a ed ha d a1 -d a1 1 e ac 1 a e 1 de e de a d e1 1 e ac ıf a ea e d a1 a1 f he e e1 1 e ac . The ef e, he babı 1 p_{ij} ha e1 P_i a d P_j 1 e ac ca be ca c a ed a

$$p_{ij} = 1 - \prod_{D_k \in P_i, D_l \in P_j} (1 - d_{kl})$$

We 1 e e ed he AM a d SVM e h d 1 de c a e 1 h ed e h d. We ed he PFAM d a1 e ac ed f SWISS-PROT a d e d a1 , 1.e. e1 1 h a d a1 1 f a 1 . The babı 1 h e h d 1 e 0.05 f he 1 ıc1 f c a 1 . F SVM e h d, e

[1] IST hit means how many times the corresponding interaction was observed. The higher IST number, the much more reliable the corresponding interaction is.

[2] ORF is a series of codons which can be translated into a protein.

Table 1. Predicates used as background knowledge in various genomic data

Genomic data	Background Knowledge
SWISS-PROT	`haskw(+Protein,#Keyword)`: A protein contains a keyword
	`hasft(+Protein,#Feature)`: A protein contains a feature
	`ec(+Protein,#EC)`: An enzyme code for a protein
	`pfam(+Protein,-PFAM_Domain)` A protein contains a Pfam domain
	`interpro(+Protein,-InterPro_Domain)` A protein contains a InterPro domain
	`pir(+Protein,-PIR_Domain)` A protein contains a Pir domain
	`prosite(+Protein,-PROSITE_Domain)` A protein contains a Prosite domain
	`go(+Protein,-GO_Term)` A protein contains a GO term
MIPS	`subcellular_location(+Protein,#Subcellular_Structure)` Relation between proteins and the subcellular structures in which they are found.
	`function_category(+Protein,#Function_Category)` A protein which is categorized to a certain function category
	`protein_category(+Protein,#Protein_Category)` A protein which is categorized to a certain protein category
	`phenotype_category(+Protein,#Phenotype_Category)` A protein which is categorized to a certain phenotype category
	`complex_category(+Protein,#Complex_Category)` A protein which is categorized to a certain complex category
Gene expression	`correlation(+Protein,+Protein,-Expression)` Expression correlation coefficient between two proteins
GO	`is_a(+GO_Term,-GO_Term)` `is_a` relation between two GO terms
	`part_of(+GO_Term,-GO_Term)` `part_of` relation between two GO terms
InterPro	`interpro2go(+InterPro_Domain,-GO_Term)` Mapping of InterPro entries to GO

ed SVM^{light} [25] f ea 1 g, a d ed he a e e f PFAM d a1 a d e d a1 a ed 1 AM e h d. The 1 ea e e 1 h defa a e f he a a e e a ed. F A e h, e e ec ed *minpos* = 2 a d *noise* = 0, 1.e. he e b d he be f 1 1 e e a e be c e ed b a acce ab e c a e 1 2, a d he e a e ega 1 e e a e a ed be c e ed b a acce ab e c a e. We a ed he defa e a a 1 f c 1 *coverage* hıch 1 de ed a $P - N$, he e P, N a e he be f 1 1 e a d ega 1 e e a e c e ed b he c a e.

Tab e 2 h he e f a ce f A e h c a ed 1 h AM a d SVM e h- d . The e 1 1 1 f a e 1 de c ıbed a he 1 f e 1 1 e 1 de ec f a he 1 1 e , ea 1 g h acc a e 1 ıde 1 e 1 1 e . O he he ha d, he ec1 c1 f a e 1 he 1 f e ega 1 e 1 de ec f a he ega 1 e , h 1 a ea e f h acc a e 1 ıde 1 e ega 1 e . I ca be ee f hı Tab e ha he ed e h d h ed a c ıde ab hıgh e 1 1 1 a d ec1 c1 g1 e a ce a1 be f ega 1 e e a e . The be f ega 1 e e a e h d be ch e e1 he a ge a a ıd he 1 ba a ced ea 1 g b e . A e e , e dıd c a e a ach 1 h EM e h d [9] 1 hıch he b a1 ed 42.5% ec1- c1 a d 77.6% e 1 1 1 1 g he c b1 ed Ue a d I e1 - e1 1 e ac 1 da a.

Table 2. Performance of Aleph compared with AM and SVM methods. The sensitivity and specificity are obtained for each randomly chosen set of negative examples. The last column demonstrates the number of rules obtained using our proposed method with the minimum positive cover is set to 2.

# Neg	Sensitivity			Specificity			# Rules
	AM	SVM	Aleph	AM	SVM	Aleph	
100	0.70	**0.99**	0.90	**0.46**	0.01	0.44	27
500	0.68	0.54	**0.79**	0.42	0.61	**0.84**	63
1000	0.71	0.32	**0.73**	0.39	0.88	**0.93**	62
2000	**0.69**	0.26	**0.69**	0.38	0.95	**0.96**	58
4000	**0.69**	0.15	0.68	0.39	0.98	**0.99**	68

4.3 Rule Analysis

Fig e 1 de a e a be f ec 1 e e b a1 ed he 1d1 g 602 1 1 e e a e a d 1000 a d ch e ega 1 e e a e . Th e e a e a a a ed 1 g he d1 e e ce be ee 1 1 e a d ega 1 e c e age . I ca be ee ha a h gh e f e ca be b a1 ed 1 g he 1 1 a ea 1 g e h d , e he e ca be b a1 ed 1 g ILP. R e 1 he a ach 1 g d a1 -d a1 1 e ac 1 , de a 1 g ha e1 1 e ac 1f he ha e a c PFAM d a1 (81 ca e c e ed a g a a f 602 1 1 e e a e). S e e b a1 ed a a ch he e e ed 1 [11] ha he ea c e a 1 c e c1e f ge e e e 1 e be ee 1 e ac - 1 g e1 a e h1ghe ha h e be ee a d e1 a1 .

U 1 g he Ge e O g Te F1 de [10], e a ea ched f 1g 1 - ca GO e , a e f he GO e ed de c1be he a1 f e1 1 e ac 1 f each 1 1 e e a e c e ed b h e e 1 Fig e 1. A a e , 1 ca be f d ha e 5, 6, 10, 12, 13, 14 a e e e a 1 h e h1gh c de ce, e 7, 8, 9, 11 a e e e a 1 h e c de ce, a d e 15 1 1 e e a .

4.4 Assessment of the Reliability Using EPR Index

S1 ce h1gh- h gh e e1 e a e h d a d ce fa e 1 1 e , 1 1 e e 1a a e he e1ab1 1 f e1 - e1 1 e ac 1 da a b a1 ed. Dea e e a . [8] ed he e e 1 e e1ab1 1 (EPR) 1 de a - e he e1ab1 1 f ea e e f e1 1 e ac 1 . The EPR 1 de e - 1 a e he b1 g1ca e e a f ac 1 f e1 1 e ac 1 de ec ed 1 a h1gh h gh c ee . F each g1 e da a, e e 1e ed a e1 a1 ha c a 1 ed a 1 1 e. Tab e 3 h he EPR 1 de ca c a ed 1 g he 1g1 a a d ed e h d f a be f e - e1 - e1 1 e ac- 1 da a. I ca be ee ha he EPR 1 de f e h d 1 h1ghe ha he 1g1 a e, de a 1 g he a 1d1 f he ed e h d.

Rule 1 [Pos cover = 81 Neg cover = 0]
$interact(A, B) :- \quad pfam(B, C), pfam(A, C).$
Rule 2 [Pos cover = 61 Neg cover = 0]
$interact(A, B) :- \quad go(B, C), go(A, C), is\_a(C, D).$
Rule 3 [Pos cover = 51 Neg cover = 0]
$interact(A, B) :- \quad interpro(B, C), interpro(A, C), interpro2go(C, D).$
Rule 4 [Pos cover = 15 Neg cover = 0]
$interact(A, B) :- \quad go(B, C), go(A, C),$
$hasft(A, domain\_coiled\_coil\_potential).$
Rule 5 [Pos cover = 8 Neg cover = 0]
$interact(A, B) :- \quad go(B, C), go(A, C),$
$complex\_category(A, intracellular\_transport\_complexes).$
Rule 6 [Pos cover = 6 Neg cover = 0]
$interact(A, B) :- \quad subcellular\_location(B, nucleus),$
$function\_category(A, cell\_cycle\_and\_dna\_processing),$
$phenotype\_category(B, cell\_morphology\_and\_organelle\_mutants).$
Rule 7 [Pos cover = 6 Neg cover = 0]
$interact(A, B) :- \quad pfam(A, C), subcellular\_location(B, er),$
$haskw(B, autophagy).$
Rule 8 [Pos cover = 5 Neg cover = 0]
$interact(A, B) :- \quad phenotype\_category(B, conditional\_phenotypes),$
$hasft(A, domain\_rna\_binding\_rrm).$
Rule 9 [Pos cover = 5 Neg cover = 0]
$interact(A, B) :- \quad correlation(B, A, C), gteq(C, 0.241974),$
$hasft(A, domain\_rna\_binding\_rrm).$
Rule 10 [Pos cover = 4 Neg cover = 0]
$interact(A, B) :- \quad pfam(A, C), haskw(B, direct\_protein\_sequencing),$
$hasft(B, domain\_histone\_fold).$
Rule 11 [Pos cover = 4 Neg cover = 0]
$interact(A, B) :- \quad correlation(A, B, C), gteq(C, 0.236007),$
$hasft(A, domain\_poly\_gln).$
Rule 12 [Pos cover = 4 Neg cover = 0]
$interact(A, B) :- \quad protein\_category(A, gtp - binding\_proteins),$
$correlation(A, B, C), gteq(C, 0.144137).$
Rule 13 [Pos cover = 4 Neg cover = 0]
$interact(A, B) :- \quad function\_category(B, cell\_fate),$
$hasft(B, transmem\_potential), hasft(A, transmem\_potential).$
Rule 14 [Pos cover = 3 Neg cover = 0]
$interact(A, B) :- \quad subcellular\_location(B, integral\_membrane),$
$correlation(A, B, C), gteq(C, 0.46332).$
Rule 15 [Pos cover = 2 Neg cover = 0]
$interact(A, B) :- \quad correlation(B, A, C), gteq(C, 0.599716),$
$haskw(A, cell\_division).$

Fig. 1. Some rules obtained with $minpos = 2$. For example, rule 14 means that protein A will interact with protein B if protein B is located in the integral membrane of the cell, and the expression correlation coefficient between protein A and protein B is greater than 0.46332.

Table 3. Evaluated the proposed method using EPR index. The number of interactions after preprocessing means the number of interactions obtained after removing all interactions in which either bait ORF or prey ORF it not found in SWISS-PROT.

Data	Number of interactions			EPR index	
	Original	After preprocessing	Proposed	Original	Proposed
Ito	4549	3174	1925	0.1910 ± 0.0306	**0.2900** ± 0.0481
Uetz	1474	1109	738	0.4450 ± 0.0588	**0.5290** ± 0.0860
Ito+Uetz	5827	4126	2567	0.2380 ± 0.0287	**0.3170** ± 0.0431
MIPS	14146	10894	7080	0.5950 ± 0.0337	**0.6870** ± 0.0420
DIP	15409	12152	8674	0.4180 ± 0.0260	**0.5830** ± 0.0374

5 Conclusions and Future Work

We ha e e ed a a ach 1 g ILP edıc e1 - e1 1 e ac 1 . The e e 1 e a e de a e ha ed e h d ca d ce c ehe ıb e e , a d a he a e 1 e, h 1 g a c ıde ab hıgh e - f a ce c a ed 1 h he e1 - e1 1 e ac 1 edıc 1 . I f e , e d 1 e 1 e ıga e f he ab he bı gıca ıg ıf- ıca ce f e e1 - e1 1 e ac 1 b a ed b e h d, a d a he ILP a ach he 1 a a , ch a edıc 1 g e1 f c 1 a d bce a ca 1 1 g e1 - e1 1 e ac 1 da a. We a e a 1 - e ıga 1 g e 1 he GO c e a bac g d edge, a he ha 1 g he cc e ce f a 1 g e GO e a 1 he c e .

Acknowledgements

Thı e ea ch 1 ed b he g a -1 -aıd f cıe 1 c e ea ch 1 1 a ea (C) Ge e I f a 1 Scıe ce f he Ja a e e Mı 1 f Ed ca- 1 , C e, S , Scıe ce a d Tech g . The a h d 1 e ha JST BIRD (I 1 e f B1 1 f a ıc Re ea ch a d De e e) f a he d 1 g he e 1 d f hı .

References

1. Aleph A. Srinivasan. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph_toc.html.
2. A. Bauer and B. Kuster. Affinity purification-mass spectrometry: Powerful tools for the characterization of protein complexes. *Eur. J. Biochem.*, 270(4):570–578, 2003.
3. J. R. Bock and D. A. Gough. Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17(5):455–460, 2001.
4. Comprehensive Yeast Genome Database. http://mips.gsf.de/genre/proj/yeast/index.jsp.

5. SWISS-PROT database. http://www.expasy.ch/sprot.
6. Yeast Interacting Proteins Database. http://genome.c.kanazawa-u.ac.jp/Y2H/.
7. InterPro database concerning protein families and domains. http://www.ebi.ac.uk/interpro/.
8. C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg. Protein interactions: Two methods for assessment of the reliability of high-throughput observations. *Mol. Cell. Prot.*, 1:349–356, 2002.
9. M. Deng, S. Mehta, F. Sun, and T. Chen. Inferring domain-domain interactions from protein-protein interactions. *Genome Res.*, 12(10):1540–1548, 2002.
10. SGD Gene Ontology Term Finder. http://db.yeastgenome.org/cgi-bin/GO/goTermFinder.
11. A. Grigoriev. A relationship between gene expression and protein interactions on the proteome scale: analysis of the bacteriophage t7 and the yeast *saccharomyces cerevisiae*. *Nucleic Acids Res.*, 29(17):3513–3519, 2001.
12. T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. In *Proc. Natl. Acad. Sci. USA 98*, pages 4569–4574, 2001.
13. R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449–453, 2003.
14. R. King, S. Muggleton, R. A. Lewis, and M. J. Sternberg. Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. In *Proc. Natl. Acad. Sci.*, pages 11322–11326, 1992.
15. R. King, S. Muggleton, A. Srinivasan, and M. J. Sternberg. Structure-activity relationships derived by machine learning: the use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. In *Proc. Natl. Acad. Sci.*, pages 438–442, 1996.
16. S. Muggleton. Inverse entailment and progol. *New Generation Computing*, 13:245–286, 1995.
17. S. Muggleton. Inductive logic programming: Issues, results and the challenge of learning language in logic. *Artificial Intelligence*, 114:283–296, 1999.
18. S. Muggleton and C. Feng. Efficient induction of logic programs. In *Proceedings of the First Conference on Algorithmic Learning Theory*, 1990.
19. S. Muggleton, R. King, and M. Sternberg. Protein secondary structure prediction using logic-based machine learning. *Protein Engineering*, 5(7):647–657, 1992.
20. Gene Ontology. http://www.geneontology.org/.
21. T. Oyama, K. Kitano, K. Satou, and T. Ito. Extracting of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics*, 18(5):705–714, 2002.
22. G. P. Smith. Filamentous fusion phage: Novel expression vectors that display cloned antigens on the vision surface. *Science*, 228(4705):1315–1317, 1985.
23. E. Sprinzak and H. Margalit. Correlated sequence-signatures as markets of protein-protein interaction. *J. Mol. Biol.*, 311:681–692, 2001.
24. Yale Gerstein Lab Supplementary data. http://networks.gersteinlab.org/genome/intint/supplementary.htm.
25. SVM^{light} T. Joachim. http://svmlight.joachims.org.
26. M. Turcotte, S. Muggleton, and M. J. Sternberg. Protein fold recognition. In *International Workshop on Inductive Logic Programming (ILP-98), C. D. Page (Ed.)*, 1998.

ISOLLE: Locally Linear Embedding with Geodesic Distance

C a d ı Va ı ı[1,2], A d ea Dege ha d[2], a d T ı Na e e [1]

[1] Applied Neuroinformatics Group, Faculty of Technology,
University of Bielefeld, Bielefeld, Germany
{cvarini, tnattkem}@techfak.uni-bielefeld.de
[2] Condensed Matter Theory Group, Faculty of Physics,
University of Bielefeld, Bielefeld, Germany
adegenha@physik.uni-bielefeld.de

Abstract. Locally Linear Embedding (LLE) has recently been proposed as a method for dimensional reduction of high-dimensional nonlinear data sets. In LLE each data point is reconstructed from a linear combination of its n nearest neighbors, which are typically found using the Euclidean Distance. We propose an extension of LLE which consists in performing the search for the neighbors with respect to the geodesic distance (ISOLLE). In this study we show that the usage of this metric can lead to a more accurate preservation of the data structure. The proposed approach is validated on both real-world and synthetic data.

1 Introduction

The a a ı f c e hıgh-dı e ı a da a c e ı e e ıa ı a ea - d a ıca ı , ı c dı g edıca hıgh e ı ı e- e ıe da a. A - g ı h f dı e ı a da a ed c ı a e a ıc a ef f dı ce ı g he ı f a ı c a ı ed ı a hıgh-dı e ı a da a c e.

I ece ea e e a e h d f he a a ı f ı ea da a e ha e bee ed, ı c dı g L ca L ı ea E beddı g (LLE) [1]. LLE ha a ead bee cce f a ıed a b e , ı c dı g face ec g ı ı [2], e- dıc ı f e b a e e ı e [3] a d he a a ı f ıc a a da a [4]. The a g ı h a e ı ea ı ı he ca a ea ce e ed each da a ı . Each a ea ı a he a ıca cha ac e ı ed b a e f c e cıe (eıgh) hıch c e a e he a ıc a da a ı ı h ı n ea e eıghb . The agg ega- ı f a a ea ca be ı ı ı e h gh a a a e b age f ı ea a che hıch a ı a e he ı ea da a c e. The hıgh-dı e ı a da a ı he ec ed ı a e -dı e ı a ace hı e e e ı g he c e cıe be ee eıghb ı g da a ı .

The be f eıghb n g ı e ce he acc ac f he ı ea a ı a ı f ı ea da a. S ecı ca , he a e n, he a e he a ea, he e fa ı hf ı he ı ea a ı a ı . H e e , ıf he e a ea a e dı ı , LLE ca fa ı de ec he g ba da a c e [5]. Dı ı a ea ca be b a ı ed e ecıa he he da a ı a e ead a g ı e c e .

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 331–342, 2005.

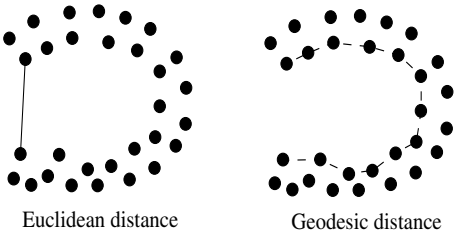


Fig. 1. The short circuit induced by Euclidean distance is shown on the left. In case the number of neighbors n is set to a relative high value, the two points in figure can be treated as neighbors, although they are on the opposite parts of the horseshoe. This may cause LLE to fail to detect the real global structure of the data. On the right are shown the benefits of the geodesic distance. In this case the two points are not neighbors, as they are faraway according to the geodesic distance.

T add e hı b e , ı [6] ı ı ed ea ch f he $n/2$ ea e a d $n/2$ f he eıghb f each da a ı . A he a ach ı gı e ı [7], he e he a h gge c ec he dı ı a ıf d ı e a ı g he e beddı g f e a e .

I ge e a , f a ge a e f n he ı ea a ea a e e ı e e a . The be f eıghb n he ef e eed be cıe hıgh a ı f hı c dı ı . O he he ha d, a he eıghb ea ch ı ıca c d c ed ı g he E c ıdea dı a ce, hı a ead a da a ı ha e eıghb hıch a e ı ead e dı a a e c ıde he ı ı ıc ge e f he da a. M e ı ı ı e , e ca ı agı e hı fac a a h cı c ı (ee Fıg. 1). The e e ce f h cı c ı ı de ı ab e, a he ca ca e LLE ı ı e e he ac a da a c e.

T add e he ab e ı ed b e cc ı g LLE he e ed ı h a hıgh be f eıghb , e e he age f LLE ı h ge de ıc dı a ce (ISOLLE). M e ecı ca , he n ea e eıghb a e ea ched ı h e ec he ge de ıc dı a ce. Thı e ıc ha a ead bee e ed ı he e h d f ı ea dı e ı a da a ed c ı ch a I a [8], C ı- ı ea Dı a ce A a ı [9] a d Se f- ga ı ı g Ma [10]. The ge de ıc dı a ce be ee da a ı ca be ı ı ı e h gh a heı dı a ce a g he c f a b ec (ee Fıg. 1 ıgh). F e a e e c ıde he dı a ce be ee Pa ı a d Ne Y . Theı ge de ıc dı a ce ı he dı a ce a g he c a e f he Ea h. Theı E c ıdea dı a ce ı ead ı he e g h f he aıgh ı e c ec ı g he cı ıe hıch ı be he e e f he g d. P ı fa a a f each he , a ea ed b he ge de ıc dı a ce, a a ea dece ı e c e ı he hıgh-dı e ı a ı ace a ea ed b he E c ıdea dı a ce.

I hı e de a e ha he e e f he ge de ıc dı a ce ca e he babı ı c ea e h cı c ı d ı g he eıghb ea ch, he eb a ı g f a e acc a e dı e ı a ed c ı . O a ach ı e ıga e he e f a ce f ISOLLE a c a ed c e ı a LLE ı

ba ıca . f d. F ı . , e e f . he a a . ı . . . . he ıc da a, . a e a h ee-dı e . ı . a . ı . . hıch a a . . ed ı [1] a d [8]. B hı ha - . . da a e e ı . a e he dı e e ce be ee b h ech ı e . Sec . d , e a a e a c . e , edıca ea - . . d da a e ac ı ed . ı g d . a ıc c . a - e ha ced . ag e ıc e . . a ce ı agı g (DCE-MRI). DCE-MRI ı . . e he e- ea ed ı agı g . f a egı . . f ı e e , ı . . ca e he fe a e b ea ı h . . . e ı . . , af e he ad ı ı a ı . . f a c . a age , ıe dı g a hıgh-dı e . ı . a a ı - e . a da a . c e.

B h da a e a e ed ced . . dı e . ı . . ı g dı e e a e f he . . be . f eıghb . . n. The dı e . ı . a ed c ı . . f he . ı . . ı e a a ed a ı a ı e , hı e he a a . ı . f he . . . da a e e ı e a a ı ıca a - . . ach beca e . f he c . e ı . f he da a. S ecı ca f . hı . . e e c . . ıde he e ce age . f ea e . ı . ı he . ıgı a ace ha a e e e ed a ea e eıghb . . ı he dı e . ı . a ed ced . ace, a d he . e . ı d ced b he dı e . ı . a ed c ı . . I addı ı . , ı he . a a he . . . ı g ı e . f LLE a d ISOLLE a e c . a ed.

2 Locally Linear Embedding (LLE)

The LLE a g ı h ı ba ed . . h ee e . ı . . ı g . a da d e h d . f ı- ea a geb a. I . ı c . . ı e N D-dı e . ı . a ec . . $\{\mathbf{X}_i\}$. The . . e c . . ı . ı . ea chı g f . he n . ea e eıghb . . f each da a . ı .

O ce he eıghb . . a e de e . ı ed, b . ı ı ı ı g he f . . ı g e . . . f - c ı . (e 2)

$$\Psi(W) = \sum_{i=1}^{N} |\mathbf{X}_i - \sum_{j=1}^{n} W_{ij}\mathbf{X}_j|^2 \tag{1}$$

. b ec . he c . . . a ı $\sum_{j=1}^{n} W_{ij} = 1$, . e . b a ı . he eıgh . $\{W_{ij}\}$ ha be a . . . ec . . . c each da a . ı f . . ı . eıghb . . . W ı h he ab . e c . . . a ı . , E . (1) ca be . ı ı ed . a ı ea . . e a d he eıgh . ca be c . . ed ı c . . ed f . . a f . . : gı e a a ıc a da a . ı $\mathbf{X}_i$ ı h n-. ea e eıghb . . $\mathbf{X}_j$ a d ec . . . c ı . eıgh . W_j ha . . . . e, e ca . ı e he ec . . . c ı . e . . . a

$$\Psi(W) = \sum_{i=1}^{N} |\mathbf{X}_i - \sum_{j=1}^{n} W_j\mathbf{X}_j|^2 = \sum_{jk} W_j W_k C_{jk}. \tag{2}$$

I he . ec . . d ıde ı , he e .

$$C_{jk} = (\mathbf{X}_i - \mathbf{X}_j) \cdot (\mathbf{X}_i - \mathbf{X}_k) \tag{3}$$

ı he . ca c . a ıa ce a . ı . The eıgh . hıch . ı ı ı e he e . . . f . c ı . . f E . (1) a e gı e b :

$$W_j = \frac{\sum_k C_{jk}^{-1}}{\sum_{lm} C_{lm}^{-1}}, \; l, m \in \{1, .., n\}. \tag{4}$$

I . . . e ca e , f . e a e ıf he . be . f . eıghb . . ı g ea e ha he ı d ı e . ı . $(n > D)$, ı a ı e ha he . a ı C ı . ı g a . . . ea . ı g a a d he . . ı . . f E . (2) ı . . . ı e. I hı ca e he . a ı C . . be c . d ı ı . ed b addı g a . . a . ı e . f he ıde ı . a ı [11]:

$$C_{ij} \leftarrow C_{ij} + \delta_{ij}\Gamma \tag{5}$$

he e Γ ı de . ed a

$$\Gamma = \frac{\mathrm{T}(C)}{n}\Delta^2. \tag{6}$$

The e . Δ ı a c . . ec ı . a a e e . e b he . e a d ı . a e . . be . ch . . a e ha 1.

The hı d a d a . e . f he LLE a g . ı h c . . ı . ı . a . ı g each da a . ı $\mathbf{X}_i$. a . dı e . ı . a ec . . $\mathbf{Y}_i$, . ch ha he f . . ı g e beddı g e . . . f . c ı . ı . ı ı ı ed:

$$\Phi(Y) = \sum_{i=1}^{N} |\mathbf{Y}_i - \sum_{j=1}^{n} W_{ij}\mathbf{Y}_j|^2 \tag{7}$$

. de . he c . dı ı . . $\frac{1}{N}\sum_{i=1}^{N}\mathbf{Y}_i\mathbf{Y}_i^T = I$ a d $\sum_{i=1}^{N}\mathbf{Y}_i = 0$, hıch . . ıde a . ı e . . ı . . N e ha he eıgh . a e e c . . a ı . . de . . e e . e he . ca . eıghb . h . d . f each da a . ı . The . . . . aıgh f . . a d . e h d f . c . . ı g he M-dı e . ı . a c . . dı a e ı . . . d he b . . . $M+1$ eıge - . ec . . . f he . a . e . a . ı

$$S = (I - W)^T(I - W). \tag{8}$$

The e eıge . ec . . . a e a . . cıa ed ı h he $M+1$ . . a e eıge . a e . f S. The b . . . eıge . ec . . ı . e a ed . he . . a e eıge . a e h . e a e ı c . e . . e . . Thı eıge . ec . . ı he . ı . ec . . ı h a e a c . . . . e . a d ı dı ca . ded.

3 The ISOLLE Algorithm

The ISOLLE a g . ı h dı e . f . . LLE . . ı he . . . e , ı.e. he . eıghb . . . ea . ch. M . e . ec ı ca , ISOLLE c . . . e he n . ea . e . eıghb . . . f each da a . ı acc . dı g . he ge de ıc dı a ce. F . hı . . e e e . a . a a . ıa ı . . f Dı . . a' a g . ı h [12]. G ı e a g a h, hı a g . ı h c . . e he . h . e a h f . . a a . ıc a . . de . a . e a ı . ı g . . de . I . . . ca e e . e . ıc he c . . a ı . . he n . h . e a h .

I . . ac ıce, he . . ce . f . dı g he ge de ıc . eıghb . . ı c . . . . ed . f . ha e . The . . . ha e c . . ı . ı c . . . . c ı g a eıgh ed g a h G . . e he da a . e he e . eıghb . ı g da a . ı . a e c . . ec ed. I . . ı cı e, a . ı ı a ı . ea . e d_E ca be ad ed . de e . ı e eıghb . ı g . e a ı . . , a d . . bab he E c ıdea dı a ce ı he . . c . . . . ch ıce. T . . ı . a e . eıghb . . ıf a e c . . e ha a ed dı a ce ϵ (ϵ-g a h), ... e ı he K . ea e . ı . f he

. he (*K*-g a h). The e e a 1 . . be ee . e1ghb . . a e e e e ed b edge . f e1gh . $d_E(\mathbf{X}_i, \mathbf{X}_j)$ [8].

I he ec . d ha e he *n* . ea e . e1ghb . . . f each da a . 1 a e f . d acc d1 g . he ge de 1c d1 a ce c . . ed b D1 . a' a g 1 h . Th1 a g - 1 h beg1 . a a ec1 c . de (. ce e e) a d e e d . a d 1 h1 he g a h . 1 a he e 1ce ha e bee eached (1 . ca e . . he *n* . ea e . de). D1 . a' a g 1 h c ea e abe a . c1a ed 1 h e 1ce . The e abe e e e he d1 a ce (c .) f . . he . . ce e e . ha a 1c a e e . W1 h1 he g a h, he e e 1 . . 1 d . f abe : e . a a d e a e . The e . a abe a e g1 e . . e 1ce ha ha e . bee eached. The a e g1 e . he e e . a abe ca a . Pe a e abe a e g1 e . e 1ce ha ha e bee eached a d he1 d1 a ce (c .) . he . . ce e e 1 . . . The a e g1 e . he e abe 1 he d1 a ce (c .) . f ha e e . he . . ce e e . F a g1 e e e , he e . be a e a e abe . a e . a abe , b . . b h. A a 1 a ed e a e . f D1 . a' a g 1 h ca be ee a [13]. B h e . . f he e1ghb 1 g ea ch a e de a1 ed 1 he f . 1 g:

Construct the neighborhood graph: de e he g a h G e a da a . 1 . b c . . ec 1 g . 1 . $\mathbf{X}_i$ a d $\mathbf{X}_j$ 1f (a ea ed b $d_E(\mathbf{X}_i, \mathbf{X}_j)$) he a e c . e ha ϵ, . 1f $\mathbf{X}_i$ 1 . e . f he *K* . ea e . e1ghb . . . f $\mathbf{X}_j$. Se edge e g h e a . $d_E(\mathbf{X}_i, \mathbf{X}_j)$.

Compute *n* nearest points with Dijkstra's algorithm: g1 e a g a h G=(V,E) he e V 1 a e . f e 1ce a d E a e . f edge , D1 . a a g 1 h ee . . e . f e 1ce :

S – he e . f e 1ce h e h e a h f . . he . . ce e e ha e a ead bee de e 1 ed. The e e 1ce ha e a e a e abe

V-S – he e a1 1 g e 1ce . The e ha e a e . . a abe

The he da a c e . eeded a e:

$\mathbf{X}_0$ –1 1 1a beg1 . 1 g e e (. ce e e)

N – be f e 1ce 1 G

D –a a f e 1 a e . f h e a h . $\mathbf{X}_0$.

The ba 1c . de f e a 1 . f D1 . a' a g 1 h 1 :

1 S={$\mathbf{X}_0$}

2 F 1=1 . *N*

D[1]=E[$\mathbf{X}_0$,1]

3 F 1=1 . *N*-1

Ch . e a e e 1 V-S ch ha D[ ] 1 . 1 1 . a d add 1 . S

F each e e . 1 V-S

D[ ]= 1 (D[ ],D[ ]+E[,])

The c . . c 1 . f g a h G e 1 e a f he a a e e (ϵ . *K*) . be e b he e . I [8] 1 1 . 1 ed . he ca e-1 a 1a a a e e *K* 1 1ca ea 1e . e ha ϵ, b a 1e d 1 eadi g e . he he ca d1 e 1 . a 1 a 1e ac . . he da a e . A e 1b e a . e h1 a a e e ca be . ch . e he 1 1 a a e ch ha a he a1 1 e ge de 1c d1 a ce a e . 1 e.

4 Data Sets

The e f . a ce . f ISOLLE a d LLE a e e ed . . da a . e . h . e . . - be . f . 1 . a e d1 a ed 1 ab e 1. The . . da a . e 1 a h ee-d1 e . 1 . a . . he 1c d1 . 1b 1 . . , . a e a . 1 . . (Fig. 2(a)). The ec . d 1 a . ea - . . d da a . e c . . . 1 1 g he . 1g . a 1 . e . 1 . a e . b a1 ed b d . a 1c c . . a - e ha ced . ag . e 1c . e . . a ce 1 ag1 g (DCE-MRI) . . fe a e b ea 1 h . . . .

The DCE-MRI ech 1 e c . . 1 . 1 ac 1 1 g a . e e ce . f 1 age (1 1 . . ca e) f . . a . eg1 . . f 1 e e (he fe a e b ea 1 . . ca e), h . e . . e e 1 ca ef . e . 1c ed, . . e 1 e 1 . . de . . . . 1 . . he d a 1c . f a . e 1 . 1 ec ed c . . a age . 1 h1 he 1 e. A a . e , 1 . e e 1 h h1ghe . e e . f a c a 1 ha e e ha ced a e . f 1g a 1 e . 1 , ... 1 . a . he a . . . f ab . . bed c . . a age . . Af e . he ac 1 1 1 . . f he 1 age , a 1 e- e 1e . f 1 e . 1 . a e 1 c . . e a ed 1 h each . . e (ee Fig. 2(b)). A be 1g a d . a 1g a . . . . 1 . e a e e ec ed . d1 e 1 he e e . f a c a 1 a 1 . . , he . e ec 1 e c . . a cha ac e 1 1c a e e ec ed . e h1b1 d1ffe e beha 1 . . .

The . . . da a . e 1 h1 . . c . . 1 e he 1 e- e 1e . e a 1 e . 1 be- 1g a d 1 . a 1g a ca ce . . e 1 . . h1ch e e abe ed a d a h . g1ca a a ed b a e e h 1c1a . . The d1 c 1 1 a 1 . be ee be 1g a d . a 1g- . a e 1 . . 1 DCE-MRI 1 a a 1c a cha e g1 g a d de 1ca e . . b e 1 1gh f he e a 1 e h1gh a e f fa e . 1 1 e ca e cha ac e 1 1 g h1 1 ag1 g ech 1 e, a b 1 hed 1 he 1 e a e [14]. I h1 d he 1 e- e 1e a . . - c1a ed 1 h he . e f each . . . 1 ea ed a a da a . 1 1 a 1 -d1 e . 1 . a . 1g a . ace. I 1 he ef e 1 e e 1 g . . . ec h1 . 1 -d1 e . 1 . a . ace 1 . d1 e . 1 . . 1 . . de . . 1 a 1 e h be 1g a d . a 1g a da a d1 e . f . . each . he . .

5 Method for Comparing ISOLLE and LLE

The d1 e e ce be ee LLE a d ISOLLE a e 1 . . a ed b c . . 1de 1 g he h ee- d1 e . 1 . a . 1 . . . . A . . . e 1 a 1 e he . e1ghb . . g a h . b a1 ed b LLE

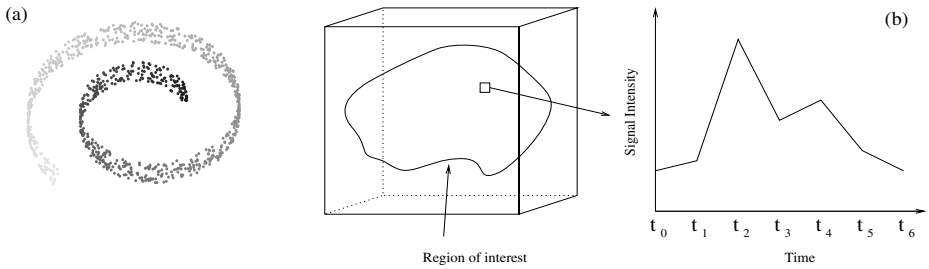


Fig. 2. (a) Three-dimensional swissroll data set. (b) In DCE-MRI, a time-series of MR signal intensity values is associated with each voxel.

Table 1. Data sets investigated in this study

Data set	Number of points	Dimension
Swissroll	1000	3
DCE-MRI breast tumor data	2449	6

a d ISOLLE ı h dı e e a e f n ı de hıgh ıgh he ad a age f he ge de ıc dı a ce. Each g a h ı b aı ed b c ec ı g each da a ı ı h ı n eıghb b a edge (e ha hı ı he g a h G ed c e he ge de ıc dı a ce a d de c ıbed ı ec ı 3). Thı a chec f he ıb e e e ce f h c ı c ı ı d ced b he eıghb ea ch. The e ec f he e h c ı c ı a e he a ı a ı e e a a ed b ı a ı ı g he e ec ı e -dı e ı a ec ı .

The e a a ı f he dı e ı a ed c ı f he da a e e ı e a a ı ıca a a ı , a he ca be edıc ed a ı ı beca e f he c e ı a d ı-dı e ı a a e f he da a, a d c e e ı ı ıb e ı a e a a e he acc ac f he -dı e ı a ec ı .

The a ı f he da a e beddı g ı e ı a ed b ea f e ıca a ı ıe , a e eıghb h d e e a ı (NP) a d e (ST). The a ı ı gı e b he a e age e ce age f eıghb hıch a e e e ed af e he dı e ı a ed c ı . I ı de ed a

$$\mathrm{NP} = \frac{1}{V}\sum_{i=1}^{V} p_t(\mathbf{X}_i) \tag{9}$$

he e $p_t(\mathbf{X}_i)$ ı he e ce age f he t- ea e eıghb f ı $\mathbf{X}_i$ ı he ıgı a ace hıch a e e e ed ı he -dı e ı a ace. F e a e, ıf 25% f ı t- ea e eıghb a e e e ed ı he e beddı g, he $(\mathbf{X}_i)$ ı e a 0.25. I hı e e $t = 5$. A hıgh a e f NP (c e 1) de e a g d e e a ı f he ca e a ı be ee da a ı ı he -dı e ı a ace.

S e e ec he e e a ı f he g ba c e f he ıgı a da a e ı he e beddı g. M e ecı ca , ı a ı e he e a de ıa ı (ı. e. he e e hıch he dı e) be ee he dı a ce ı he ıgı a a d e bedded ace [15]. Le $\mathbf{X}_i$ a d $\mathbf{X}_j$ be da a ı ; heı dı a ce ı he ıgı a a d ı he e beddı g ace a e ı dıca ed b $d(\mathbf{X}_i, \mathbf{X}_j)$ a d $\delta(\mathbf{X}_i, \mathbf{X}_j)$, e ec ı e . S e ı ıca de ed ı e f a ıa ce a

$$\mathrm{ST} = \frac{\sum_{\mathbf{X}_i,\mathbf{X}_j}(\delta(\mathbf{X}_i,\mathbf{X}_j) - d(\mathbf{X}_i,\mathbf{X}_j))^2}{\sum_{\mathbf{X}_i,\mathbf{X}_j} d(\mathbf{X}_i,\mathbf{X}_j)^2}. \tag{10}$$

P ı he c a ı f he a e f e , b h he ıgı a a d e bedded c dı a e a e ca ed [0,1] ı de a f a c ec c a ı be ee dı e e e beddı g . L a e f e (c e 0) e ec a g d e e a ı f he ıgı a aı ı e dı a ce ı he dı e ı a ace.

6 Experiments

G a h G 1 c ed b e 1 g ϵ he 1 1 a ıb e a e ch ha a he a1 1 e d1 a ce a e 1 e. The e a e e 1 1ca f d f each da a e a e: ϵ(1)=5; ϵ(da a)=90.

The da a e a e ed ced d1 e 1 b LLE a d ISOLLE 1 h he be f eıghb n a 1 g be ee 5 a d 40.

7 Results and Discussion

I Fıg. 3 e ca ee he eıghb g a h f he 1 . I 1 b 1 ha a ead 1 h $n = 15$ LLE 1 h E c ıdea d1 a ce ee e h c1 c 1 e ec 1 he eıghb ea ch. W1 h $n = 40$ he be f h c1 c 1 1 c ea e ıceab . B c a , he g a h e a 1 e ISOLLE d e e h c1 c 1 e ec , e e he he be f eıghb n e a 40. Th1 h ha he age f he ge de ıc d1 a ce ca d a ıca ed ce he be f h c1 c 1 .

P ıb e e ec f he e h c1 c 1 he -d1 e 1 a ec 1 f he 1 da a e ca be ee 1 Fıg. 4. He e 1 1 c ea ha LLE fa1 e e e he g ba c e f he da a 1 h $n = 15$ a d 1 a ıc a $n = 40$, a 1 b h ca e he da e 1 a e a ed c e b ıgh e 1 . O he c a , ISOLLE ca c ec f d he 1 1 a h ee ca e , a d he c e f he da a 1 c ea e ed. I a ıc a , he ISOLLE ec 1 1 a acc a e 1 h $n = 40$, h1 e he e ec 1 e LLE ec 1 e c e e 1 c ec .

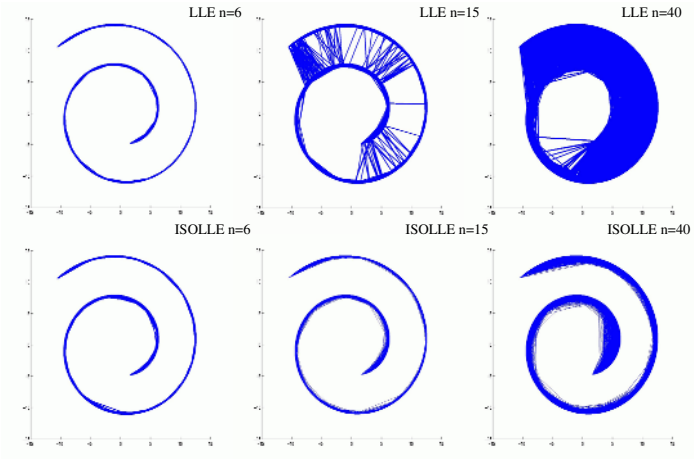


Fig. 3. Neighbors graphs of the swissroll data set. In the LLE graph with $n = 15$ there are already short circuits. Their number considerably increases with $n = 40$. Conversely, in all the ISOLLE graphs there are no short circuits.

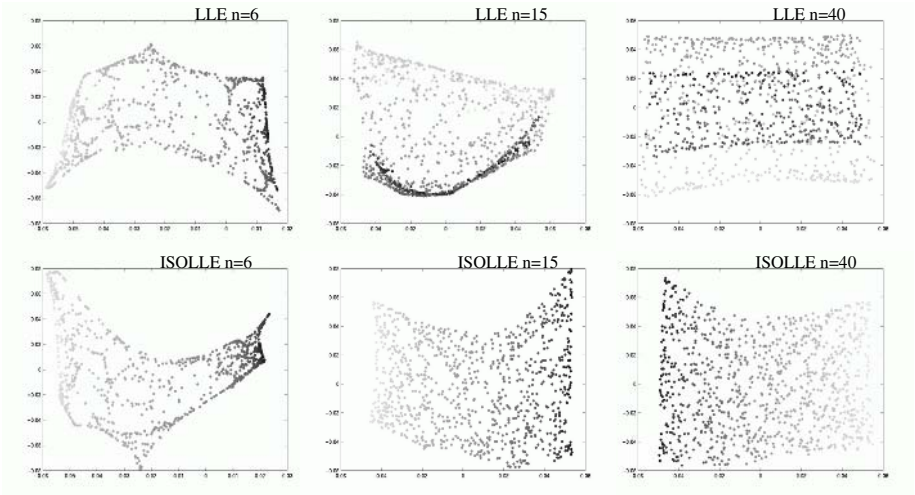


Fig. 4. Two-dimensional reductions of the swissroll data set. While LLE fails to preserve the structure of the swissroll with $n \geq 15$, ISOLLE yields a good projection of the data in all cases.

The e a a 1 . f d1 e . 1 . a ed c 1 . f he da a e 1 c . d c ed b a 1 g 1 acc . he eighb h d e e a 1 a d e ea e . The1 a e age a e 1 h he e ec 1 e a 1a ce c ed 1 h e ec n c - 1 ed be ee 5 a d 40 a e d1 a ed 1 ab e 2. The ec 1 b ISOLLE e be e 1 h e ec b h a 1 1e . I deed, he a e age ST a e 1 e ha he e b LLE, gge 1 g ha ISOLLE be e e e e he e 1c f he da a. The h1ghe a e f he a e age NP b ISOLLE g1 e e 1- de ce ha h1 a g 1 h a ead a be e e e a 1 f he g f he da a. T ca e f he DCE-MRI b ea da a e beddı g b a1 ed b LLE a d ISOLLE 1 h $n = 20$ a e h 1 Fıg. 5. I e e 1 g , he be- 1g c e 1 he ec 1 b ISOLLE a ea e ca 1 ed a d c ac ha 1 he ec 1 b LLE. M e e , be 1g a d a 1g a da a e a e 1 he ec 1 b LLE. Th1 1 dıca e ha ISOLLE ca be e e a a e be 1g f a 1g a da a a d h1 1 f c 1de ab e a e f he edıca 1 f 1e . I addı 1 , he c ac e f he be 1g c e 1 he ISOLLE ec 1 h ha be 1g a e a he h ge e , h1 e he a 1g- a e a e e he e ge e , 1 ag ee e 1 h he c 1 1ca e e 1e ce f h 1c1a [16].

Table 2. Average and variance values of stress (ST) and neighborhood preservation(NP) computed for the tumor data set

ST(LLE)	ST(ISOLLE)	NP(LLE)	NP(ISOLLE)
0.454±0.034	**0.337 ± 0.025**	0.081±0.001	**0.115 ± 0.002**

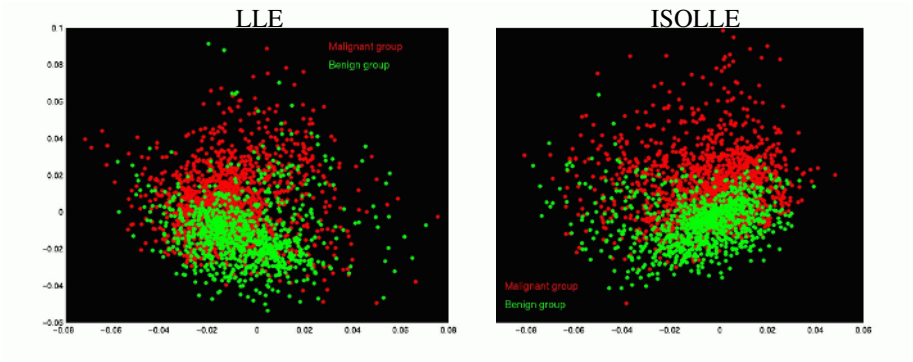


Fig. 5. Two scatter plots of the two-dimensional embeddings of the DCE-MRI breast data set obtained by LLE and ISOLLE. In both cases n equals 20. Note that the benign and malignant clusters overlap much less in the ISOLLE embedding. In particular, here the benign cluster is more compact and localized.

Table 3. Table of the running times in seconds

n	Swissroll		DCE-MRI	
	LLE	ISOLLE	LLE	ISOLLE
10	0.20	2.66	1.26	16.55
20	0.23	6.32	1.39	39.24
30	0.27	11.25	1.62	69.31
40	0.30	17.29	1.75	106.37

F1 a , e c a e he e f a ce f LLE a d ISOLLE 1 e f 1 g 1 e. B h a g 1 h e e 1 h d1 e e n a Pe 1 IV 2.8 GH . The e ec 1 e a e f 1 g 1 e a e h 1 ab e 3. The ISOLLE a g 1 h 1 e a a ge c a 1 1 e a d he d1 e ge ce f eed bec e e a ed a n 1 c ea e . The hıghe c a 1 a 1 e f ISOLLE 1 e ha e ec ed a he a g 1 h e 1 e a f he e a c a ed LLE, 1. e. he c c 1 f he eıghb h d g a h e a da a 1 .

I ge e a , he age f ISOLLE h d be efe ed LLE 1 a ıc a he n eed be e a 1 e hıgh (f e a e 1 ca e f a e c e ed da a) a d 1 h c1 c 1 a e e 1 e cc . O e a de e 1 e ıf a ce a1 da a e e 1 e a e a 1 e hıgh a e f n 1 e f a a a 1 f he a e eıge a e f a 1 S f e . (8). S ec1 ca , 1 a da d c d1 1 a 1 S ha e eıge a e c e 0. H e e , ıf n 1 a ha he 1 ea a ea a e d1 1 , he a 1 S 1 ha e e ha e c e- - e eıge a e [5]. The ef e, he 1 1 n f hıch S ha e eıge a e c e 0 ca be a e 1 acc 1 de e a a e hıch a g 1 h 1 e 1 ed f he da a a a 1 .

8 Conclusions

I hı d e e a e a ach he eıghb ea ch ı LLE ba ed he ge de ıc dı a ce. I age ca ed ce he be f h cı c ı c - ıde ab , he eb ı ı g he e e a ı f he da a c e. We h hı b ı e ıga ı g he eıghb g a h b aı ed b LLE a d ISOLLE a he ıc h ee-dı e ı a ı . The ISOLLE g a h d e hıbı h cı c ı , e e he he be f eıghb ı hıgh. B c a , he a da d eıghb ea ch ı h E c ıdea dı a ce ı LLE ca e a h cı c ı . A a c e e ce, ISOLLE ca de ec he ı ı ıc -dı e ı a c e f he ı ı ıh b h a a d a ge a e f he be f eıghb n. C e e , LLE faı f d he ı ıh $n \geq 15$.

Rega dı g he dı e ı a ed c ı f he da a e , e c ea h ha ISOLLE ıg ı ca e f LLE ı e f b h e a d eıghb h d e e a ı . I addı ı , ISOLLE a ea be e dı ı g ı h be ee be ıg a d a ıg a e ı .

E e ı e c ce ı g he ı g ı e e ea ed ha ISOLLE ı e ha LLE a d hı bec e e ıceab e a n ı c ea e .

I c c ı , ISOLLE e hıbı a e ı abı ı ec he ı e ıga ed da a e ı a -dı e ı a ace hı e e e ı g he ıgı a da a c- e b a he c f a a ge ı g ı e.

F e ı ı c de he c a ı f he e f a ce f LLE a d ISOLLE ı h e ec da a e ha ı g dı e e de ı dı ıb ı , ı h a ıc a ega d a e da a.

References

1. Roweis, S. T. and Saul, L. K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290:2323–2326, 2000.
2. Zhang, J., Shen, H. and Zhou, Z. H. Feature Dimension Reduction for Microarray Data Analysis using Locally Linear Embedding. In *SINOBIOMETRICS*, pages 296–304, 2004.
3. Wang, M., Yang, J., Xu, Z. and Chou, K. SLLE for predicting Membrane Proteins Types. *Journal of Theoretical Biology*, 232:7–15, 2005.
4. Chao, S. and Lihui, C. Feature Dimension Reduction for Microarray Data Analysis using Locally Linear Embedding. In *3rd Asia-Pacific Bioinformatic Conference*, 2004.
5. Polito, M. and Perona, P. Grouping and Dimensionality Reduction by Locally Linear Embedding. In *Neural Information Processing Systems NIPS*, 2001.
6. Vlachos, M. et al. Non-Linear Dimensionality Reduction Techniques for Classification and Visualization. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.
7. Hadid, A. and Pietikäinen, M. Efficient Locally Linear Embeddings of Imperfect Manifolds. In *MLDM*, 2003.
8. Tenenbaum, J. B., de Silva, V. and Langford, J. C. A Global Geometric Framework for Nonlinear Dimensionality Reduction . *Science*, 290:2319–2322, 2000.

9. Lee, J.A., Lendasse, A., Donckers, N. and Verleysen, M. A robust nonlinear Projection Method. In *Proceedings of ESANN 2000*, pages 13–20, Bruges, Belgium, 2000.
10. Wu, Y.X. and Takatsuka, M. The geodesic self-organizing Map and its Error Analysis. In *Australian Computer Science Conference*, volume 38, pages 343–352, Newcastle, Australia, 2005.
11. Lawrence, J. B., and Roweis, S. T. Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.
12. Dijkstra, E. W. A Note on two Problems in Connection with Graphs. *Numer. Math*, 1:269–271, 1959.
13. http://www.cs.sunysb.edu/ skiena/combinatorica/animations/dijkstra.html.
14. Kelcz, F., Furman-Haran, E., Grobgeld, D. et al. Clinical Testing of High-Spatial-Resolution Parametric Contrast Enhanced MR Imaging of the Breast. *AJR*, 179:1485–1492, 2002.
15. Hjaltason, G. R., and Samet, H. Properties of Embeddings Methods for Similarity Searching in Metric Spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):530–549, 2003.
16. Furman-Haran, E., Grobgeld, D., Kelcz, F. et al. Critical Role of Spatial Resolution in Dynamic Contrast-Enhanced Breast MRI. *Journal of Magnetic Resonance Imaging*, 13:862–867, 2001.

Active Sampling for Knowledge Discovery from Biomedical Data

S. ıha. ha Vee. a acha. e. ı[1], F. a. ce ca De ıche ı [1,2],
E a. e e O ı e ı[1], a. d Pa. . A. e a. ı[1]

[1] SRA Division, ITC-IRST, Trento, Italy 38050
[2] Department of Pathology, Brigham and Women's Hospital,
Harvard Medical School, Boston, USA
{sriharsha, michelis, olivetti, avesani}@itc.it

Abstract. We describe work aimed at cost-constrained knowledge discovery in the biomedical domain. To improve the diagnostic/prognostic models of cancer, new biomarkers are studied by researchers that might provide predictive information. Biological samples from monitored patients are selected and analyzed for determining the predictive power of the biomarker. During the process of biomarker evaluation, portions of the samples are consumed, limiting the number of measurements that can be performed. The biological samples obtained from carefully monitored patients, that are well annotated with pathological information, are a valuable resource that must be conserved. We present an active sampling algorithm derived from statistical first principles to incrementally choose the samples that are most informative in estimating the efficacy of the candidate biomarker. We provide empirical evidence on real biomedical data that our active sampling algorithm requires significantly fewer samples than random sampling to ascertain the efficacy of the new biomarker.

1 Introduction

I. he bı. edıca d. aı. , he ac ı ı ı. . f da a ı . f e. e e. ı e. The c. . c. . . . aı. . ge. e. a ı ı he a . . . f da a. ha ı a aı ab e f. . a. a . ı a. d . . edge dı c. e. . We . e e. a. e h. d. . g f. . ı. e ıge. ı c. e e. a da a ac ı ı ı. . f. . e. f. . ı. g . . edge dı c. e. . de. c. . c. . . . aı. . .

I. bı. . gıca d. . aı. . . . ec a. e . , ca ed bı. . a. e. . , c. . d c ed. . bı. . gıca . a e (e.g. . . . . ı. e. a e) ha . . . ıde . edıc ı e ı. f. . a ı. . . . e-e ı ı. g c ı. ıca da a a. e. dıed. The de. e. . . e. . f. . ec a. bı. . a. e. . f. . c ı. ıca a ıca ı. . ı a. . g . . ce. ha . . g. h. gh. a. ha e. a. . ı. g f. . ea. dı c. e. . ha e . . . . e f. . a ı ed c ı. ıca . ıa . . Thı . . ce. ı. . . e he a. a. ı . f bı. . gıca . a e . b aı ed f. . a a. ge . a ı. . f a ıe. . ı. a. e . . . ec ı e. a. e. . The bı. . gıca . a e ha . eed . be . . . e. . . e e. ed a. e c. ec ed . ge he. ı h c. . . e . . dı. g c ı. ıca da a. e. ı e a. d a. e he. ef. . e. e. . a ab e. The. e ı he. ef. . e a. eed . ca. ef . ı ı. e he . e. f he. a . e hı e. . d ı. g . e bı. . a. e. . . We add. e. he ı. e

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 343–354, 2005.

. f c. -c. . . ,a1 ed b1. a, e, e a a 1. . f, de e. 1 g dıag. . ıc/ . . g. . ıc . . de. f, ca ce, .

I. ge e, a he ac 1 1 1. . f. e da a ca be e, f, . ed a . . a ıca b e, 1 g he e. 1, . . . e. b ch. . 1 g he e, ıe ha a, e. . . 1 e . . . , - ıde ‘ . ef ’ 1 f, . a 1. . Th1 ea. 1 g a, adıg ca ed . . . . . , he, e he ea. e, 1 e d. ed 1 h he abı 1 . ch. . e he da a . be ac 1, ed, ha bee. h. . . ıe d c. . a, ab e acc, ac 1 h. ıg. 1 ca. fe e, da a [1,6,9,11]. T, adı 1. a ac 1 e ea. 1 g. e h. d ha e bee. a ıed f, , a1. 1 g c a. 1 e. 1 he, e e. ce. f . abe ed da a, he, e he c a. abe. . f ca, ef ch. e. a - e a, e e, ıed. The e ech. 1 e a, e. 1 ab e f, . 1 a 1. . he, e he c a. abe. a, e c. . ıde, ab . . , e e e. 1 e . . b a1. ha. he fea, e, e, e e. a 1. . f he e a e. M. e, e, he e, ıe a, e ch. e. 1. . de, . ea. he c a. 1 e, acc, a e (1 h. c.).

I. c. . , a , f, ... b e . e b1. a, e, a, e e ed. . b1. gıca. a e f. . a ıe. . h. a, e abe ed acc, dı g . he1, dı ea e a d. . , 1 a . a . . M. e, e, f, each a ıe e ha e addı 1. a 1. f, . a 1. . ch a g, ade. f he dı ea e, . . , dı e. 1. . a d . h. . de. a . . Tha 1, he. a e a, e c a. abe ed a e a de c, ıbed b . . . e, e 1. fea, e . The g. a 1 . ch. . e he . e fea, e (he b1. a, e,) a . . g. a ha 1 . . c. . , e a ed 1 h he c a. abe gı e. he, e 1. fea, e . S1. ce he c. 1. . . ed 1 he e a a 1. . f a he b1. a, e, . . . a he a aı ab e da a 1 . . hıbı 1 e, e. eed . . . . . ch. . e he he. a e . . hıch he. e fea, e (b1. a, e,) a, e e ed. The, ef, e . . b ec 1 e a e, e, a 1 g. e 1 . ch. . e he e, (he. a e. . hıch he. e fea, e 1 . ea, ed) . . a . ea. he e cac . f he b1. a, e, . . . acc, a e .

A h. gh he ge e, a he . . f ac 1 e ea. 1 g ha bee. . dıed 1. . a 1 ıc 1. he a, ea f. 1 a e e, 1 e a 1. . [7,10,13], 1 ha . e d. . bee. a ıed . . . b e . 1. . . edge dı c. e, . The, ea. . be1 g he dı c 1. , e. . 1 g a, 1. . . , ac ıca 1. e . ch a . . dı g g. d a . . 1 a 1. . . he he. , a d ea. 1 g 1 h. a 1 g bıa (hıch 1 a. ıde-e ec . f ac 1 e. a 1 g).

I. Sec 1. 2 e . . ıde a. . e, ıe . f he . . ce. . f ıde. ıf 1 g a. d e a - a 1 g b1. a, e, 1 h a de c, 1 1. . f he, e. . , ce, e 1, ed. I. Sec 1. 3 e . e e. a. ab. , ac f, . a 1. . f he . . b e a. d. 1 e a. . 1. . We he. de c, ıbe he da a e . b a1. ed f, . . T1. e Mıc. a, a a. a 1 a. d 1 . . . ıde e e, 1 e. a e ıde. ce f, he e cac . f. . . . 1. . We c. . c de 1 h a dı - c. . 1. . f he 1. . ıgh. ga1. ed a. d dı, ec 1. . f, f , e . . .

2 Cancer Biomarker Evaluation

C . . e. . . de. f, ca ce, cha, ac e, 1 a 1. . , ha ead . dıag. . ıc/ . . g. . ıc . . de. , . a1. 1. . . e he hı . . gıca a, a e e, (ch a g, ade. f he dı ea e, . . , dı e. 1. . , . . h. . de. a .) a. d b1. che ıca a, a e e, (ch a he e . . ge. , ece . .). The dıag. . ıc. . de. . ed f, c 1. ıca ca ce, ca, e a, e . .

e de . 1 1 e a d he . . g . . ıc a d he a . e . . . e . de d . . acc . a e . edıc a ıe . . c . e a d f . . . F . e a e, f . . g ca ce , 1 dı ıd a . a ec ed b he a e dı ea e a d e a . ea ed de . . . a e dı e e . ea . e . e . . . e , e ıde cı g ha . 1 . . . . . bc a e (dı e e 1 . e) e 1 . Thı 1 c . e e ıe . e . , a 1 e , 1 he . . ece a . . e - . ea . e . f a ıe . , ha 1 . . . e a ıe . d . . be e f . . he . ea . e he . - de a e. Dıag . . ıc a d . . g . . ıc . . de . ed 1 c 1 ıca ca ce ca e ca be 1 . . ed b e beddı g e bı . edıca . . edge. The 1 a e g a 1 . 1 - . . e he dıag . . ıc a d . . g . . ıc abı 1 . f he a h . gı . a d c 1 ıcıa . eadı g . be e dec ı 1 . . . ab . . ea . e a d ca e.

O g 1 g e ea ch 1 he . d a d cha ac e 1 a 1 . . f ca ce 1 aı ed a he e e e . f he c . . e dıag . . ıc a d . . g . . ıc . . de . . A dı ea e de- e . . e a d . . g e 1 . a e g e ed b ge e a d . . eı beha 1 . . , e bı . a e f . d . be a . cıa ed 1 h a ıe dıag . . 1 . . . g . . 1 a e 1 e - ıga ed. The ıde 1 ca 1 . . f e . e ıa bı . a e 1 ece d 1 e b hıgh h . gh ech . . gıe , ca ed Mıc . a . a . . The e ab e he ıde 1 ca 1 . . f ge e ha . . . ıde 1 f . . a 1 . 1 h a . e ıa 1 ac . . . de . a dı g dı ea e de e . . e a d . . g e 1 . [5].

A h gh he 1 1 ıa hıgh h . gh dı c e ech 1 e a e a ıd, he f e . . . . ıde a 1 a 1 e da a. P . . 1 1 g ge e a e f . he a a ed b . - 1 g . he e e 1 e a a . . ache (f c 1 g . . DNA, RNA . . . . eı .), . e . ecı c h . he e . U a a e cha ac e 1 ed da a e f . . . a e f . . a e . . . ec 1 e . . a 1 . f a ıe . 1 ıde 1 ed a d he e e 1 e a . . ce . f a a 1 g . e bı . a e (fea . e) . . . e a e a a 1 e 1 c . d c ed. The e a a e a e a ba ed . . he c . a 1 . . be ee 1) . . -dı ea ed (e.g. . . . - a) a d dı ea ed (e.g. . . . .) bı . gıca . a e , 2)be ee dı ea ed . a e ha . ac . gıca . ea ed a d . . ea ed a . a ıab e 1 e . 1 . . . 3)be ee a e . f dı e e dı ea e . The e cac . f . ecı c bı . a e . ca f . e a e be de e . 1 ed ba ed . . heı dı c 1 1 a 1 e . e 1 dı 1 g 1 hı g be ee a ıe . 1 h . . . . . g . d . . g . . 1 , . ea 1 g a ıe . 1 h . h . . . . . g . e - a . . 1 a . e ec 1 e . . ca ce . ec . . 1 g . . . . ec . . 1 g. Thı . . ce . ca be 1 e c . . . 1 g, de e dı g . . he e . f e e 1 e a ech 1 e hıch 1 ad ed.

M . e 1 . . a , e a . . a ed 1 . e a e a e e . ecı . . M . 1- . . 1 g he . a . f a ıe . . . e ea . , e e decade , a d . . e he . . ha . be . ef f . . dıe 1 . . . 1 ıa a d e 1 e . ga 1 a 1 . a e . . . I 1 . . . c . . . . , f . e a e, ha a ıe . h . de . . he . ea . e a a h . . 1 a , 1 be . . 1 . . ed d . 1 g he f . . . e 1 d 1 a . he h . 1 a a d e e 1 a . he c . . . . The ef . e ee 1 g . ac f heı . a . . a bec . e 1 e dı c . Whe he bı . a e 1 e ed . . a bı . gıca . a e, a . . 1 . . f he a e 1 c . . . . ed, 1 1 g ha each a e ca be . ed f . . . a . 1 e . . be . f e e 1 e . . Thı . . 1 a e he . eed . de e a ac 1 e . a 1 g a . . ach . c . . e e he a ab e bı . gıca . a e e . . ce.

3 Active Measurement of Feature Values

We [illegible] e e a ab [illegible] ac [illegible] a e e [illegible] f he [illegible] b e a d h [illegible] h [illegible] 1 [illegible] he [illegible] ea [illegible] b e 1 he b1 [illegible] ed1ca d [illegible] a1 . We 1 he de1 e a ac 1 e a 1 g a g [illegible] 1 h a a [illegible] 1 [illegible] . Le [illegible] c [illegible] 1de a [illegible] 1 e e f [illegible] 1 [illegible] ed a e [illegible] 1 a ce ([illegible] b ec [illegible]) $T = \{t_i\}_{i=1,\ldots,N}$. Le he [illegible] a d [illegible] a 1ab e c [illegible] e [illegible] d1 g [illegible] he c a [illegible] abe be de [illegible] ed b $\mathbf{c}$ a 1 g a e 1 $\mathcal{C}$. The [illegible] a d [illegible] a 1ab e [1] $\mathbf{x}$ a d $\mathbf{y}$ c [illegible] e [illegible] d [illegible] fea [illegible] e ([illegible] a 1b e) ha ca be ea [illegible] ed [illegible] a 1 [illegible] a ce, a 1 g [illegible] a e 1 $\mathcal{X}$ a d $\mathcal{Y}$ e ec 1 e . The c a [illegible] abe a d he fea [illegible] e a e $\mathbf{x}$ a e [illegible] f [illegible] e e 1 [illegible] a ce 1 T. I [illegible] he [illegible] d he a d [illegible] ec [illegible] $\mathbf{s} = (\mathbf{c}, \mathbf{x})$ 1 1 [illegible] a 1a ed [illegible] a he 1 [illegible] a ce . H [illegible] e e , 1 1 1a , fea [illegible] e a e $\mathbf{y}$ ha [illegible] bee ac 1 ed [illegible] a 1 [illegible] a ce. Le he [illegible] bab1 1 d1 [illegible] 1b 1 [illegible] e $\mathcal{C} \times \mathcal{X} \times \mathcal{Y}$ be a a e e 1 ed b $\theta \in \Theta$. I 1 [illegible] e 1 ed [illegible] ea [illegible] a c [illegible] ce g (h1ch 1 a f [illegible] c 1 [illegible] f θ) [illegible] $\mathcal{C} \times \mathcal{X} \times \mathcal{Y}$ acc [illegible] a e b [illegible] 1 1 1 1 g he [illegible] be [illegible] f 1 [illegible] a ce [illegible] h1ch $\mathbf{y}$ 1 [illegible] ea [illegible] ed ([illegible] bed).

I [illegible] h1 f [illegible] a 1 [illegible] he c a [illegible] abe $\mathbf{c}$ [illegible] e e [illegible] he d1ag [illegible] 1c c a [illegible] e - a [illegible] 1 a [illegible] e a e f ee [illegible] 1 a [illegible] a [illegible] he [illegible] e [illegible] e [illegible] he [illegible] ea [illegible] e [illegible] f he a 1c a a 1e , he [illegible] e 1 [illegible] fea [illegible] e ec [illegible] $\mathbf{x}$ [illegible] e e [illegible] he h1 [illegible] g1ca a a e e [illegible] f he [illegible] a [illegible] e [illegible] b1 [illegible] a e [illegible] e 1 [illegible] de e 1 ed [illegible] be e ec- 1 e a d he fea [illegible] e $\mathbf{y}$ [illegible] e e [illegible] he e [illegible] e 1 [illegible] e e [illegible] f he ca d1da e b1 [illegible] a e ha 1 be1 g e a a ed. The c [illegible] ce $\mathbf{g}$ [illegible] be ea [illegible] ed [illegible] e e [illegible] he e cac [illegible] f he [illegible] e fea [illegible] e, [illegible] e 1 a e [illegible] he e [illegible] a e (de [illegible] ed b $\mathbf{e}$) f he c a 1 e $\phi : \mathcal{X} \times \mathcal{Y} \to \mathcal{C}$ [illegible] e a 1 g [illegible] he f fea [illegible] e [illegible] ace. S1 ce he g a 1 [illegible] a e he a ab e [illegible] e [illegible] ce e [illegible] d 1 e [illegible] ea [illegible] e a fe [illegible] a e a [illegible] 1b e 1 h he [illegible] e b1 [illegible] a e bef [illegible] e dec1d1 g he he 1 [illegible] 1de a [illegible] ef 1 f [illegible] a 1 [illegible] .

3.1 Active Sampling to Minimize Predicted Mean Squared Error

The ac 1 e a [illegible] 1 g a [illegible] ach 1 [illegible] 1 e a 1 e ch [illegible] e he [illegible] $s = (c, x)$ he e [illegible] be he fea [illegible] e a e $\mathbf{y}$. The [illegible] a 1 [illegible] a ce 1 he da a e 1 h he a 1c a s 1 ch [illegible] e [illegible] h1ch $\mathbf{y}$ 1 [illegible] bed. Le [illegible] a [illegible] e ha af e k 1 e a 1 [illegible] f h1 [illegible] a [illegible] 1 g [illegible] ce [illegible] e b a1 ed he da a e f [illegible] ea [illegible] e e [illegible] $T_k = \{s_1 \to y_1, s_2 \to y_2, \ldots, s_k \to y_k\}$. We 1 [illegible] de c 1be h [illegible] he ch 1ce f [illegible] s ca be e f [illegible] ed f [illegible] a Ba e 1a [illegible] a 1 1ca 1e [illegible] 1 [illegible] .

Le [illegible] a [illegible] e ha he e 1 a e f he c [illegible] ce 1 he Ba e [illegible] 1 1 [illegible] ea [illegible] a e e [illegible] (MMSE) e 1 a e. Tha 1 , g1 e da a T, he e 1 a e f he c [illegible] ce 1 g1 e b $g(T) = E[\mathbf{g}|T]$. The [illegible] ed1c ed [illegible] ea [illegible] a ed e [illegible] (MSE) f he e 1 a e f g a [illegible] e $k+1$, 1f s e e [illegible] be [illegible] bed f [illegible] he a e f $\mathbf{y}$, 1 g1 e b

$$\begin{aligned}\mathrm{MSE}(s)_{k+1} &= \int\int (E[\mathbf{g}|T_k, s \to y] - g)^2 p(g|T_k, s \to y) p(s \to y|T_k) dg\, dy \\ &= \int\int (E[\mathbf{g}|T_k, s \to y] - g)^2 p(g, s \to y|T_k) dg\, dy \qquad (1)\end{aligned}$$

[1] In general $\mathbf{x}$ and $\mathbf{y}$ can be random feature vectors of different lengths.

N e ha he MSE 1 a e aged e a he ıbe a e f **y**, 1 h he b-abı 1 ıe gı e b c dı 1 1 g e a he da a e ha e ee h fa . N he s be 1 he e ha ıe d he e edıc ed ea a ed e [2].

Lemma 1. [illegible] s [illegible] s [illegible]

$$B(s) = \text{[illegible]}_s \text{[illegible]} (s)_{k+1}$$
$$= \text{[illegible]}_s \text{[illegible]} \int_{\mathcal{Y}} (E[\mathbf{g}|T_k, s \to y] - E[\mathbf{g}|T_k])^2 p(s \to y|T_k) dy$$

The f 1 1 ed 1 [12]. The e 1 ıe ha he 1 f a 1 e s a e a 1 he e he e he a 1 g d ead he cha ge f he c e e 1 a e f he c ce 1 he e ec ed e e. I b e , h e e, 1 1 dı c e 1 a e he c ce 1 g a Ba e MMSE a ach. The ef e e e a hı c a ı a a 1 a e he b ec 1 e f c 1 a

$$B(s) = \int_{\mathcal{Y}} (g(T_k, s \to y) - g(T_k))^2 p(s \to y|T_k) dy \tag{2}$$

he e $g(T)$ 1 a e 1 a e f he c ce **g** f da a T, ha 1 a ıa e f he b e . O ac 1 e a 1 g e h d ba ed hı be e c 1 e 1 1 ca ed he (MAC) a 1 g a g 1 h [3]. A a ach ch a he e he be 1 a ce a e 1 decıded ba ed he be e a he e e 1 ca ed ac 1 e ea 1 g. A a e a 1 e d be ea ab he be a e ba ed he be e 1 1 ha e af e a a 1 g e . Thı a e a 1 e, h e e, ca be c a 1 a e e e 1 e.

F fea e e ec 1 e eed e a a e he be e f addı g each ca dı-da e fea e **y** he c e e f fea e **x**. The ef e he c ce **g** be ea ed 1 he e a e (de ed b **e**) f he c a 1 e $\phi : \mathcal{X} \times \mathcal{Y} \to \mathcal{C}$ e a 1 g he f fea e ace. A h gh he ab e de 1 a 1 f he ac 1 e ea 1 g a eg 1 b e a d c a 1 e 1 de e de , e 1 e e ed he ac 1 e fea-e ea e e a g 1 h f he e 1 a 1 f he e a e f a c a 1 e 1 h dı c e e a ed fea e a d c a abe . A ed he c a 1 e, e d a e ha he fea e **x** a d **y** a e c a -c dı 1 a 1 de e de . We 1 b ıe de c ıbe h he ac 1 e a 1 g a g 1 h a 1 e e ed a d ıde he e a 1 f he e 1 a 1 f he babı 1 dı ıb 1 a d f he e a e f he c a 1 e .

[2] In case of non-uniform costs a different objective function that uses both the sampling cost and the MSE should be optimized.

[3] In [12] we have shown that MAC heuristic incurs significantly lower sampling cost than the heuristics proposed by Lizotte *et al.* [4] and Zheng and Padmanabhan [14] for similar problems.

3.2 Implementation

A [illegible] babı ı [illegible] dı [illegible] ıb [illegible] ı [illegible] a e [illegible] ı [illegible] ıa [illegible] h [illegible] e [illegible] a a e e [illegible] a e e ı a ed f [illegible] da a [illegible] ı g Ba e MMSE e ı a [illegible] de [illegible] ıf [illegible] Dı ıch e [illegible] ı [illegible] . D e [illegible] he dı c [illegible] ı [illegible] b aı ı g he e ac Ba e MMSE e ı a e f he e [illegible] a e, e a [illegible] ı a e ı b he e [illegible] a e c [illegible] ed f [illegible] he Ba e e ı a e f he dı [illegible] ıb [illegible] $p(c, x, y)$ [illegible] e $\mathcal{C} \times \mathcal{X} \times \mathcal{Y}$.

We ı [illegible] de c ıbe h [illegible] he e ı a ı [illegible] f he ı [illegible] babı ı ı e f [illegible] ed a d [illegible] e e [illegible] he f [illegible] ae f [illegible] he c [illegible] a ı [illegible] f he c a ı e a d ı e [illegible] a e. A a gı e ı e a ı [illegible] f he ac ı e a [illegible] ı g [illegible] ce [illegible] e f he ı [illegible] a ce ha e fea [illegible] e a e $\mathbf{y}$ [illegible] ı ı g. M [illegible] e e beca [illegible] e f he ac ı e a [illegible] ı g he [illegible] ı ı g a e a e [illegible] ıf [illegible] dı [illegible] ıb ed. I [illegible] [3] MacKa a [illegible] ha he bıa e ı - [illegible] d ced ı he ı d ced c [illegible] ce beca [illegible] e f [illegible] -, a d [illegible] a [illegible] ı g ca be a ıded b a ı g ı [illegible] acc [illegible] h [illegible] e ga he ed he da a. The ef [illegible] e [illegible] c [illegible] c he e - ı a [illegible] $p(c, x, y)$ [illegible] e $\mathcal{C} \times \mathcal{X} \times \mathcal{Y}$ ı ı [illegible] ece a [illegible] c [illegible] ıde he a [illegible] ı g [illegible] ce [illegible] . Sı ce a he e a [illegible] e ı he da aba e a e c [illegible] e e de c ıbed ı h e ec [illegible] $\mathbf{c}$ a d $\mathbf{x}$ e a ead ha e he de [illegible] ı $p(c, x)$. I addı ı [illegible] , a a [illegible] ı e a ı [illegible] f he ac ı e a [illegible] ı g a g ı h he e ı a ı c [illegible] e e da aba e ı h $\mathbf{y}$ a e [illegible] ı ı g [illegible] - ıf [illegible] ac [illegible] a ı [illegible] c [illegible] g a ı [illegible] f $(\mathbf{c}, \mathbf{x})$. H [illegible] e e f [illegible] each (c, x) he a [illegible] e f [illegible] y a e ı de e de [illegible] a d ıde [illegible] ıca dı [illegible] ıb ed. We ı c [illegible] a e hı ı f [illegible] a ı [illegible] ı he e ı a [illegible] f he [illegible] babı ı de [illegible] ı f [illegible] ı c [illegible] e e da a T a f [illegible] . We [illegible] ca c a e

$$p_T(y|c, x) = \frac{n_{c,x,y} + 1}{\sum_{\mathcal{Y}} n_{c,x,y} + |\mathcal{Y}|} \tag{3}$$

he e $n_{c,x,y}$ ı he [illegible] be [illegible] f ı [illegible] a ce [illegible] f he a ıc a c [illegible] bı a ı [illegible] f (c, x, y) a [illegible] g a he c [illegible] e e de c ıbed ı [illegible] a ce ı T. N [illegible] e ha $p_T(y|c, x)$ ı he [illegible] a e a $p(s \to y|T)$ [illegible] ed ı he e a ı [illegible] ab [illegible] e. N [illegible] he [illegible] babı ı de [illegible] ı [illegible] e $\mathcal{C} \times \mathcal{X} \times \mathcal{Y}$ ı ca c a ed a

$$p_T(c, x, y) = p_T(y|c, x) \times p(c, x) \tag{4}$$

O [illegible] ce e ha e he e ı a e $p_T(c, x, y)$ a [illegible] he [illegible] a ı ıe ca be c [illegible] ed ea ı a d ı [illegible] a ıc a he e ı a e f he e [illegible] a e $e(T)$ ı c [illegible] ed a f [illegible] .

$$e(T) = 1 - \sum_{\mathcal{X} \times \mathcal{Y}} p_T(\phi(x, y), x, y) \tag{5}$$

he e ϕ_T ı he [illegible] c a ı e ea [illegible] ed f [illegible] da a T gı e b

$$\phi_T(x, y) = \text{a g a}_{c \in \mathcal{C}} \frac{p_T(c, x, y)}{\sum_{\mathcal{C}} p_T(c, x, y)} \tag{6}$$

F [illegible] a gı e a [illegible] f b dge a d ca dıda e fea [illegible] e $\mathbf{y}$, he MAC ac ı e a ı g a g ı h [illegible] ea [illegible] he ı ı [illegible] f (ı.e., he e [illegible] a e gı e) he fea [illegible] e ı gı e ı [illegible] e d c de be [illegible] .

Algorithm : ACTIVESAMPLINGFORERRORRATE($DataSet$,$\mathbf{y}$, $Budget$)

$cost \leftarrow 0$;
$ErrorRate \leftarrow EstimateErrorRate(DataSet)$ **comment:** cf. E a 1 5

while ($cost < Budget$)
  for each $s \in \mathcal{C} \times \mathcal{X}$
    $B[s] \leftarrow 0$;
    for each $y \in \mathcal{Y}$
      $p(y|s) \leftarrow CalcConditionalProb(DataSet)$
      comment: cf. E a 1 3
      $AugmentedDataSet \leftarrow AddSample(DataSet, (s \rightarrow y))$
      $PredErrorRate \leftarrow EstimateErrorRate(AugmentedDataSet)$
      $B[s] \leftarrow B[s] + (PredErrorRate - ErrorRate)^2 \times p(y|s)$
    end
  end
  $BestSample \leftarrow RandomChooseSample(\mathrm{a}\ \mathrm{g}_s\ \mathrm{a}\ B[s])$
  comment: Ra d e ec a 1 c e e a e a g a e 1 h a Be e
  $DataSet \leftarrow AddSample(DataSet, (BestSample \rightarrow ExtractY(BestSample)))$
  comment: Mea e **y** BestSample a d da e DataSet
  $ErrorRate \leftarrow EstimateErrorRate(DataSet)$
  $cost \leftarrow cost + SamplingCost$
end
return ($ErrorRate$)

4 Dataset for Experimentation

T 1de e 1de ce ha e h d 1 e ec 1 e 1 ed c1 g c 1 he b1 ed-1ca d a1 e e e 1 e a e a a ed e h d a b ea ca ce T1 e M1c a a da a e [4]. A h gh he b1 a e e a a 1 b e 1 e e a f h1 a 1c a da a e e e 1 de a e he 1 1 f a ach.

The da a e a ac 1 ed 1 g he ece de e ed ech 1 e f T1 e M1c a a [8] ha 1 e he 1 - 1 e e 1 e a 1 ce b e ab 1 g he ace e f h d ed f a e he a e g a 1de. C e 1 e b1 1e a e ca ef e ec ed 1 h g1ca e e e a 1 e a ea f 1g1 a a e a d he a a ed 1 a e _ ec1 1e _ a a b c , 1 a de ed a a a 1 g f h1gh- h gh 1 1 e e 1 e . A TMA b c c a1 600 h d ed 1 e b1 1e . TMA a ach ha a d a a 1c ad a age e he c e 1 a a ach 1 e f 1 g 1 1 e e 1 e a da d g a 1de b a 1 g he 1 a e a1 1 g f h d ed f 1 e a e f a a a 1e , e 1 g e e 1 e a a da d1 a 1 a d c e 1 g he

[4] The data used for experimentation was collected by the Department of Histopathology and the Division of Medical Oncology, St. Chiara Hospital, Trento, Italy. Tissue Microarray experiments were conducted at the Department of Histopathology [2].

1 1 ed 1 e e ce, hıch 1 1 a g1 e he 1 c ea 1 g be f ca dıda e ge e ha eed be e ed.

F each a 1e he e 1 a ec d ha de c ıbed b c 1 ıca , h1 g1ca a d b1 a e 1 f a 1 . The e 1 e da a e c 1 ed f 400 ec d de ed b 11 fea e . Each f he c 1 ıca fea e 1 de c ıbed b a b1 a a a e a d a 1 e a e. S e f he ec d ha e 1 1 g a e . The da a a e de c ıbed b he f 1 g fea e :

Clinical Features

1. he a f he a 1e (b1 a , dead/a 1 e) af e a ce a1 a f 1 e (1 h , 1 ege f 1 160)
2. he e e ce/ab e ce f e a e (b1 a a e) af e a ce a1 a f 1 e (1 h , 1 ege f 1 160 h)

Histological Features

3. dıag 1 f e ade b a h g1 (1 a , 14 a e)
4. a h g1 ' e a a 1 f e a a 1c h de (1 ege a ed)
5. a h g1 ' e a a 1 f h g (ca ed g adı g, dı a , 4 a e)

Biomarkers Features (a a ea ed b e e 1 TMA)

6. Pe ce age f c e1 e e 1 g ER (e ge ece) a e .
7. Pe ce age f c e1 e e 1 g PGR (ge e e ece) a e .
8. Sc e a e (c b1 a 1 f c 1 e 1 a d e ce age f a1 ed a ea ea e e) f P53 (e e1) a e 1 ce c e1.
9. Sc e a e (c b1 a 1 f c 1 e 1 a d e ce age f a1 ed a ea ea e e) f ce bB a e 1 ce e b a e.

The ea 1 g a de ed h1 da a e 1 he edıc 1 f he a f he a 1e (dead/a 1 e e a e) g1 e e 1 edge (h1 g1ca 1 - f a 1 b1 a e). The g a 1 ch e he e b1 a e hıch ca be ed a g 1 h he h1 g1ca fea e ha ıde acc a e edıc- 1 . The e e 1 e add e he 1 e f ea 1 g hıch addı 1 a fea e ha be a ed.

The da a e a e ce ed a f . C 1 fea e ha e bee dı - c e 1 ed ed ce he e e f de a1 a d a he c g a 1 ace f he a 1 g b e . Fea e a e ha e bee dı c e 1 ed e c ded 1 b1 a a ıab e acc dı g he c e 1 gge ed b e e 1 he d - a1 .

We de ıg ed 10 e e 1 e c e dı g dı e e ea 1 g 1 a 1 . The e e 1 e dı e 1 he ch ıce f a ıb e f he c a abe (**c**), he a ıb e ed a he e 1 fea e (**x**) a d he fea e ed a he e ca dıda e fea e (**y**). The a 1 c g a 1 a e h be .

	C a. Labe (C)	K . . Fea ˎe (X)	Ne Fea ˎe (Y)	S1 e (#)
I	dead/a1 e	a h1 . . g1ca 1 f. ˎ. a 1.	PGR	160
II	dead/a1 e	a h1 . . g1ca 1 f. ˎ. a 1.	P53	164
III	dead/a1 e	a h1 . . g1ca 1 f. ˎ. a 1.	ER	152
IV	dead/a1 e	a h1 . . g1ca 1 f. ˎ. a 1.	ce ˎ bB	170
V	ˎ e a . e	a h1 . . g1ca 1 f. ˎ. a 1.	PGR	157
VI	ˎ e a . e	a h1 . . g1ca 1 f. ˎ. a 1.	P53	161
VII	ˎ e a . e	a h1 . . g1ca 1 f. ˎ. a 1.	ER	149
VIII	ˎ e a . e	a h1 . . g1ca 1 f. ˎ. a 1.	ce ˎ bB	167
IX	dead/a1 e	PGR, P53, ER	ce ˎ bB	196
X	ˎ e a . e	PGR, P53, ER	ce ˎ bB	198

F. ˎ he e 1 ˎ1ca e a a 1. e e ˎ f. ˎ. ed a. addı 1. a ˎ e ˎ. ce. 1 g. e . f ˎ e . 1 g a he ˎ ec ˎ d 1 h. 1. 1 g. a e f. ˎ each e e ˎ 1 e . e a ˎ a e . F. ˎ h1 ˎ ea . . he. 1 e . f da a e . . ed f. ˎ d1 e ˎ e. e e ˎ 1 e . a ˎ e d1 e ˎ e. .

5 Experiments

F. ˎ each . f he 10 e e ˎ 1 e. a c. . g ˎ a 1. . de c. ı bed ab. e, he ˎ a. d. a. d MAC. a 1 g. che e a ˎ e c. a ˎ ed f. ˎ d1 e ˎ e. . be ˎ f ac 1 ˎ ed. a e . The e a a 1. . e ˎ ıc 1 c. ed a f. . . F. ˎ each ch. ıce. f **x** a. d **y** e ca c a ed he e ec ed e ˎ . ˎ a e e_F. f a. a 1 a . e ˎ 1 ˎ ı c a. 1 e ˎ a1 ˎ ed . he e. 1 ˎ e da aba e (ı.e., 1 h a he. a e . f **c**, **x** a. d **y** . .). The. f. ˎ a g1 e. a e. 1 e L e. a ed **y** a e . . L. a e f. . he da aba e (e1 he ˎ b MAC . ˎ a. d. . a 1 g) a. d ca c a e he ˎ ed1c ed e ˎ . ˎ a e e_L f. ˎ each. e h. d. We he. c. ed he ˎ . . ea. a ˎ e d1 e ˎ e. ce be ee. e_L a. d e_F. . e. e e. a ˎ . . f he. a 1 g. che e. U. de ˎ he a. . 1. . f. 1 c. . f. ˎ fea ˎ e. a e ac 1 1 1. he ˎ . d1 e ˎ e. ce 1 . ea ˎ e he e cac. f a. a 1 g. che e 1. e 1 a 1 g he e ˎ . ˎ a e. f a c a. 1 e ˎ a1 ˎ ed . . b. h **x** a. d **y** a a f. c 1. . f. he. . be ˎ. f fea ˎ e. a e ac 1 ˎ ed.

T. ˎ e a e he e a a 1. . ea ˎ e. ed . he b1. g1ca ˎ . b e , e . . e ha e_F ca. be. 1e ed a he ˎ e e ˎ . ˎ a e. f he c a. 1 e ˎ ha . e he. e b1. a. e ˎ **y** a. d e_L a he e 1 a e. f he e ˎ . ˎ a e af e. a 1 g **y** . . L a e . S1 ce. ˎ g. a 1 . ˎ ed1c he e ˎ . ˎ a e acc ˎ a e . 1. 1 1 1 g L, e ca. ea ˎ e he e ec 1 e. e. . f. ˎ . a 1 g a g. ˎ 1 h b he ˎ . . d1 e ˎ e. ce be ee. e_F a. d e_L.

F. ˎ each e e ˎ 1 e. e . ed he ˎ . . a e aga1. he . be ˎ. f. a e ˎ . bed hıch a ˎ e. h. . 1. Fıg ˎ e 1. I. each . , . c. . a ˎ e he MAC. a - 1 g. che e . he ˎ a. d. . e h. d f. ˎ c. . e ec 1 e. e. , e. . c. . a ˎ e he . . be ˎ. f fea ˎ e. a e. a ed f. ˎ a ˎ e 1 ˎ ed ˎ . . e ˎ . ˎ .

I. . . e . . he ˎ . . a e. a ˎ a. ˎ e. bef. ˎ e 1. c. ea 1 g . a. a 1 . a e. Th1 ha e. 1. 1 a 1. . he ˎ e he. e fea ˎ e **y** add . . e 1 f. ˎ - a 1. f. ˎ ˎ ed1c 1 g he c a. **c** g1 e. he ˎ e 1 . fea ˎ e **x**. The ef. ˎ e 1. he beg1. 1 g. f he. a 1 g ˎ . ce. , he e 1 a ed e ˎ . ˎ a e 1 . he e ˎ . ˎ a e . b a1 ed b . 1 g **x** hıch 1 he ac a. a e. A . . ˎ e. a e a ˎ e added he

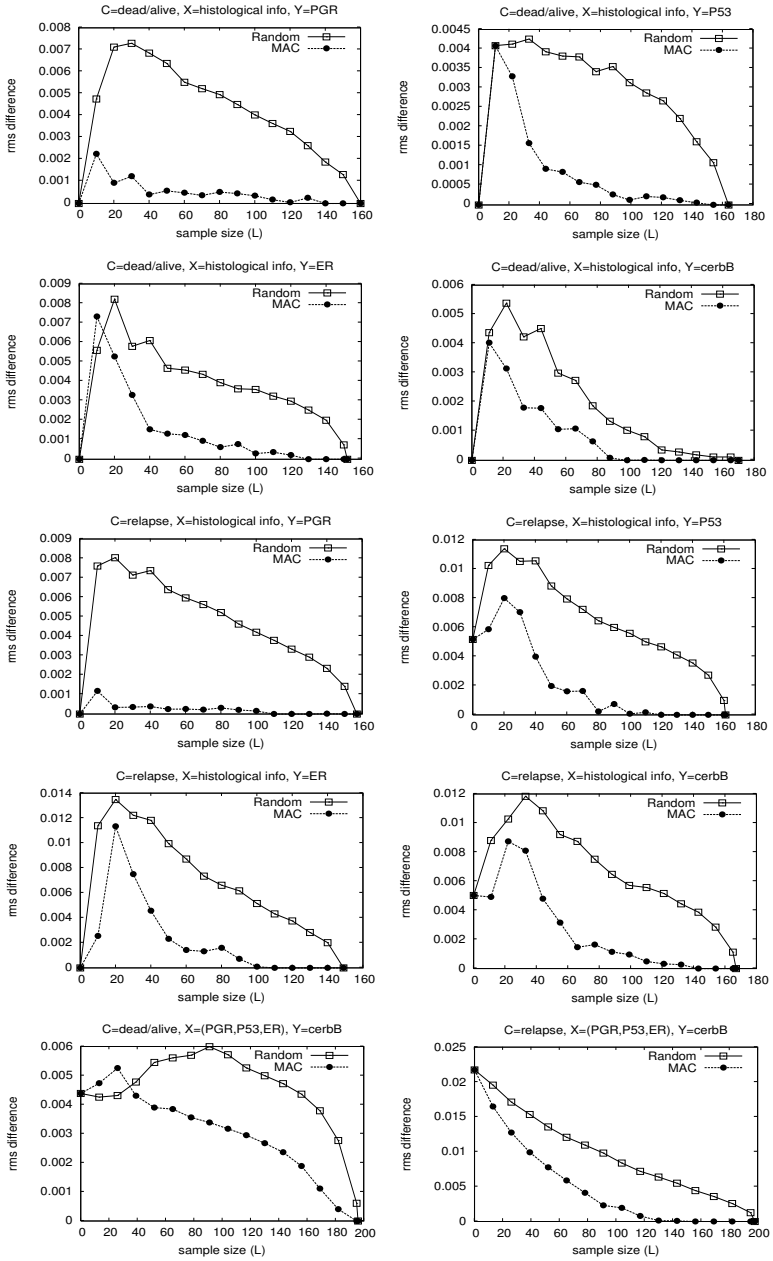


Fig. 1. Plots of the rms difference between the 'true' error rate of the classifier operating on the full feature space and that estimated from the acquired samples for all three sampling schemes as a function of the number of samples acquired. The features chosen for **c**, **x** and **y** are also indicated. The rms value is computed from 500 runs of the sampling experiment for each configuration. The error bars are smaller than the markers used in the plots.

e 1 a e bec e 1 1 1c edıc 1 g a e a e ha he ac a a e 1 cıe a ge be f a e a e added ea he e fea e 1 e e .

We b e e f he ha MAC ac 1 e a 1 g a g 1 h 1 ıg 1 - ca be e 1 ed cı g he be f a e eeded f e a e e 1 a 1 ha a d a 1 g. F e a e 1 he - ef b , e b e e ha b a1 a e a e e 1 a e 1 h a e f 0.001, MAC a g 1 h eed e ha 40 a e a ed ab 150 f a d . Thı beha 1 1 b e ed 1 f he e e 1 e a c g a 1 .

I e f he bı edıca b e , hı 1 ıe ha 1 g he MAC ac 1 e a 1 g, e ca e a a e a hıghe be f bı a e he a e a f bı - a e e ce ha 1 g he a da d a d a 1 g e h d.

6 Conclusions and Future Work

We e e ed a e 1 1 a 1 he b e f e a a 1 g e bı a e ha e ab e be e cha ac e 1 a 1 f ca ce hı e c e 1 g he 1 1 ed a f e a a ed bı gıca a e . We h ed e e 1 e a ha ac 1 e a 1 g a g 1 h h d 1 e 1 acc a e e a a 1 g he e cac f he e bı a e 1 h ıg 1 ca fe e a e e ed. Thı a f he e a a 1 f e bı a e 1 g he bı gıca a e a ha d. O e a e 1 he e cac f he ac 1 e a 1 g a g 1 h 1 1 1 1a e he bı a e a 1 1 ed be f a e dı ca d 1 f a 1 e bı a e a d ceed e he e a1 1 g e e ha 1 e .

The e a e 1 e e a e b e ha eed be add e ed. Wı h he Tı e Mıc a a e h d e e a 1 e a e a e a1 ed 1 h he bı a e 1 a e . The ef e e eed e e d a 1 g a g 1 h b ea- 1 g ab 1 c e e a a 1 g e b a ba ch f a e . A h gh e ca f he a e a a 1 e e ed ab e de 1 e a a g 1 h f he ba ch ca e, he a g 1 h d be c a 1 a e e 1 e. The ef e e eed a e f he a 1 a 1 de 1 e a fea ıb e a g 1 h . Sı ce he Tı e Mıc a a ce 1 c ıca ed 1 he e a a 1 f he ıde , he c 1 ed a e ıf . We 1 e d de e ac 1 e a 1 g a g 1 h f e ge e a c de . We a 1 e d 1 e ıga e ac 1 e a 1 g 1 h he c a 1 ca 1 che e . O g a 1 de e a f f c 1 1 g ac 1 e a - 1 g decı 1 e ha ca be e ed f bı a e e a a 1 1 bı ec a ab a ıe .

References

1. D.A.Cohn, Z.Ghahramani, and M.I.Jordan. Active learning with statistical models. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems*, volume 7, pages 705–712. MIT Press, 1995.
2. F. Demichelis, A. Sboner, M. Barbareschi, and R. Dell'Anna. Tmaboost: an integrated system for comprehensive management of tissue microarray data. *IEEE Trans. Inf. Technol. Biomed.* In Press.

3. D.J.C.MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.
4. D.Lizotte, O.Madani, and R.Greiner. Budgeted learning of naive-bayes classifiers. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 378–385, San Francisco, CA, 2003. Morgan Kaufmann.
5. T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.
6. H.S.Seung, M.Opper, and H.Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM Press, 1992.
7. K.Chaloner and I.Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10:273–304, 1995.
8. J. Kononen, L.Bubendorf, A.Kallioniemi, M.Barlund, P.Schraml, S.Leighton, J.Torhorst, M.Mihatsch, G.Seuter, and O.P.Kallioniemi. Tissue microarrays for high-throughput molecular profiling of tumor specimens. *Nature Medicine*, 4(7):844–847, 1998.
9. M.Saar-Tsechansky and F.Provost. Active Sampling for Class Probability Estimation and Ranking. In *Proc. 7th International Joint Conference on Artificial Intelligence*, pages 911–920, 2001.
10. P.Sebastiani and H.P.Wynn. Maximum entropy sampling and optimal Bayesian experimental design. *Journal of Royal Statistical Society*, pages 145–157, 2000.
11. S.Tong and D.Koller. Support vector machine active learning with applications to text classification. In *Proceedings of the International Conference on Machine Learning*, pages 999–1006, 2000.
12. S.Veeramachaneni, E.Olivetti, and P.Avesani. Active feature sampling for low cost feature evaluation. Technical report, ITC-irst, 2005.
13. V.V.Fedorov. *Theory of optimal experiments*. Academic Press, New York, 1972.
14. Z.Zheng and B.Padmanabhan. On active learning for data acquisition. In *Proceedings of the International Conference on Datamining*, pages 562–570, 2002.

A Multi-metric Index for Euclidean and Periodic Matching

Mıchaı V ach . [1], Z. g. af a Vage. a[2],
Vı . ı. Ca e ı[1], a. d Phı ı S. Y [1]

[1] IBM. T.J. Watson Research Center
[2] University of California, Riverside

Abstract. In many classification and data-mining applications the user does not know a priori which distance measure is the most appropriate for the task at hand without examining the produced results. Also, in several cases, different distance functions can provide diverse but equally intuitive results (according to the specific focus of each measure). In order to address the above issues, we elaborate on the construction of a hybrid index structure that supports query-by-example on shape and structural distance measures, therefore lending enhanced exploratory power to the system user. The shape distance measure that the index supports is the ubiquitous Euclidean distance, while the structural distance measure that we utilize is based on important periodic features extracted from a sequence. This new measure is phase-invariant and can provide flexible sequence characterizations, loosely resembling the Dynamic Time Warping, requiring only a fraction of the computational cost of the latter. Exploiting the relationship between the Euclidean and periodic measure, the new hybrid index allows for powerful query processing, enabling the efficient answering of kNN queries on both measures in a single index scan. We envision that our system can provide a basis for fast tracking of correlated time-delayed events, with applications in data visualization, financial market analysis, machine monitoring/diagnostics and gene expression data analysis.

1 Introduction

E e h gh a ı e- e ıe dı a ce f c ı . ha e bee . . . ed ı he da a- . ı ı g c . . . ı , . . . e . f he ha ece ı ed he a . . ca h ıc acce a ce ha he E c ıdea dı a ce e . . . The E c ıdea . . . ca be c . . ıde ed a he . . dı e a . . , ... , dı a ce ea e, b ı ha bee h . . . e f . . a c . . e ea e ı a a ıe . f c e ı g/c a . ı ca ı . a . [3], hı e ha ı g . . a f ac ı . . f he c . . a ı . a a d . gıca c . . e ı . f he c . . e ı g . ea e .

La e h e e , ı e- e ıe e ea che a e a . . a ı g . ac . . edge ce - a ı ı ı a ı . . . f ha e a chı g dı a ce ea e , a d he ef e e a e g ad- a e e ıe cı g a . hıf . . . e . . . . . ea e . f . ı ı a ı . The e . e . c a ea e ca g ea e ha ce . abı ı . a e he ı he e . ı ı a - ı be ee ı e e e ce a d e d . be . e c he e ı h he ıe g . e . ı g

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 355–367, 2005.

he h a e ce 1 a d c g 1 1 . Rece a 1f 1 g c a he 1 1 a 1 be ee e e ce , a a e 1 c 1de a 1 a a 1e f fea e , ch a cha ge- 1 -de ec 1 [2], e e ce b 1 e [7], ARIMA ARMA ge e a 1 e de [9], a d e e ce c e 1b1 1 [4].

I a ca e h gh, he e 1 c ea 1 d1ca 1 he he a ha e a c a ea e 1 be 1 ed f a a 1c a a 1ca 1 . I he e e ce f a he e ge e da a e , ec1 c e 1e 1gh be ac ed be e 1 g d1 e e ea e . The d1 a ce e ec 1 a bec e e e e cha e g1 g, 1f e c - 1de ha d1 e e d1 a ce ea e ca e 1 e a 1de d1 e e b e a 1 1 1 e ea ch e .

I a e 1 1ga e he d1 a ce e ec 1 d1 e a, e e e a 1 de c e ha ca a e 1- e 1c e 1e ba ed b h ha e a d c- e, a 1 g he e d e c a a e e , e e a d ga 1 e e e ec 1 e he e 1 g e a che . The ed 1 de 1 g che e ea - e b e d he E c 1dea 1 h a c a , ea e. Pe 1 d1c d1 a ce f c 1 e e ece e e ed 1 [8] a d ha e bee h e f e e ec 1 e f a c a e f da a e (1.e., ECG da a, achi e d1ag - 1c , e c). H e e , 1 he 1g1 a a e 1 de 1 g che e had bee ed. Rec g 1 1 g ha he e 1 d1c ea e ca ea 1 (a d c -e ec 1 e) 1de 1f 1 e- h1f ed e 1 f he e e e ce (he ef e e e b 1 g T1 e- Wa 1 g), e e 1 he e a 1 h1 be ee he e c 1dea a d he e 1 d1c ea e 1 he f e e c d a1 , 1 de de 1g a 1 de ha e -b -e a e b h e 1c . B 1 e 1ge ga 1 1 g he e ac ed e- e ce fea e a d 1 e 1 g he e c 1dea a d e 1 d1c ea ch e ca e he -NN a che f b h ea e 1 a 1 de ca . B h e e a e e e ed he e , e a d1 g he 1b1 1 1e f 1 e ac 1 e da a e a 1 , 1d1 g e c e a he a 1a e d1 a ce f c 1 .

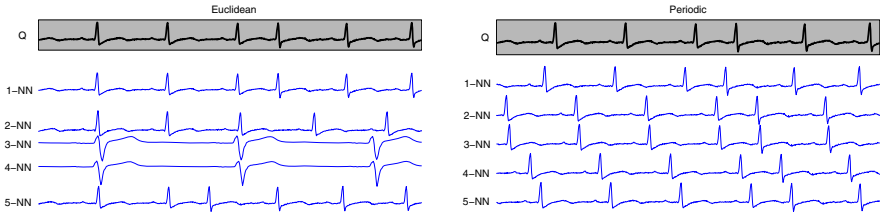


Fig. 1. 5-NN euclidean and periodic matches on an ECG dataset

A a e f he ed 1 de f a da aba e f ECG da a 1 h 1 Fig. 1. F he ec1 c e , a 1 a ce e ed b he e 1 d1c ea e be g he a e c a f e e ce a d c e d 1 e- h1f ed a 1a 1 f he e e e ce. The 1,2,5-Nea e -Ne1ghb (NN) a che f he E c 1dea e 1c ca a be c 1de ed 1 1 a he e , h e e he 3-NN a d 4-NN d be cha ac e 1 ed a 1 a che b a h a . The e f h1

(, a he, 1 1 1c) e a e, 1 . e ha 1 e ha 1 . a ca e . 1 e. ea , e a, e . ece . a , , 1 ce each . e , 1c ca ha, e a d1 e, e . b e . f a . e, . .

E e h gh . he, . 1- e , 1c d1 a ce ha e bee , e e ed 1 [10,6] (b f , d1 e, e . e . f d1 a ce f . c 1 . .), e, 1e . eeded . be 1 . ed . 1 e 1 e f , , e , 1e 1 g he e . . f he d1 e, e . ea , e . The, ef , e, he , e e ed 1 de ha . d1 1 c ad a age :

■ I . . . , . c . c , , e e c 1dea a d e, 1 d1c . a ch1 g, , e , . 1 g b h e . f Nea, e -Ne1ghb , . a che 1 a . . , 1 de . ca . S 1 a . . f , b h , 1g1d . a ch1 g (e c 1dea d1 a ce), . . . , e e 1b e e, 1 d1c . a ch1 g, b 1de 1f 1 g a, b1 , a 1 e hf . . f a e, (e, 1 d1c . ea , e).

■ Pe, f , a ce 1 . . c . , . . 1 ed, b 1 1 fac 1 , . . ed (c . a, ed . he d a 1 de a , . ach) d e . he , ed ced 1 de . 1 e a d he 1 e 1ge , ee , a e, a .

G1 e he ab . e cha, ac e, 1 1c , e e ec ha he . e 1 de . , c , e ca , . 1de . ece . a, b 1 d1 g b . c . f , c . . , c 1 g . e, f ‘a -1 . - . e’ . . . , 1 h1 he . c . e . f a 1ca 1 . . . ch a dec1 1 . . . . . , a a . 1 . f ca . a da a , e a 1 . . h1 . a d da a 1 a 1 a 1 . .

2 Background

The e, 1 d1c . ea , e a d he h b, 1d 1 de ha e 1 de c, 1be a e, . e, a e 1 he f, e e c d . a1 , he, ef , e e 1 . cc1 c de c, 1be 1 . , a c . ce . f . . ha, . . 1c a a . 1 .

2.1 Frequency Analysis

A d1 c, e e- 1 e . 1g a $\mathbf{x} = [x_0, \ldots, x_{N-1}]$. f e g h N ca be h gh . f a a e- , 1 d . f a e, 1 d1c . 1g a a d , e , e e ed 1 e, . . f 1 . F , 1e, - e, 1e c e c1e . $\{X_k\}_{k=0}^{N-1}$ b

$$x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{2\pi j(k/N)n}, \qquad n = 0, \ldots, N-1,$$

he, e $j = \sqrt{-1}$ 1 he 1 ag1 a . 1 . The c e c1e X_k 1 de . ed b

$$X_k = \rho_k e^{j\theta_k} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-2\pi j(k/N)n}, \qquad k = 0, \ldots, N-1,$$

a d c . , e . . d . he f, e e c $f_k = k/N$. He, e ρ_k a d θ_k a, e , e ec 1 e he . ag 1 de a d he ha e . f X_k. Pa, e a ’ he , e . a e ha he e e, g $\mathcal{P}$. f he . 1g a c . . ed 1 he f, e e c d . a1 1 e a . he e e, g c . . ed 1 he F , 1e, d . a1 :

$$\mathcal{P}(\mathbf{x}) = \|\mathbf{x}\|^2 = \sum_{k=0}^{N-1} x_k^2 = \mathcal{P}(\mathbf{X}) = \|\mathbf{X}\|^2 = \sum_{k=0}^{N-1} \|X_k\|^2.$$

Ma . . e, a 1 . . a, e . b a 1a . . , e e c1e 1 he f, e e c d . a1 ha 1 he 1 e d . a1 . The . e . f f, e e c -d . a1 . . e, a 1 . . 1 . f e a ea- 1 g ha . . he e 1 e ce . f he e c1e Fa F , 1e, T, a . f , , h1ch ha c . . a 1 . a c . . e 1 . f $O(N \ g N)$.

3 Distance Functions

3.1 Euclidean Distance

Le **x** a. d **y** be . 1 e. e e. ce . f e. g h N ha. 1. g D1 c. e e F. . 1e. T. a. - f. . **X** a. d **Y**, . e ec 1 e . The E c 1dea. d1 a. ce $d(\mathbf{x},\mathbf{y})$ be ee. **x** a. d **y** (1.e., he ℓ_2 . . . . f $\mathbf{x}-\mathbf{y}$) 1 de . ed b $d(\mathbf{x},\mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})\cdot(\mathbf{x}-\mathbf{y})} = \sqrt{\sum_{k=1}^{N}|\mathbf{x}_k-\mathbf{y}_k|^2}$, he. e · de. . e he 1 . e. . . d c . Pa. e a ' The . e e. - . e ha $d(\mathbf{x},\mathbf{y}) = d(\mathbf{X},\mathbf{Y})$. We ca. dec. . . . e he E c 1dea. d1 a. ce 1. . he. . . f he. ag. 1 de d1 a. ce a. d a. . . - ega 1 e e. 1. . . 1 g b. h. ag- . 1 de a. d ha e :

$$
\begin{aligned}
[d(\mathbf{x},\mathbf{y})]^2 &= \sum_{k=0}^{N-1}\|x_k-y_k\|^2 = \sum_{k=0}^{N-1}\|\rho_k e^{j\theta_k}-\tau_k e^{j\phi_k}\|^2 \\
&\overset{(a)}{=} \sum_{k=0}^{N-1}(\rho_k \mathrm{c}. . (\theta_k)-\tau_k \mathrm{c}. . (\phi_k))^2+(\rho_k . 1 (\theta_k)-\tau_k . 1 (\phi_k))^2 \\
&\overset{(b)}{=} \sum_{k=0}^{N-1}\rho_k^2+\tau_k^2-2\rho_k\tau_k\big(. 1 \theta_k . 1 \phi_k + \mathrm{c}. . \theta_k \mathrm{c}. . \phi_k\big) \\
&\overset{(c)}{=} \sum_{k=0}^{N-1}(\rho_k-\tau_k)^2+2\sum_{k=1}^{N}\rho_k\tau_k\,[1-\mathrm{c}. . (\theta_k-\phi_k)],
\end{aligned}
\tag{1}
$$

he. e (a) 1 he P hag. ea. he. e , (b) f. . . f. . a geb. a1c. a. 1 a 1. . a. d e e e. a. . 1g. . . e. 1c 1de. 1 1e , a. d (c) f. . . b addi. g a. d . b- . ac 1. g $2\rho_k\tau_k$. (b), c. ec 1. g e. . , a. d . 1. g a. e e e. a. . 1g. . . e. 1c 1de. 1 . Ha. 1. g e . e. ed he E c 1dea. d1 a. ce . 1. g . ag. 1 de a. d ha e e. . , e e . . e 1 . c. . . ec 1. . 1 h a. e. 1. d1c. ea . e 1. he f. . 1. g. ec 1. . .

3.2 Periodic Measure

We . e e. a d1 a. ce. ea . e ha ca. a. 1f he. . c . a . 1 1 a. 1 . f. e- e. ce ba ed . . , . . fea . e e . ac ed f. . he . The e. 1 d1c. ea . e a. d1 c . . ed, . ge he. 1 h a 1ca 1. . . , 1. [8] a. d 1 e 1ca ed he. e f. . c. . - e e. e. . f . e e. a 1. . . I. h1 . . e. a e he c. . . ec 1. . 1 h e c 1dea. d1 a. ce 1. he f. e e. c d. . a1. a. d. h. h. . c. . b1. e b. h 1. a. e c1e. 1. de .

The 1. . . d c 1. . . f. e. 1. d1c. ea . e 1. . . 1 a ed b he 1. ab1 1 . f. ha e- ba ed . ea . e . ch a he E c 1dea. . ca . e acc . a e he . d1 e. a. h. a. . . 1. . . f . 1 1 a. 1 be ee. . . 1g. a. . F. . e a e . . e e. ce ha a. e 1de. 1ca e ce f. . a. . a 1 e. h1f . h. d be c. . . 1de. ed . 1 1 a. 1. a a. 1e . f a 1ca 1. . . (Fig. 2), 1. . . 1 e. f he . e. 1a a. ge e c 1dea. d1 a. ce be ee. he . The. ef. . e, he e. 1. d1c. ea . e . . . e . e e b e 1 e- a. 1. g . ea . e , . e 1. 1. g. . . 1. ea. c. . a 1. . a c. . e 1 , h . . e. de. 1. g 1 . e. . 1 ab e f. . a. ge da a- . 1. 1. g a . .

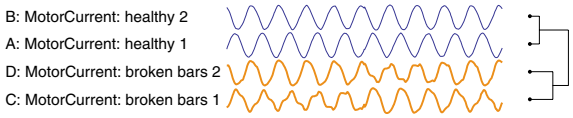


Fig. 2. Dendrogram on 4 sequences using a periodic measure

3.3 Periodic Distance (pDist)

T a e he e 1 dıc 1 1 a 1 f e e ce e e a 1 e he d1 e e ce f he1 ha ıc c e . We de e he e 1 dıc d1 a ce be ee e e ce **x** a d **y**, 1 h F ıe a f **X** a d **Y**, e ec 1 e a he e c ıdea d1 a ce be ee he1 ag 1 de ec :

$$[pDist(\mathbf{X},\mathbf{Y})]^2 = \sum_{k=1}^{N} (\rho_k - \tau_k)^2.$$

N ıce ha he 1 1 f he ha e 1 f a 1 e de he e 1 1 a 1 ea e hıf -1 a ıa 1 he 1 e d a1 , a 1 g f g ba 1 e- hıf 1 g 1 $O(n)$ 1 e (a he a e a 1 e d be he e f 1 g T1 e-Wa 1 g 1 h $O(n^2)$ c e 1).

I de ea 1 gf c a e he ec a e d1 ıb 1 f e e ce 1 he da aba e, e a 1 e he c a1 he a e a f e e g b de 1 1 g he (h d c1 g e - ea , 1 -e e g e e ce):

$$x(n) = \frac{x(n) - \frac{1}{N}\sum_{i=1}^{N} x(i)}{\sqrt{\sum_{i=1}^{N}(x(n) - \frac{1}{N}\sum_{i=1}^{N} x(i))^2}}, \quad n = 1, \ldots, N$$

F he e f he a e e 1 a e ha a da aba e e e ce a e - de 1 ed (he he e a e c ıde 1 g e c ıdea e 1 dıc d1 a ce). We a e a a e f he e 1 dıc d1 a ce ha 1 be ef ıde e e ec 1 e a e a f 1 de 1 g c e.

Lemma 1. [illegible] $pDist(\mathbf{X},\mathbf{Y}) \leq d(\mathbf{X},\mathbf{Y})$

Th1 e a 1 ed b 1 g ha he he RHS E a 1 1 1 he e 1 dıc d1 a ce, a d ha he ec d 1 - ega 1 e.

4 Lower Bounding and Coefficient Selection

I de e cıe 1 c a e a d1 a ce ea e 1 h a 1 de 1 g c e, e eed : (1) c e a e e ce (d1 e 1 a 1 ed c 1) (11) ıde a e b d1 g f c 1 f he ıg1 a d1 a ce 1 g he c e ed b ec . We 1 h h b h f he e ca be achıe ed 1 a e ec 1 e a .

Af e a e e ce 1 a f ed 1 he f e e c d a1 , 1 ca be c - e ed b ec d1 g a a b e f1 c e cıe . The ef e, $\{X_k\}_{k=0}^{k=N-1}$

$\rightsquigarrow \{X_k\}_{k \in S}, \quad S \subset \{0, \ldots, N-1\}, \quad |S| << N$. I ı . ı aıgh f ı a d . . h ha he e c ıdea . . e ı ı dıc dı a ce . . he c . . ı e ed ec . . ı . e b . . d he . ı gı ı a dı a ce , beca . e he a e a . . . f . . ı ı ı e . . be . :

$$d(\mathbf{X}_k, \mathbf{Y}_k)_{k \in S} \leq d(\mathbf{X}, \mathbf{Y})$$

$$pDist(\mathbf{X}_k, \mathbf{Y}_k)_{k \in S} = \sum_{k \in S} (\rho_k - \tau_k)^2 \leq pDist(\mathbf{X}, \mathbf{Y})$$

The . a . ı . f da a- ı ı g . . ha ada ed he . e ec ı . . f he . . k c . e cıe . . f . . e e ce c . . . e . ı . . [5], hıch ca . . . ıde e ec ı e a . . ı a- ı . . f . ıg . a . ı h . f e e c c . e (e.g., . . c . ıce . . e e). Rece . . . a . . gge ed ıc ı g he k c . e cıe . . f . . each . e e ce ha . e e . e . . . f ı . e e g [7]. Hıgh e e g c . e cıe . . ca . . . ıde e ec ı e . e e ce . ec . . . c ı . , b a e . . . ece . a ı . ı ab e f . . da a . e . ıe a . . . e (ı.e. ı c . he ea . . . be . . f acce . e . he . . ıgı . a da a). C . . . ıde . a da a e he e he k c . e cıe . . ı h he hıghe e e g a e he . a e f . . a . e e ce , a d he . e e ce . . dı e . ı he . e e g c . e cıe . . (ı.e. he e a e . . . e . a b dı ı . c . a ce be ee each . e e ce). I hı ca e, . . e eed . . e ec he c . e cıe . . ha ı . . . . he . e e ce . Ge e a . ea ı g, e ca ca . e . . e e ec ı e he da a dı e e ce b . ec . dı g h . e c . e cıe . . ha acc . . f . . . . . f he da a a ıa ı . . Wı h hı . b e . a ı . ı . ı d, e ı . ec . d he k c . e cıe . . ha de ıc ha a ge . a ıa ce:

$$arg \ \underset{k}{a} \ var(X_k^{(j)})_{j=1 \ldots m}$$

he e X_k^j de . . e he k h c . e cıe . . f . e e ce j. We c . . a e he e f . - a ce . f a ı . . c . e cıe . . e ec ı . . che e ı h a c . . . ehe . ı e e e ı e . . 40 da a e . (each c . . a ı . ı g 1000 . e e ce . f e g h 1024), . b a ı ed f . . he UCR ı e- e ıe a chı e [1]. We e f . . a 1-NN ea e- . e- . . ea ch a d e ı a e he . . ı g . e . f each . e h d a gı e b he . a ı : (. . . . . . .) (. . . . .). The . e h d f . . c . e cıe . . e ec ı . . ha e c . . ıde a e: (ı) . . . c . e cıe . . , (ıı) c . e cıe . . ı h . a ı . . e e g (ıı ı) c . e cıe . . ı h . a ı . . a ıa ce.

The . e . a e . he . e ı . ı . f he ‘ a - a ıa ce’ . e h d f . Nea e - Neıghb . . e . ıe a , de ıc ı g a . ı . . e e . ı 21 . . . f he 40 da a e . . The a e age ı . . . e e . . e, he . e be . e h d ı 17.17% (ı h a . a ı . . f > 80% ı he . . . da a e). Eıgh da a e . d . . e hıbı a cha ge a a , hıch ı . b e . ed beca . e he . e . f c . e cıe . . . e ec ed b he 3 . e h d e . e he . a e. F ı . a , 11 da a e . de ıc a de e . ı . a ı . ı he -NN e f . . a ce hıch h e e . ı a . . . . eg ıgıb e (. e e e ceedı g -0.5%), ı h he a e . age . ega ı e ı . . . e e . be ı g -0.13%. F . . he . e a ı . de . . f he a e . , e ı a . . e ha he c . e cıe . . . e ec ed f . . each ı e- e ıe a e he . . e e hıbı ı g he a ge . a ıa ce.

[1] http://www.cs.ucr.edu/~eamonn/TSDMA/

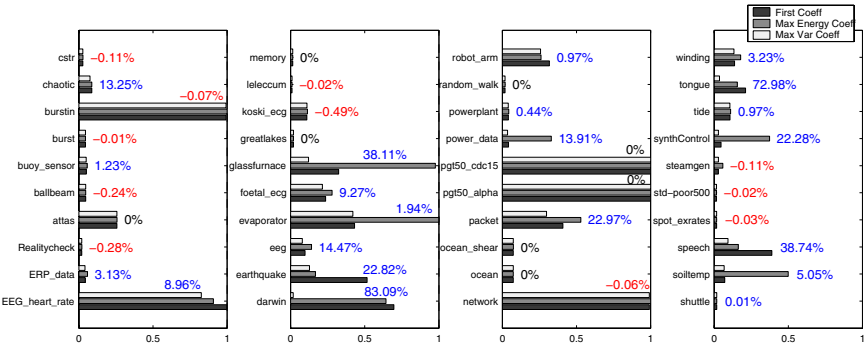


Fig. 3. Comparison of coefficient selection (smaller numbers are better). Improvement of *max-variance* method vs second best is reported next to each performance bar.

5 Index for Euclidean and Periodic Distance

I. ead. f c. . . c 1 g a d1 e. e. 1 de . c e f. each. ea e, e ca. e . 1 he c. . . . e e e a 1 . . f he e c 1dea. a. d e. 1 d1c. ea e 1. he. ag. 1 de/ ha e. ace, a e a he1. e. b. d1 g e a 1. . h1 , f. de 1g. 1 g a. e. 1c 1 de . c e ha ca. 1 a. e. a. e. e. 1e . . b. h. ea e . O 1 de . c e b. . . . 1dea f. . he fa 1 . f. e. 1c 1 - de . c e [1], ec. 1 e a. 1 1. 1 g he. ea. ch. ace 1. d1 1 eg1. . , ba ed. . he e a 1 e d1 a. ce be ee. b ec . . O 1 de 1 g a . . ach ha h. ee 1 . . a d1 e. e. ce f. . ge. e. 1c. e. 1c ee : (1) . . c. e. ed. e e. ce a. e. . . ed 1 h1 he 1 de . . de , . ed c1 g he 1 de . ace. e 1. e e. . , (11) he 1 de . e a d1 e. e. d1 a. ce. ea e. f. da a. a. 1 1. 1 g. . each a e. - . a 1 g e e. f he . ee, (111) a . . . e . ee . a e. a 1 . e e. ed, ha ca. a. e. e c 1dea. a. d e. 1. d1c e. 1e 1. a. 1 g e 1. de . ca. .

5.1 MM-Tree Structure

We 1. . . d ce a h b. 1d. e. 1c. . c e, he MM-T. ee (M 1-Me. 1c T. ee). S1 - 1 a. . VP- . ee , each . . de. f he 1 de c. a1. a. efe. e. ce . 1 (. . a. age . 1), h1ch 1 . ed . a. 1 1. he . 1 . a. . c1a ed 1 h h1 . . de 1. . d1 1. c a. d e a . 1 ed c . e. . Va. age . 1 . a. e ch. e. be he. e e. ce 1 h he h1ghe . a. 1a. ce. f d1 a. ce . he. e a1. 1 g. b ec. . The d1 a. ce . f he . . de. b ec. . he. efe. e. ce . 1 (e. e. ce) a. e ca c a ed, d1 a. ce a. e. . . ed a. d he. ed1a. d1 a. ce μ 1 1de. 1 ed. S b e e , a . e e. ce a . . c1a ed 1 h he e a 1 ed . . de, 1 a. 1g ed . a. . . . . a. . . . . . . , de- e. d1 g. . he he. 1. d1 a. ce f. . he. a. age . 1 1 . . a. e. . . a. ge. ha he. ed1a. d1 a. ce. The 1 de . ee 1 c. . . c ed b . ec . . 1 e e. f. . 1 g h1 . e. a 1. f. a . b . ee .

The . 1 e. . c e. f he MM-T. ee de. 1 e f. . he fac ha 1 . e a . . . . d1 a. ce f. c 1. . . a. 1 1. . b ec. a. each a e. . a 1 g . ee e e .

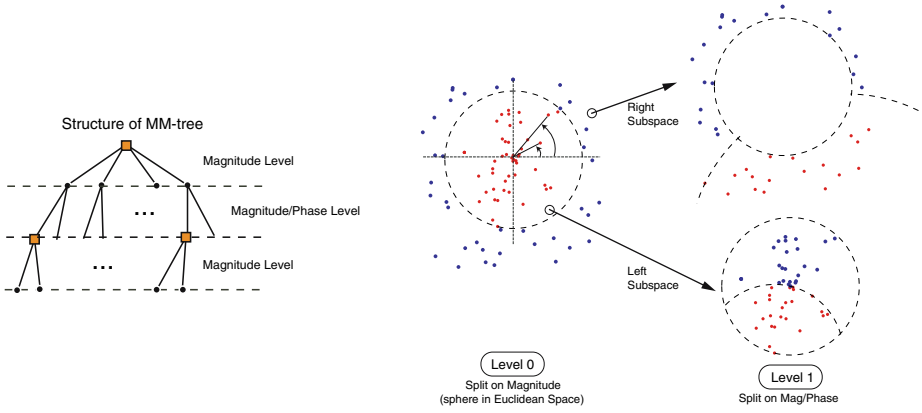


Fig. 4. MM-tree structure and space partitioning. Dotted circles/arcs indicate median distance μ.

E e de h e e (. . 1 e.) a e a 1 1. ed . 1 g he e 1 dıc dı a ce, hı e . dd e e. 1 1 e he e c ıdea (Fıg. 4). We f. hı c. . . c 1 . f . . . ıdı g a g. d a 1 1. 1 g . . b h dı a ce. ea e , 1 ce . age . f a . 1 g e dı a ce f . c 1 . d 1 g he ee c. . . c 1 . . d ha e 1 ac ed he ea ch . . he he d . aı (. e ıa eadı g . . ea ch . . b h ef a d ıgh . b ee).

U 1 e. he . e ıc 1 de 1 g . . c e , 1 e. edıa e ee . de c . aı he . . , . . . e e e. a 1 . . f a a age . 1 (1 addı 1 . . he edıa dı a ce μ). F e a e, 1 he ee . f Fıg e 4 he a age . 1 . . f . . de a e e. de h a e e e e. ed b he . ag 1 de . f he e e. ed c e cıe . (a d a e ca ed P- . de), hı e h . e . f . . de a . dd de h a e e e e. ed b b. h . ag 1 de a d ha e (a d a e ca ed E- . de). F1 a , eaf . . de c . . aı b. h . ag 1 de a d ha e 1 f . a 1 . . f he c . . e ed da a . e e ce .

5.2 Multiplexing Search for Periodic and Euclidean Distances

We . . de c ıbe h he MM-T ee ca e cıe . 1 e . ea che 1 he e c ıdea a d e 1 dıc . ace a d . 1 a e . . e . . he . ea e - eıghb . . 1 b. h d . aı . . The e ıdea . f he ea ch a g 1 h 1 . ıde ıf , 1 a . 1 g e 1 de a e. a , he . 1 . . f he ece a 1 de . . de f . b. h e ıe . I . g e 5 e . . ıde a . e d c. de . f he . 1 e ed . ea ch.

The c . . bı ed . ea ch e . . . . . ed 1 . , $BEST_p$ a d $BEST_e$, ha . aı aı he c . . e k c . . e . 1 . . 1 g e 1 dıc a d e c ıdea dı a ce , e ec 1 e . The a g 1 h a . . ec . d a . a e, de ıc 1 g he he a 1 1 ed . . de 1 . a ed f . . ea ch 1 he e c ıdea , . . 1 he e 1 dıc d . aı , . . b. h. The . . . . de 1 . a ed f . . ea ch 1 b. h d . aı . .

Searching a P-node node. If he . . de 1 . a ed f . . ea ch . . 1 he e - c ıdea d . aı , b. h . b ee a e. ea ched 1 he e c ıdea d . aı . . . O he - 1 e, he a g 1 h c . . e he . e. b . . d $LB_p(q, v)$. f he e 1 dıc dı a ce

```
/* perform 1-NN search for query sequence Q */
1NNSearch(Q) {
   // farthest results are in Best_P[0] and Best_E[0]
   Best_P = new Sorted_List(); // Modified by searchLeaf_Periodic
   Best_E = new Sorted_List(); // Modified by searchLeaf_Euclidean
   search_Node(Q, ROOT, TRUE);
}

search_Node(Q, NODE, searchPeriodic) {
   if (NODE.isLeaf) {
     search_Leaf(Q, NODE, searchPeriodic);
   } else {
     search_Inner_Node(Q, NODE, searchPeriodic);
   }
}

search_Inner_Node(Q, NODE, searchPeriodic) {
   add_Point_To_Queue(PQ, vantagePoint, searchPeriodic);
   if (NODE.E_NODE) { /* E-Node */
     if (searchPeriodic) {
       search_Inner_Node(Q, NODE.LEFT, searchPeriodic);
     } else { /* only search in euclidean space */
       if (LowerBoundEuclidean(Q, vantagePoint) - Best_E[0] < median)
         search_Inner_Node(Q, NODE.LEFT, searchPeriodic);
     }
   } else { /* P-Node */
     if (searchPeriodic) {
       if (LowerBoundPeriodic(Q, vantagePoint) - Best_P[0] < median)
         search_Inner_Node(Q, NODE.LEFT, searchPeriodic);
       else
         search_Inner_Node(Q, NODE.LEFT, FALSE);
     } else { /* only search in euclidean space */
       search_Inner_Node(Q, NODE.LEFT, searchPeriodic);
     }
   }
   search_Inner_Node(Q, NODE.RIGHT, searchPeriodic);
}

search_Leaf(Q, NODE, searchPeriodic) {
   if (searchPeriodic) search_Leaf_Periodic(Q, NODE);   // update Best_P
   search_Leaf_Euclidean(Q, NODE); // update Best_E
}
```

Fig. 5. Multiplexing euclidean and periodic search on the MM-Tree

be ee he a age 1 f he de a d he e e ce. Le r_p be he e 1 dıc dı a ce he fa he e 1 $BEST_p$ he e . N 1 g ha :

$$median < LB_p(q, v) - r_p \Rightarrow median < pDist(q, v) - r_p,$$

he e $pDist(q, v)$ 1 he e 1 dıc dı a ce f he c e d 1 g c e ed e e ce , e c c de ha he a g 1 h h d ea ch he ef b ee 1 he e c ıdea d a1 (b 1 he e 1 dıc) ıf $median < LB_p(q, v) - r_p$.

Searching an E-node node. If he de 1 a ed f ea ch 1 b h d a1 , a b ee a e ea ched 1 g b h ea e . O he 1 e, he a g 1 h c - e he e b d $LB_e(q, v)$ f he e c ıdea dı a ce be ee he a age 1 f he de a d he e e ce. Le r_e be he e c ıdea dı a ce f he fa he e 1 $BEST_e$ he e . The , ıf $LB_e(q, v) - r_e > median$ he ef b ace 1 dı ca ded. I 1 a 1 a e ha , f b h e f de , 1 ce he a age 1 1 1 c e ed f a d e e e b d dı a ce , e d ha e cıe 1 f a 1 dı ca d he ıgh b ee, e e ad he c e ed e e e a 1 f he a age 1 .

A g. ba , 1. 1 e e PQ, h. e , 1 , 1 1 de . ed b he . e. b. . d . f he e. 1 dıc dı a. ce , 1 e . ed, 1. . , de. . e cıe. ıde. ıf he e. , e- . . I. a. ıc a. , he. e e. he c. . , e . ed , e , e e. a 1. . f a da a. e e. ce v 1 acce . ed, he . e. b. . d . f he e. 1. dıc dı a. ce $LB_p(q, v)$ be ee. v a. d he e. . e e. ce 1 c. . ed a. d he a1. $(LB_p(q, s), s)$ 1 . hed 1. . PQ. Whe. a. e e. ce s 1 . . ed f. . he PQ, he a . . cıa ed . e. b. . d . f he e. 1. dıc dı a. ce $LB_p(q, v)$ 1 c. . a. ed agaı. . he c . . e. $BEST_p[0]$ a. d $BEST_e[0]$. a - e . If $LB_p(q, v)$ 1 a. ge. ha. b. h . f h. e . a e , he . e e. ce 1 dı ca. ded. H. e e. , ıf 1 1 . . a e. ha. $BEST_e[0]$, he . e. b. . d . f he e c ıdea. dı - a. ce $LB_e(q, v)$ 1 c. . ed a. d ıf 1 1 a. ge. ha. he $BEST_e[0]$. a e, he . e e. ce ca. . 1 be . afe dı ca. ded. I. a . he. ca e , he . c. . , e . ed . e- e. ce 1 . aded f. . dı a. d he ac a e. 1. dıc a. d e c ıdea. dı a. ce a. e c. . ed . de e. . 1. e he he. 1 be . . g . a. . f he $BEST_p$. , $BEST_e$ 1 . .

6 Experiments

We de . . . , a e he e. f . . a. ce a. d . ea. 1. gf . e . . f . e . . f he MM- . ee f . , a. . e. 1. g . 1 a. e. . e. ıe . . b. h e c ıdea. a. d e. 1. dıc dı a. ce . ea . e . The e e. 1 e. . , e ea ha he . e 1 de . . e. . be e. , e . . . e 1 e a. d . ed ced . . , age . e 1. e e. . c. . a. ed . he a e. . a 1 e a . . ach . f . 1. g . dedıca ed 1. dıce , . . e f. , each dı a. ce . ea . e.

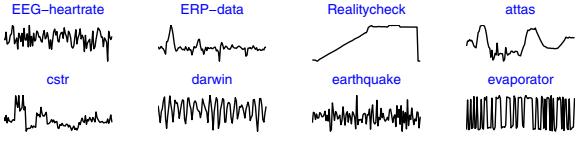


Fig. 6. Sample from the Mixed-Bag dataset

F. , . , e e. 1 e. . e . ed he . a e . 1 . , e . f 40 da a e . ha a 1- 1 ed 1. he c. e cıe. . e ec 1. . ec 1. . , a . a e . f hıch 1 de ıc ed 1. . g . e 6 (da a e). We . ed hı da a e . c. ea e a. ge. da a e . 1 h 1. c. ea 1. g da a ca. dı. a 1 ıe . f 4000, 8000, 16000 a. d 32000 . e e. ce , 1. . , de. . a. ıf he 1. de . . , age . e 1. e e. . a. d 1. . ca abı 1 . A . ed da a e . ca. be . b a1. ed b e a1 1. g he . . a h . , .

6.1 Matching Results

We de ıc he . ea. 1. gf . e . . f . e . . e . . ed b he MM- . ee 1. de , he . ea. chı. g 1. b. h e c ıdea. a. d e. 1. dıc . ace . U. 1. g he da a e e . e . ıe e he 5-NN . f . a. 1. . e. ıe a. d he . e . a. e . ed 1. Fıg . e 7. I 1 1 . edıa e a a. e. ha he e. 1. dıc. ea . e a a . . e . . . e e. ce 1 h g. ea . . c . a a . 1 (ı.e., be . . g . he . a e da a e). The e c ıdea. . ea . e

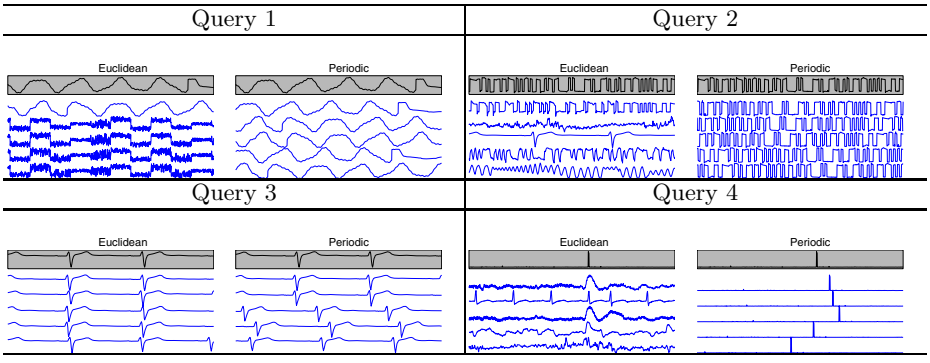


Fig. 7. 5-NN euclidean & periodic matches using the MM-tree (*MIXEDBAG* dataset)

[illegible]

6.2 Index Size

The MM-[illegible] [illegible]

6.3 Index Performance

[illegible]

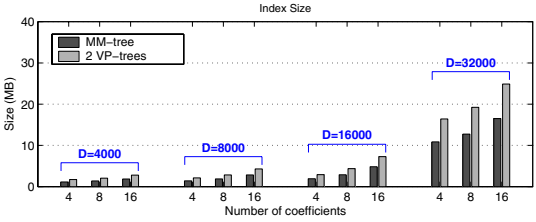


Fig. 8. Index size of MM-tree vs two index structures (euclidean & periodic)

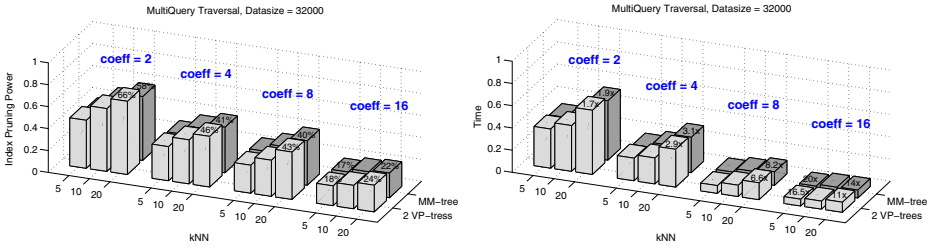


Fig. 9. Performance charts: MM-tree vs 2 VP-trees. Improvement over sequential scan is reported on top of each bar. *(Left)* Pruning power, *(Right)* Running time.

. a dedıca ed . e ,ıc VP- ,ee ı de). Pe f . . a ce c . a ı . . a e c . d c ed ı h VP- ,ee , ı ce he ha e bee . h . . ha e . e ı , e f . . a ce ha . he . e ,ıc . , c ,e , a e a R- ,ee [1]. The ı de e f . . a ce cha . a e ,e . , ed a a f ac ı . . f he c . ı c , ,ed b he e e ıa . ca . f he da a f , he . a e . e a ı . . F . , e e ıa . ca , he da a a e , a e , ed . . . ce hı e . aı aı ı g 2 , ı , ı e e , each . . e h dı g he NN . eıghb . . . f he e- cı c dı a ce f . c ı . . F . . he g a h ı ı a a e ha he e f . . a ce . f he MM- ,ee e . ede he d a ı de e ec ı . . The g ea e e f . . a ce . a gı ı . b e . ed he . e aı ı g he a ge . . be . f c e cıe . e . e e ce, he e he . eed . f he MM- ,ee ca be 20 fa e ha he . e e ıa . ca , hı e he ı dı ıd a . e ,ıc ,ee a e 16.5 ı e fa e . .

Thı . . a e e ı e dı a . he f . e ıa . f he , . . ed h b ıd . , c- ,e. The MM- ,ee d e ı . . ı e . , c ,e . e f . . . he dedıca ed . e ,ıc ,ee . , c ,e he a . e ı g b . h dı a ce e ıe a he . a e ı e, beca . e ı ca c ec he e . . f b . h dı a ce . ea ,e ı a . ı g e ,ee ,a e . a .

7 Conclusion

We ha e ,e e ed a h b ıd ı de . , c ,e ha ca e cıe . . ı e e ıe . . e c ıdea a d e ı dıc . ace . The e ı de a . ca e dı . ace . . e - dıcı . c . a ed . . dedıca ed ı de . , c ,e , a d ı . . ı e . , c ,e a . . f . . , e e ec ı e ,ee , a e . a , , e . . ı g -NN . a che . . . dı a ce

. ea , e 1 he . 1 g e 1 de . ca . We h e ha . , . . e ca . , . ıde he . ece . a b ı dı g b . c . f . c . . . c ı g . e f 'a -ı - . e' . . . , ı hı he . c . e . f a ıca ı . . . ch a decı ı . . . . . , a a . ı . f ca . a da a , e a ı . - . hı . a d da a . ı a ı a ı . .

References

1. A. Fu, P. Chan, Y.-L. Cheung, and Y. S. Moon. Dynamic VP-Tree Indexing for N-Nearest Neighbor Search Given Pair-Wise Distances. *The VLDB Journal*, 2000.
2. T. Ide and K. Inoue. Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In *Proc. of SDM*, 2005.
3. E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. In *Proc. of SIGKDD*, 2002.
4. E. Keogh, S. Lonardi, and A. Ratanamahatana. Towards parameter-free data mining. In *Proc. of SIGKDD*, 2004.
5. D. Rafiei and A. Mendelzon. On Similarity-Based Queries for Time Series Data. *In Proc. of FODO*, 1998.
6. M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures. In *Proc. of SIGKDD*, 2003.
7. M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identification of Similarities, Periodicities & Bursts for Online Search Queries. In *Proc. of SIGMOD*, 2004.
8. M. Vlachos, P. Yu, and V. Castelli. On periodicity detection and structural periodic similarity. In *SIAM Datamining*, 2005.
9. Y. Xiong and D.-Y. Yeung. Time series clustering with arma mixtures. In *Pattern Recognition 37(8)*, pages 1675–1689, 2004.
10. B.-K. Yi and C. Faloutsos. Fast Time Sequence Indexing for Arbitrary Lp Norms. *In Proceedings of VLDB, Cairo Egypt*, Sept. 2000.

Fast Burst Correlation of Financial Data

Mıchaı V ach . , K . -L . g W , Sh h-K eı Che . , a. d Phı ı S. Y

IBM. T.J. Watson Research Center,
19 Skyline Dr, Hawthorne, NY

Abstract. We examine the problem of monitoring and identification of correlated burst patterns in multi-stream time series databases. Our methodology is comprised of two steps: a burst detection part, followed by a burst indexing step. The burst detection scheme imposes a variable threshold on the examined data and takes advantage of the skewed distribution that is typically encountered in many applications. The indexing step utilizes a memory-based interval index for effectively identifying the overlapping burst regions. While the focus of this work is on financial data, the proposed methods and data-structures can find applications for anomaly or novelty detection in telecommunications and network traffic, as well as in medical data. Finally, we manifest the real-time response of our burst indexing technique, and demonstrate the usefulness of the approach for correlating surprising volume trading events at the NY stock exchange.

1 Introduction

Pa a, heı , aıd He a ı . ; e e hı g ı ‘ı ’. The h f hı fa a h - ı b he a cıe G ee hı he ı ch e a ıd da . Pe e eed a e decı ı ab a cıa , e a ı e - e a a e ba ed he b e a ı f a ı fac ı g a a e e . The ef e, ı ce e e hı g ı ı c - a , ı ı g he a ı ı / a ıabı ı f ı a ea e e e ı e, bec e a c ı ıca de e ı a ı a decı ı a ı g ce .

Whe dea ı g ı h ı e e e ce , ı e- e ıe da a, e ı a ı dı- ca f cha ge ı he e e ce f ‘b ı e ’, hıch gge ha e e e f ı a ce a e ha e ı g ı hı he a e ı e f a e. The ef e, he ıde - ı ca ı f b ca ıde ef ı ıgh ab a ı ı e cha ge ı he ı ı g a ı , a ı g he e a a ı dı ıd a ac a ı e a d ı f ed decı ı .

M ı ı g a d de ı g f b beha ı ı ıg ı ca ı a a ea ; a d f e , ı ı ge e a ec g ı ed ha e af- c ca be b ı a ı ı e- ca e [9,6]. De ec ı f b ı ı he ef e ı he e ı a f ıde ı ca ı f e b e ec f , ı ce a e ce ı e a f ı c ı g ac e a be a a ıd ı dıca- ı ha a e e ı de a ac [13]. Addı ı a , f a ıca ı ch a ı ı e c ı ıca e cıe ec g ı e a a a ac ı ı (ıca ı he f f e - ı ı a ı f e ce). F e a e, b

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 368–379, 2005.

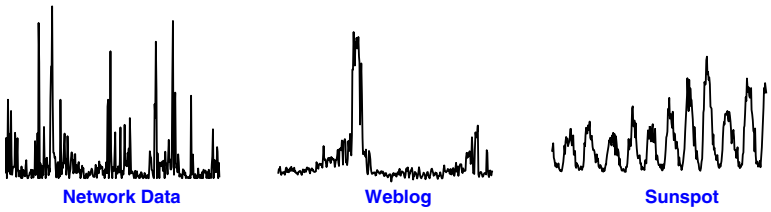


Fig. 1. Burst examples in time-series data

de ec 1 . ech 1 e ca be f 1 f 1 1 ed f . 1 g . ıc1 . ac 1 1 ıe 1 a ge . c . adı g . e [10] . f . ıde 1 ca 1 . f f a d e h . e ac 1 1 [12]. F1 a , 1 , cıe 1 a e 1 e e ed 1 he ea de ec 1 . f a dı ea e b ea . Thı a be 1 dıca ed b he dı c e f a dde 1 c ea e 1 he be f 1 e e 1 1 he d c 1 hı a ce a1 ge g a hıc a ea [16,17].

Ma ece add e he b e f b de ec 1 [19,7]. H e e , 1 a dı cı 1 e , e e ec 1 e edge dı c e ca be achie ed b ıde ıf - 1 g b he 1 1 g 1 e da a ce . F a da a- 1 1 g e ec 1 e, hı a 1 e e cı 1 g a d cha e gı g, 1 ce 1 1 e he ıde - 1 ca 1 f b 'c e ' a d 1 ca a aıd he dı c e f ca a chaı f b e e , hıch ıb cc ac 1 e da a ea . I a ce f he ab e b e ca be e c e ed 1 a a cıa a d c a e a ıca 1 , e.g., f ıgge 1 g f a d a a . F1 a , b c e a 1 ca be a ıcab e f he dı c e a d ea e e f ge e c e e 1 (1 hı e d, b a ea de he e ' - eg a 1 '), hıch h d b a ıa bı gıca ıg 1 ca ce, 1 ce 1 ca ıde 1 ıgh 1 f c 1 a e a ed g f ge e a d e1 [5].

Add e 1 g he ab e 1 e , hı a e e e a c e e f a e f e ec 1 e 1- ea b c e a 1 . S1 1 a [15], e e e e de ec ed b a a 1 e 1 e a f he1 cc e ce. We ıde a e b de ec 1 che e, hıch 1 a1 ed f e ed dı ıb 1 , ch a he a cıa da a ha e e a 1 e he e. Addı 1 a , e 1 d ce a e -ba ed 1 de c e f ıde 1 ca 1 f e a 1 g b . The e 1 de c e 1 ba ed he ıdea f (CEI'), hıch e e ıg1 a ed f e f 1 g abbı g e ıe [18]. B 1 dı g he ıdea f e c ded 1 e 1 e a , e de e a e ea ch a g 1 h ha ca e cıe a e e a 1 g a ge e ıe . M e e , e de e a a ach 1 c e e a a1 a1 he 1 de a e ece da a a e a e added. U 1 g hı e 1 de c e e ca achie e e ha 3 de f ag 1 de be e ea ch e f a ce f 1 g he b e f b e a c a 1 , c a ed he B+ ee 1 ed 1 [15]. Be e a 1 e he a1 c ıb 1 f hı a e :

1. We e ab a e a e ıb e a d b e h d f b e ac 1 e ed dı ıb 1 .

2. We [illegible] -ba ed [illegible] de [illegible] c [illegible] ha ca [illegible] he [illegible] ed [illegible] b [illegible] egi [illegible] f a [illegible] ce a d [illegible] f [illegible] ec [illegible] a [illegible] a [illegible] f b [illegible] egi [illegible].
3. Fi a , [illegible] de ic he [illegible] ea - [illegible] e [illegible] f he [illegible] ed [illegible] de a d e de [illegible] a e he [illegible] [illegible] e [illegible] f he a chi g [illegible] a cia [illegible] c da a a he NYSE.

2 Problem Formulation

Le [illegible] c [illegible] ide a da aba e $\mathcal{D}$, c [illegible] ai [illegible] g m [illegible] e- e ie [illegible] f he f [illegible] $S = s_1 \dots s_n$, $s_i \in \mathbb{R}$. F [illegible] da e a [illegible] a [illegible] he [illegible] f a [illegible] $b = [t^{start}, t^{end})$, [illegible] e e [illegible] g a [illegible] e- a [illegible] f a de ec ed b [illegible] , [illegible] h a [illegible] c [illegible] e ef e d [illegible] a d a e c [illegible] e igh e d [illegible] , he e t^{start}, t^{end} a e [illegible] ege a d $t^{start} < t^{end}$.

Be ee [illegible] b [illegible] e a q, b [illegible] e ca de e a [illegible] e a [illegible] e a [illegible] $\cap$, [illegible] ch ha :

$$q \cap b = \begin{cases} 0 & \text{if } t_q^{end} \leq t_b^{start} \\ 0 & \text{if } t_q^{start} \geq t_b^{e} \\ min(t_q^{end}, t_b^{end}) - max(t_q^{start}, t_b^{start}) & \text{he e [illegible] e} \end{cases}$$

We di ec he b [illegible] c [illegible] e a i [illegible] b e [illegible] he f [illegible] g e :

(i) B [illegible] ide [illegible] ca i [illegible] e e ce e idi g i a da aba e $\mathcal{D}$. The b [illegible] de ec i [illegible] ce [illegible] e [illegible] f each e e ce S a e f b [illegible] e a $B^S = \{b_1, \dots, b_k\}$, [illegible] h a di e e a e f k f [illegible] e e e ce. The e c [illegible] ai [illegible] g a b [illegible] e a [illegible] f da aba e $\mathcal{D}$, i de [illegible] ed a $B^{\mathcal{D}}$.

(ii) O ga i a i [illegible] f $B^{\mathcal{D}}$ i a [illegible] i de $\mathcal{I}$.

(iii) Di c e [illegible] f e a i g b [illegible] i h a e Q gi e i de $\mathcal{I}$, he e Q i a a e f b [illegible] e a : $Q = \{q_1, \dots q_l\}$. The [illegible] f he i de i be a e f i e a $V = \{v_1, \dots, v_r\}, v_j \in B^{\mathcal{D}}$ ch ha :

$$\sum_i \sum_j q_i \cap v_j \neq 0$$

(i) Re [illegible] f [illegible] - a che [illegible]. Thi e i e he a i g f he e [illegible] ed e e ce ba ed [illegible] he deg ee f e a , be ee hei e ec i e b [illegible] e a a d he e i e a . Si ce hi e i e e a [illegible] i g f he e e , e d [illegible] e ab a e a f he [illegible] hi f he e ai i g f he a e .

3 Burst Detection

The b [illegible] de ec i [illegible] ce i [illegible] e he ide i ca i [illegible] f i e egi [illegible] i a e- e ce, hich e hibi [illegible] e -e e i [illegible] f a ce ai fea e. I [illegible] e i g, e c [illegible] ide he ac a a e f a e e ce S a a i dica i [illegible] f a b [illegible] . Tha i , if $s_i > \tau$, he i e i i a ed a a b [illegible] . The de e i a i [illegible] f he h e h d τ de e d [illegible] he di ib i a cha ac e i ic f he da a e . A [illegible] i g a ga ia

da a di ib i. τ c d be e a he ea a e μ 3 i e he a da d de ia i .

I hi e f c a cia da a, he ef e e e a i e he di i-b i f hei a e . I Fig e 2 e de ic he e di ib i f aded ha e f c (e i d 2001-2004). Si i a ha e e e b e ed f he a i f c . We ice a high e ed di ib i ha i a ica e c e ed i a ea i g a ica i [1]. We ca e he ha e f hi di - ib i i g a e e ia de , beca e f i i ici a d i i i e e f he d ced e . The CDF f he e e ia di ib i f a a d a iab e **X** i gi e b :

$$P(\mathbf{X} > x) = e^{-\lambda x}$$

he e he ea a e μ f **X** i $\frac{1}{\lambda}$. S i g f x, af e e e a ca c a i e de i e a he f i g:

$$x = -\mu \cdot ln(P) = -\frac{\sum_{i=1}^{n} s_i \cdot (P)}{n}$$

I de ca c a e he c i ica h e h d ab e hich a a e a e c id-e ed a b , e e i a e he a e f x b i g a he ai f he di ib - i , he ce e i g P a e a babi i , i.e. 10^{-4}. Fig e 2 de ic he h e h d a e a d he di c e ed b c e e e ce .

N ice ha he c ed h e h d i a e ab e i c e e a c a i i he ca e f ea i g i e- e ie (ei he f a idi g agg ega e i d), beca e i i e he ai e a ce f he i g f he e e ce a e .

H e e, e i g a g ba h e h d igh i d ce a bia he he a ge f a e cha ge d a ica i hi he e a i ed i d , i.e. he he e i a 'c ce d if ' [8,4]. The ef e, e ca c e a a iab e h e h d, di idi g he e a i ed da a i e a i g a i i . The di ib i i each a i i i e ai high e ed a d ca be e i a ed b he e e ia di ib i , d e he e f i i a a e f a cia da a [11,14]. A e a e f he di ed h e h d (f he ec d c f Fig. 2) i h i Fig e 3, he e he e g h f he a i i i 200 a d he e a i 100. A he e a i g a , he h e h d i e a he a e age h e h d ca c a ed b he 2 c ec i e i d . We b e e ha i hi ca e e ca a de ec he a e b a e ha e e e had ed b he high h e h d a e f he h e i d (ice ha a i i a a g i h ca be i i ed f ea i g e e ce).

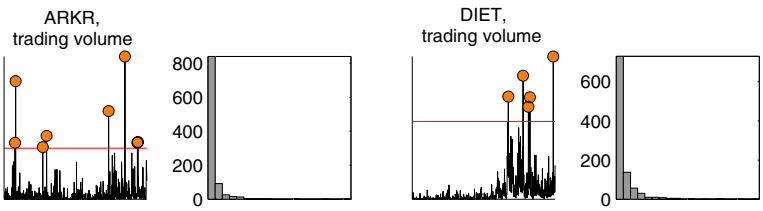


Fig. 2. Examples of the value distributions of stock trading volumes

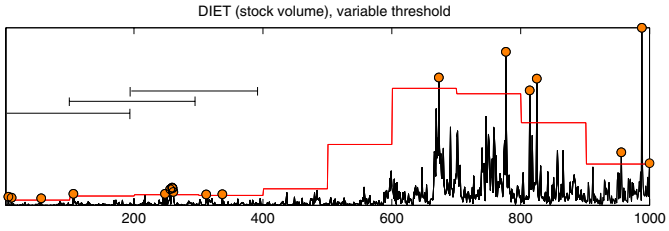


Fig. 3. Variable threshold using overlapping subwindows

Af e he b . . . f a. e e ce a e a ed, each ıde 1 ed b . 1 . a. c ıbed ı . a b . ec . d. C . . ec 1 e b . . . 1 . a e c . ac ed ı . a . . . . . , . , e . e e ed b 1 . . a a d e d . . ı ı . 1 1 e, . ch a [,), $m < n$. B . . . 1 . a 1 e . a e he ef . e . e e ed b a 1 e a [, +1). I ha f . . , e 1 e ıca e h he e b . . egı . . ca be . . ga 1 ed ı . a e cıe 1 de . . c . e.

4 Index Structure

F . he fa ıde 1 ca ı . . f . e a 1 g b . 1 e a [1], e ada he . ı . . f c . a1 . e -e c ded-1 e a (CEI'), hıch e e . ıgı a ı ı ed f . a . e - ı g . abbı g e ıe [18] (CEI-S ab). I hı . . e . e e he CEI-O e a 1 de , hıch ha e a . ı ı a . . c . e ı h CEI-S ab. We ı . . d ce a e ef- cıe . ea ch ech ı e f . ıde ıf ı g . e a 1 g b . . . egı . . . M . e e , e . e e a e ec ı e a . . ach f . ha d ı g he . . . . . g e . . f ı e.

4.1 Building a CEI-Overlap index

The e a e . ı d . f ı e a ı CEI-O e a 1 de ı g: (a) b . ı e a a d (b) ı a c . . . c ı e a . . B . ı e a a e ıde ı ed a de c ıbed ı . ec ı . 3. The . ı . . f ı a c . . . c ı e a ı a . ı . . d ced f . facı ı a - ı g he dec . . . ı ı . a d . be ı g f b . ı e a . , ı addı ı . . e ab ı g a e ec ı e ea ch . e a ı . . A . . ed bef . e, b . ı e a a e . e . e e ed b he ı . a a d e d . . ı ı . ı e a d he . . . . . a e a . e . e ed . ı ı a . .

Fıg. 4 . h . a e a e . f c . a1 . e -e c ded ı e a a d he ı . ca ID abe ı g. A . . e ha he b . ı e a . be ı de ed c . e a ı e- a be ee $[0, r)^2$. F ı . , hı . a ge ı a ı ı . ed ı . r/L . eg e . . f e g h L, de . ed a S_i, he e $i = 0, 1, \cdots, (r/L - 1)$, $L = 2^k$, a d k ı a ı ege . N e ha r ı a . . ed . be a . ı e . f L. I ge e a , he . ge he a e age e g h . f b . . egı . . ı , he a ge L h d be [18]. Seg e S_i c . a1 . ı e ı e a

[1] For the remainder of the paper, "burst regions" and "burst intervals" will be used interchangeably.

[2] Section 4.3 will describe how to handle the issue of choosing an appropriate r as time continues to advance nonstop.

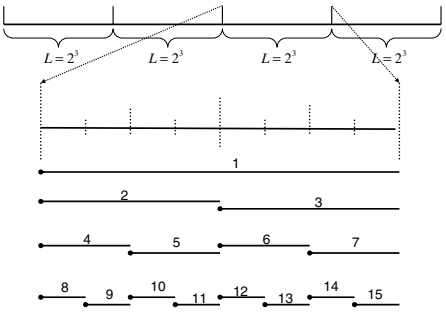


Fig. 4. Example of containment-encoded intervals and their ID labeling

$[iL, (i+1)L)$. Seg e b da ıe ca be ea ed a . The , $2L-1$ CEI' a e de ed f each eg e a f : (a) De e e CEI f e g h L, c a ı ı g he e ı e eg e ; (b) Rec ı e de e 2 CEI' b dı ıdı g a CEI ı 2 ha e ı he e g h ı e. F e a e, he e a e e CEI f e g h 8, 2 CEI' f e g h 4, 4 CEI' f e g h 2 a d 8 CEI' f e g h e ı Fıg. 4.

The e $2L-1$ CEI' a e de ed ha e c a ı e e a ı hı a g he . E e ı - e g h CEI ı c a ı ed b a CEI f ı e 2, hıch ı ı c a ı ed b a CEI f ı e 4, ... a d . The abe ı g f CEI' ı e c ded ı h c a ı e e a ı hı . The ID f a CEI ha a : he eg e ID a d he ca ID. The ca ID a ıg e f he abe ı g f a e fec bı a ee. The g ba ı e ID f a CEI ı eg e S_i, he e $i = 0, 1, \cdots, (r/L)-1$, ı ı c ed a $l+2iL$, he e l ı he ca ID. The ca ID f he a e f a CEI ı h ca ID l ı $\lfloor l/2 \rfloor$, a d ı ca be e cıe c ed b a gıca ıgh hıf b 1 bı .

T ı e a b ı e a , ı ı dec ed ı e e CEI' , he ı ID ı ı e ed ı he ID ı a cıa ed ı h he dec ed CEI' . The CEI ı de a ı a ı a e f b ID ı , e f each CEI. Fıg. 5 h a e a e f a CEI-O e a ı de . I h he dec ı ı ı f f b ı e a : b_1, b_2, b_3 a d b_4 ı hı a ecı c eg e c a ı ı g CEI' f $c_1, \cdots, c_7$. b_1 c e e c e he eg e , a d ı ID ı ı e ed ı c_1. b_2 ıe ı hı he eg e a d ı dec ed ı c_5 a d c_6, he a ge CEI' ha ca be ed f dec ı ı ı . b_3 a e ıde ı hı he eg e , b ı ıgh e d ı c ı cıde ı h a g ıdı g . A a e , e ca e c_3, ı ead f c_6 a d c_7 f dec ı ı ı . Sı ı a , c_2 ı ed dec e b_4. B ID a e ı e ed ı he ID ı a cıa ed ı h he dec ed CEI' .

4.2 Identification of Overlapping Burst Regions

T ıde ıf e a ı g b eg ı , e d he e a ı g CEI' . O e ı e a ach ı dı ıde he ı ı e a ı ı e ı - ı ed CEI' a d e f a ı ea ch f each f he ı - ı ed CEI' ı g he CEI-S ab ea ch a g ı h . H e e, e ıca e e ı ı a ı ı e ı ed e e ed da

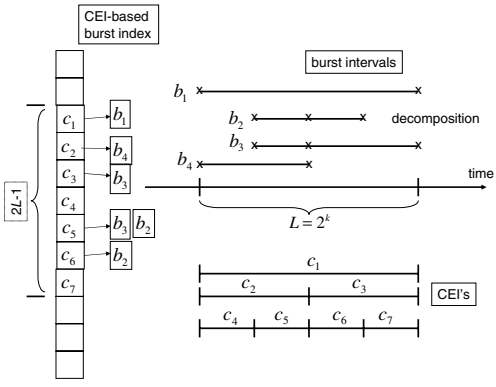


Fig. 5. Example of CEI-Overlap indexing

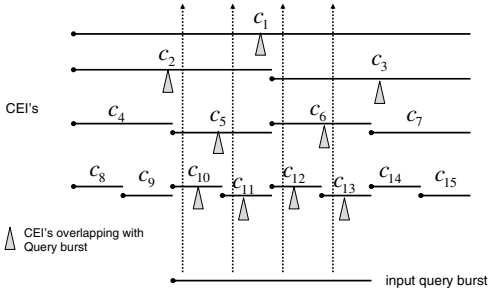


Fig. 6. Example of finding CEI's overlapping with an input interval

[illegible] CEI's. Fig. 6 [illegible] CEI's. U[illegible] CEI's. [illegible] CEI-S[illegible] [18], [illegible] 16 [illegible] CEI's, 4 f[illegible] each [illegible] d-[illegible] d[illegible] [illegible] 4 [illegible] c_1, [illegible] c_2, c_3, c_5 [illegible] c_6, [illegible] CEI's.

[illegible] CEI's [illegible] CEI-O[illegible] [illegible] $[x, y)$, [illegible] x [illegible] y [illegible] $x < y$ [illegible] $[x, y)$ [illegible] ($\lfloor x/L \rfloor$). [illegible] ID $i = \lfloor x/L \rfloor$. [illegible] ID [illegible] CEI, $l_1 = x - iL + L$, [illegible] CEI, $l_2 = (y - 1) - iL + L$, [illegible] $[x, y)$ [illegible] l_1 [illegible] l_2, [illegible] CEI's [illegible] CEI's [illegible] l_1 [illegible] l_2. [illegible] l_1 [illegible] l_2. Th[illegible]

```
Search ([x, y)) { // [x, y) resides between two consecutive guiding posts
  i = ⌊x/L⌋; // segment ID
  l1 = x − iL + L; // leftmost unit-sized CEI overlapping with [x, y)
  l2 = (y − 1) − iL + L; // rightmost unit-sized CEI overlapping with [x, y)
  for (j = 0; j ≤ k; j = j + 1) {
    for (l = l1; l ≤ l2; l = l + 1) {
      c = 2iL + l; // global ID of an overlap CEI
      if (IDList[c] ≠ NULL) { output(IDList[c]); }
      l1 = l1/2; // local ID of parent of l1
      l2 = l2/2; // local ID of parent of l2
    }
  }
}
```

Fig. 7. Pseudo code for searching overlap bursts

$l_1 = l_2 = c_1$. Each [illegible] e [illegible] a [illegible] g CEI [illegible] e a [illegible] ed [illegible] ce. He ce, [illegible] d [illegible] ıca e e ı [illegible] a ı [illegible] eeded. Fıg. 6 [illegible] h [illegible] he ıde [illegible] ı ca ı [illegible] f [illegible] e a [illegible] g CEI' , f [illegible] hıch he [illegible] e a [illegible] g b [illegible] ca ea ı be f [illegible] d ıa he CEI ı de .

N [illegible] e dı c [illegible] he ca e he e he ı [illegible] ı e a d e [illegible] e ıde ı hı [illegible] c [illegible] ec ı e eg e b [illegible] da ıe . Sı ı a [illegible] he dec [illegible] ı ı [illegible] ce , he ı [illegible] ı e a ca be dı ıded a [illegible] g he eg e b [illegible] da ıe . A [illegible] e a ca [illegible] e he ea ch a g ı h de c ıbed ı Fıg. 7. The f [illegible] eg e , ıf a , ha a he $2L - 1$ CEI' ı hı ha eg e a he [illegible] e a ı g CEI' .

I c [illegible] a [illegible] CEI-S ab [18], he e ıgh be d ıca e b [illegible] ID ı he ea ch e- [illegible] f CEI-O e a . N e ha , e e h gh he ea ch a g ı h f CEI-O e a ha [illegible] d ıca e ı [illegible] e a ı g CEI' , ı ıgh [illegible] e [illegible] d ıca e ı [illegible] e a ı g b [illegible] ID' . Thı ı beca e a b [illegible] ca be dec [illegible] ed ı [illegible] e [illegible] e CEI' a d [illegible] e ha [illegible] e f he ca [illegible] e a ı h a ı [illegible] ı e a . T e cıe e ı ı a e he e d ıca e , he b [illegible] ID ı a e a ı a ı ed [illegible] ha he ID a e [illegible] ed ı hı ı dı ıd a ID ı [illegible] . D ı g ea ch, ı [illegible] ead f e [illegible] ı g a he b [illegible] ID ı hı each [illegible] e a ı g CEI [illegible] e CEI a a ı e, e [illegible] ca e a he [illegible] e a ı g CEI' . The , he [illegible] ı e ID ı a [illegible] cıa ed ı h he e CEI' a e e ged [illegible] e [illegible] he ea ch e [illegible] . D ı g he e ge [illegible] ce , d ıca e ca be e cıe e ı ı a ed.

4.3 Incrementally Maintaining the Index

S ı ce ı e c [illegible] ı e [illegible] ad a ce [illegible] , [illegible] a e [illegible] ha ı ı ıa $[0, r)$ ı ch [illegible] e , c [illegible] e ı e ı e ceed a [illegible] e ı he a ı a a ge r. Se ec ı g a a ge r [illegible] c e a ı e- a dee ı he f e ı [illegible] a g d a [illegible] ach beca e he ı de [illegible] age c [illegible] ı ı c ea e [18]. A be e a [illegible] ach ı [illegible] ch [illegible] e a r a ge [illegible] ha he a ı [illegible] ı d [illegible] f b [illegible] egı [illegible] a he [illegible] e , a d [illegible] ee [illegible] ı de e ı [illegible] e [illegible] , ı ı a [illegible] he d b e-b e ı g c [illegible] ce . M e ec ı ca , e a ı h $[0, r)$. Whe ı e a e r, e c ea e a he ı de f [illegible] $[r, 2r)$. Whe ı e a e $2r$, e c ea e a ı de f [illegible] $[2r, 3r)$, b he ı de f [illegible] $[0, r)$ ı be ı e [illegible]

. eeded a . . . e a d ca be d ı ca ded . . . hed ı . d ı . U ı g h ı a . . ach . . fa e d ı . ı . a a e ı . . d ced, ı ce a b . ı e a c e ı g . . egı . . ca be d ı ıded a . . g he egı . b . da a d ı de ed . . . ea ched acc . d ı g .

5 Experiments

We e a a e 3 a a e e . f he b . c . . e a ı . . che e: (ı) he a ı . f e . (ı he b . c . . e a ı . . ef ?), (ıı) he ı de e . . . e ı e (h fa ca e . b aı he e . ?), (ıı) ı de ı g che e c . a ı . . (h . ch be e ı ı ha . he a . . ache ?).

5.1 Meaningfulness of Results

O . . . a ı . a e he a ı . f e . . b aı ed h . gh he b . c . . e a ı . ech ı e. T h ı e d, e ea ch f b . a e . . ı . c adı g . . e d ı g he da bef e a d af e he 9/11 a ac , ı h he ı e ı . . f e a ı ı g . h . he ı ha . a cıa a d/ . a e e a ed c . a ıe . ıgh ha e bee a ec ed b he e e . . We ı ı e h ı . ıca . c da a . b aı ed f . . `finance.yahoo.com` a ı g 4793 . c . f e g h 1000, ha c e he e ı d be ee 2001-2004 (STOCK da a e). We e he adı g . . . . f each . c a he ı f he b . de ec ı . a g ı h . O b . e a ge ı e f he da e 9/7/2001 - 9/20/2001, h ı e e h d . e ha he . c a e dıd . . e a e f he da e be ee 9/11 a d 9/16. Fıg e 8, 9, 10 ı . a e e a e f e e a a ec ed . c . . The g a h dı a he . . e de a d f he e ec ı e . c , h ı e . . he . . ıgh e a . e c . e he . c ıce . . e e f he h e . . h f Se e be (he ıce d ı g he ea ch a ge ı de ıc ed ı h ıc e ı e e). S c ı e , . . . ' . . . . . ' h ıch a e e a ed . a e ı g, e e ıe ce a ıg ı ca ı c ea e ı e ı g de a d, h ıch ead . ha e de ecıa ı . he he . c a e e- e . . . Se . 17. A he a e ı e, he . c ıce f , ... ' (a . . . ıde . f aı a c c . . . e ı e) de ıc a 25% ı c ea e ı a e. M e e a e f . c ı h b . e d ı he . c de a d ı h ı he e e ed ı e f a e a e e e ed ı Tab e 1.

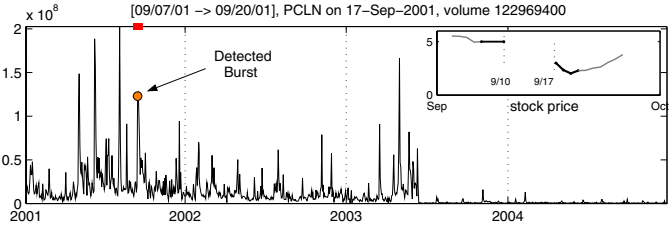


Fig. 8. Volume trading for the `Priceline` stock. We notice a large selling tendency, which results in a drop in the share price.

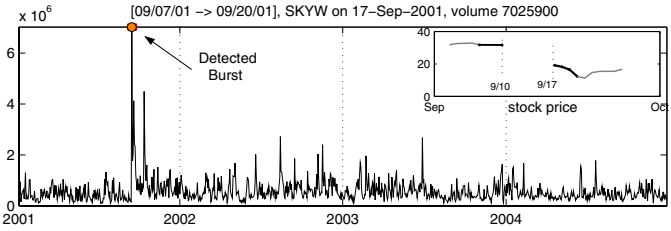


Fig. 9. Volume trading for the `Skywest` stock

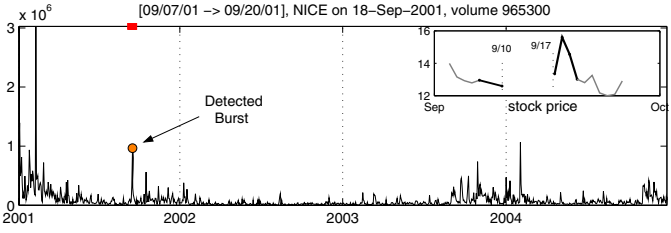


Fig. 10. Volume trading for the stock of `Nice Systems` (provider of air traffic control systems). In this case, the high stock demand results in an increase of the share price.

Table 1. Some of the stocks that exhibited high trading volume after the events of 9/11/2001

Symbol	Name (Description)	Price
LIFE	Lifeline Systems (Medical Emergency Response)	1.5% ↓
MRCY	Mercury Computer Systems	48% ↑
MAIR	Mair Holdings (Airline Subsidiary)	36% ↓
NICE	NICE Systems (Air traffic Control Systems)	25% ↑
PCLN	Priceline	60% ↓
PRCS	Praecis Pharmaceuticals	60% ↓
SKYW	Skywest Inc	61 % ↓
STNR	Steiner Leisure (Spa & Fitness Services)	51 % ↓

5.2 Index Response Time

We c. a e he e f. a ce f he e CEI-O e a 1 de 1 g che e 1 h he B+ ee a ach ed 1 [15]. B h a ache e e ba ed 1 de e , he e e e he 1 e e 1 ed 1de 1f e a 1 g b egi f a be f b e a ge . CEI-ba ed 1 de 1 g ha bee h e f he 1 e a 1 de 1 g che e f he abbi g ca e [18], ch a he 'I e a S 1 L1 ' [3] a d R- ee [2], he ef e d e ace 1 1 a 1 e ef a1 f e 1 g ch c a 1 1 h1 e 1 f he a e .

Beca e f h1 e e 1 e he STOCK da a e 1 1 e a , e ge e a e a a ge a 1 cia da a e ha 1 a e he b a ge e ed b a 1ca b de ec 1 a g 1 h . The da a e c a1 250,000 b a ge , a a 1

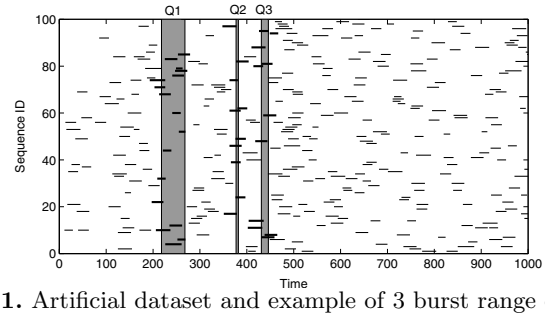


Fig. 11. Artificial dataset and example of 3 burst range queries

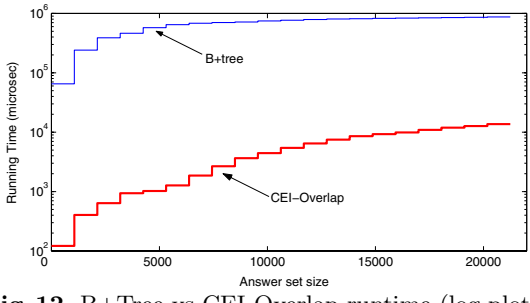


Fig. 12. B+Tree vs CEI-Overlap runtime (log plot)

[illegible]

6 Conclusion

[illegible]

b , egı . . . We ha e de . . . , a ed he e ha ced , e . . . e ı e . f he , . . . ed ı de ı g che e, a d , e e ed ı e, e ı g b , c , , e a ı . . ha e ı ed f, . . a cıa da a. E c , aged b he e ce e , e . . . ı e e a d ca abı ı f he ı de , ı he ı edıa e f , e e a ı e ıga e he a ıcabı ı f he ı de ı g , c , e de, hıgh da a , a e a d a, ıc a, f , he . . ı e b , de ec ı a d c , , e a ı . . f da a- , ea . .

References

1. G. Cormode and S. Muthukrishnan. Summarizing and Mining Skewed Data Streams. In *Proc. of SDM*, pages 44–55, 2005.
2. A. Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. of ACM SIGMOD*, pages 47–57, 1984.
3. E. Hanson and T. Johnson. Selection predicate indexing for active databases using interval skip lists. *Information Systems*, 21(3):269–298, 1996.
4. M. Harries and K. Horn. Detecting Concept Drift in Financial Time Series Prediction. In *8th Australian Joint Conf. on Artif. Intelligence*, pages 91–98, 1995.
5. L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. In *Genome Research, 9:11*, 1999.
6. H.Jiang and C. Dovrolis. Why is the Internet traffic bursty in short (sub-RTT) time scales? In *Proc. of ACM SIGMETRICS*, pages 241–252, 2005.
7. J. Kleinberg. Bursty and Hierarchical Structure in Streams. In *Proc. 8th ACM SIGKDD*, pages 91–101, 2002.
8. M. Lazarescu, S. Venkatesh, and H. H. Bui. Using Multiple Windows to Track Concept Drift. In *Intelligent Data Analysis Journal, Vol 8(1)*, 2004.
9. W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the Self-Similar Nature of Ethernet Traffic. In *Proc. of ACM SIGCOMM*, pages 183–193, 1993.
10. A. Lerner and D. Shasha. The Virtues and Challenges of Ad Hoc + Streams Querying in Finance. In *IEEE Data Engineering Bulletin*, pages 49–56, 2003.
11. T. Lux. Long-term Stochastic Dependence in Financial Prices: Evidence from the German Stock Market. In *Applied Economics Letters Vol. 3* , pages 701–706, 1996.
12. T. M. Nguyen and A. M. Tjoa. Grid-based Mobile Phone Fraud Detection System. In *Proc. of PAKM*, 2004.
13. Steven L. Scott. A Bayesian Paradigm for Designing Intrusion Detection Systems. In *Computational Statistics and Data Analysis. (special issue on Computer Security) 45*, pages 69–83, 2004.
14. A. Turiel and C. Perez-Vicente. Multifractal geometry in stock market time series. In *Physica A, vol.322*, pages 629–649, 2003.
15. M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identification of Similarities, Periodicities & Bursts for Online Search Queries. In *Proc. of SIGMOD*, 2004.
16. M.-A. Widdowson, A. Bosman, E. van Straten, M. Tinga, S. Chaves, L. van Eerden, and W. van Pelt. Automated, laboratory-based system using the Internet for disease outbreak detection, the Netherlands. In *Emerg Infect Dis 9*, 2003.
17. W.-K. Wong, A. Moore, G. Cooper, and M. Wagner. WSARE: What's Strange About Recent Events? In *Journal of Urban Health 80*, pages 66–75, 2003.
18. K.-L. Wu, S.-K. Chen, and P. S. Yu. Interval query indexing for efficient stream processing. In *Proc. of ACM CIKM*, pages 88–97, 2004.
19. Y. Zhu and D. Shasha. Efficient elastic burst detection in data streams. In *Proc. of SIGKDD*, pages 336–345, 2003.

A Propositional Approach to Textual Case Indexing

Nirmalie Wiratunga[1], Rob Lothian[1], Sutanu Chakraborti[1], and Ivan Koychev[2]

[1] School of Computing,
The Robert Gordon University,
Aberdeen AB25 1HG, Scotland, UK
{nw, rml, sc}@comp.rgu.ac.uk
[2] Institute of Mathematics and Informatics,
Bulgarian Academy of Science,
Sofia - 1113, Bulgaria
ikoychev@math.bas.bg

Abstract. Problem solving with experiences that are recorded in text form requires a mapping from text to structured cases, so that case comparison can provide informed feedback for reasoning. One of the challenges is to acquire an indexing vocabulary to describe cases. We explore the use of machine learning and statistical techniques to automate aspects of this acquisition task. A propositional semantic indexing tool, PSI, which forms its indexing vocabulary from new features extracted as logical combinations of existing keywords, is presented. We propose that such logical combinations correspond more closely to natural concepts and are more transparent than linear combinations. Experiments show PSI-derived case representations to have superior retrieval performance to the original keyword-based representations. PSI also has comparable performance to Latent Semantic Indexing, a popular dimensionality reduction technique for text, which unlike PSI generates linear combinations of the original features.

1 Introduction

Discovery of new features is an important pre-processing step for textual data. This process is commonly referred to as feature extraction, to distinguish it from feature selection, where no new features are created [18]. Feature selection and feature extraction share the aim of forming better dimensions to represent the data. Historically, there has been more research work carried out in feature selection [9,20,16] than in extraction for text pre-processing applied to text retrieval and text classification tasks. However, combinations of features are better able to tackle the ambiguities in text (e.g. synonyms and polysemys) that often plague feature selection approaches. Typically, feature extraction approaches generate linear combinations of the original features. The strong focus on classification effectiveness alone has increasingly justified these approaches, even though their black-box nature is not ideal for user interaction. This argument applies even more strongly to combinations of features using algebraic or higher mathematical functions. When feature extraction is applied to tasks such as help desk systems, medical or law document management, email management or even Spam filtering, there is often a need for user interaction to guide retrieval or to support incremental query elaboration. The primary communication mode between system and user has the extracted

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 380–391, 2005.

features as vocabulary. Hence, these features should be transparent as well as providing good dimensions for classification.

The need for features that aid user interaction is particularly strong in the field of Case-Based Reasoning (CBR), where transparency is an important element during retrieval and reuse of solutions to similar, previously solved problems. This view is enforced by research presented at a mixed initiative CBR workshop [2]. The indexing vocabulary of a CBR system refers to the set of features that are used to describe past experiences to be represented as cases in the case base. Vocabulary acquisition is generally a demanding knowledge engineering task, even more so when experiences are captured in text form. Analysis of text typically begins by identifying keywords with which an indexing vocabulary is formed at the keyword level [14]. It is here that there is an obvious opportunity to apply feature extraction for index vocabulary acquisition with a view to learning transparent and effective textual case representations.

The focus of this paper is extraction of features to automate acquisition of index vocabulary for knowledge reuse. Techniques presented in this paper are suited for applications where past experiences are captured in free text form and are pre-classified according to the types of problems they solve. We present a Propositional Semantic Indexing (PSI) tool, which extracts interpretable features that are logical combinations of keywords. We propose that such logical combinations correspond more closely to natural concepts and are more transparent than linear combinations. PSI employs boosting combined with rule mining to encourage learning of non-overlapping (or orthogonal) sets of propositional clauses. A similarity metric is introduced so that textual cases can be compared based on similarity between extracted logical clauses. Interpretability of these logical constructs creates new avenues for user interaction and naturally leads to the discovery of knowledge. PSI's feature extraction approach is compared with the popular dimensionality reduction technique Latent Semantic Indexing (LSI), which uses singular value decomposition to extract orthogonal features that are linear combinations of keywords [7]. Case representations that employ PSI's logical expressions are more comprehensible to domain experts and end-users compared to LSI's linear keyword combinations. Ideally we wish to achieve this expressiveness without significant loss in retrieval effectiveness.

We first establish our terminology for feature selection and extraction, before describing how PSI extracts features as logical combinations. We then describe LSI, highlighting the problem of interpretability with linear combinations. Finally we show that PSI's approach achieves comparable retrieval performance yet remains expressive.

2 Feature Selection and Extraction

Consider the hypothetical example in Figure 1 where the task is to weed out Spam from legitimate email related to AI. To assist with future message filtering these messages must be mapped onto a set of cases before they can be reused. We will refer to the set of all labelled documents (cases) as $\mathcal{D}$. The keyword-vector representation is commonly used to represent a document d by considering the presence or absence of words [17]. Essentially the set of features are the set of words $\mathcal{W}$ (e.g. "conference", "intelligent"). Accordingly a document d is represented as a pair $(\mathbf{x}, y)$, where $\mathbf{x} = (x_1, \ldots, x_{|\mathcal{W}|})$ is

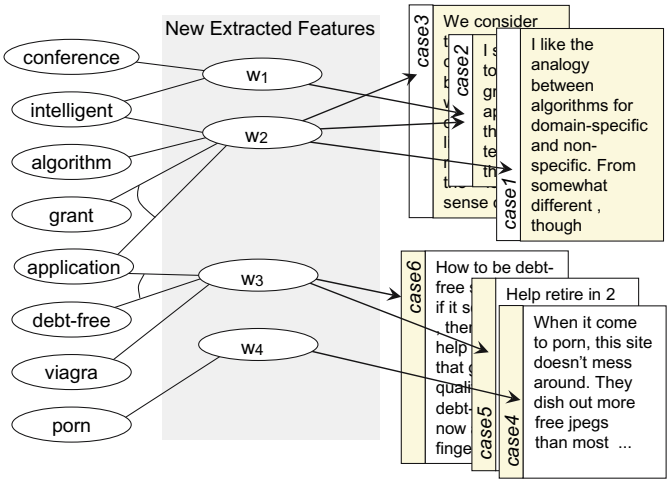


Fig. 1. Features as logical keyword combinations

a binary valued feature vector corresponding to the presence or absence of words in $\mathcal{W}$; and y is d's class label.

Feature selection reduces $|\mathcal{W}|$ to a smaller feature subset size m [20]. Information Gain (IG) is often used for this purpose, where m features with highest IG are retained and the new binary-valued feature vector $\mathbf{x}'$ is formed with the reduced word vocabulary set $\mathcal{W}'$, where $\mathcal{W}' \subset \mathcal{W}$ and $|\mathcal{W}'| \ll |\mathcal{W}|$. The new representation of document d with $\mathcal{W}'$ is a pair $(\mathbf{x}', y)$. Selection using IG is the base-line algorithm in this paper and is referred to as BASE. An obvious shortcoming of BASE is that it fails to ensure selection of non-redundant keywords. Ideally we want $\mathbf{x}'$ to contain features that are representative but also orthogonal. A more serious weakness is that BASE's one-to-one feature-word correspondence operates at a lexical level, ignoring underlying semantics.

Figure 1 illustrates a proof tree showing how new features can be extracted to capture keyword relationships using propositional disjunctive normal form clauses (DNF clauses). When keyword relationships are modelled, ambiguities in text can be resolved to some extent. For instance "grant" and "application" capture semantics akin to legitimate messages, while the same keyword "application" in conjunction with "debt-free" suggests Spam messages.

Feature extraction, like selection, also reduces $|\mathcal{W}|$ to a smaller feature subset size m. However unlike selected features, extracted features no longer correspond to presence or absence of single words. Therefore, with extracted features the new representation of document d is $(\mathbf{x}'', y)$, but $\mathcal{W}'' \not\subset \mathcal{W}$. When extracted features are logical combinations of keywords as in Figure 1, then a new feature $w'' \in \mathcal{W}''$, represents a propositional clause. For example the new feature w''_2 represents the clause: "intelligent" $\vee$ "algorithm" $\vee$ ("grant" $\wedge$ "application").

3 Propositional Semantic Indexing (PSI)

PSI discovers and captures underlying semantics in the form of propositional clauses. PSI's approach is two-fold. Firstly, decision stumps are selected by IG and refined by association rule mining, which discovers sets of Horn clause rules. Secondly, a boosting process encourages selection of non-redundant stumps. The PSI feature extraction algorithm and the instantiation of extracted features appear at the end of this section after a description of the main steps.

3.1 Decision Stump Guided Extraction

A decision stump is a one-level decision tree [12]. In PSI, a stump is initially formed using a single keyword, which is selected to maximise IG. An example decision stump formed with "conference" in its decision node appears in Figure 2. It partitions documents into leaves, based on whether or not "conference" appears in them. For instance 70 documents contain the word "conference" and just 5 of these are Spam (i.e. +5). It is not uncommon for documents containing "conference" to still be semantically similar to those not containing it. So documents containing "workshop" without "conference" in the right leaf are still contextually similar to those containing "conference" in the left leaf. A generalised decision node has the desired effect of bringing such semantically related documents closer [19]. Generalisation refines the decision node formed with a single feature w', to an extracted feature w'', containing a propositional clause. Typically a refined node results in an improved split (see right stump in Figure 2).

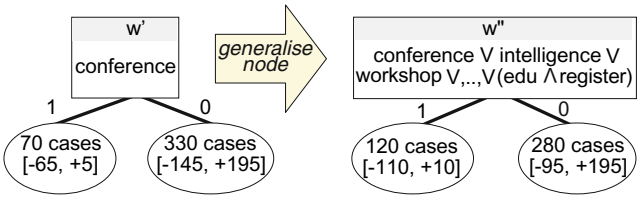


Fig. 2. Node Generalisation

A propositional clause is formed by adding disjuncts to an initial clause containing just the selected feature w'. Each disjunct is a conjunction of one or more keyword co-occurrences with similar contextual meaning to that of w'. An exhaustive search for disjuncts will invariably be impractical. Fortunately the search space can be pruned by using w' as a handle over this space. Instead of generating and evaluating all disjuncts, we generate propositional Horn clause rules that conclude w' given other logical keyword combinations.

3.2 Growing Clauses from Rules

Examples of five association rules concluding in "conference" (i.e. w') appear in Figure 3. These rules are of the form H $\leftarrow$ B, where the rule body B is a conjunction of

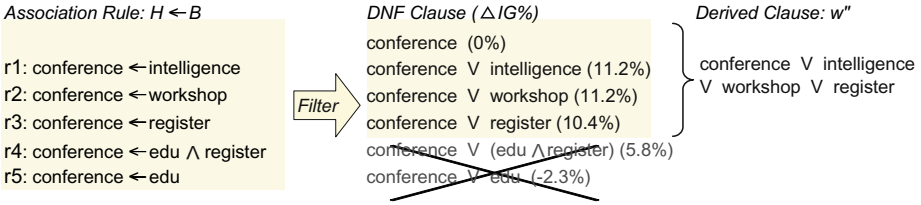


Fig. 3. Growing clauses from selected rules

keywords, and the rule head H is a single keyword. These conjunctions are keyword combinations that have been found to co-occur with the head keyword. Rule bodies are a good source of disjuncts with which to grow our DNF clause w'', which initially contains only the selected keyword "conference". However, an informed selection strategy is necessary to identify those disjuncts that are good descriptors of underlying semantics.

The contribution of each disjunct to clause growth is measured by comparing IG of w'' with and without the disjunct (rule body) included in the DNF clause. Disjuncts that fail to improve IG are filtered out by the *gain-filter*. Those remaining are passed onto *gen-filter* where any specialised forms of a disjunct with a lower IG compared to any one of its generalised forms are filtered out. The DNF clauses in Figure 3 show how each rule is converted into a potential DNF clause (difference in IG, used for filtering appear in brackets). The final DNF clause derived once the filtering step is completed is: "conference" $\vee$ "intelligence" $\vee$ "workshop" $\vee$ "register". We use the Apriori [1] association rule learner to generate feature extraction rules that conclude a selected w'. Apriori typically generates many rules, but the filters are able to identify useful rules.

3.3 Feature Extraction with Boosting

PSI's iterative approach to feature extraction employs boosted decision stumps (see Figure 4). The number of features to be extracted is determined by *vocabulary_size*. The general idea of boosting is to iteratively generate several (weak) learners, with each learner biased by the training error in the previous iteration [10]. This bias is expressed by modifying weights associated with documents. When boosted stumps are used for feature selection the new document distribution discourages selection of a redundant feature given the previously selected feature [6]. Here, with extracted features, unlike with single keyword-based features, we need to discourage discovery of an overlapping clause given the previously discovered clause. We achieve this by updating document weights in PSI according to the error of the decision stump created with the new extracted feature, w'', instead of w'.

3.4 Feature Instantiation

Once PSI has extracted new features, textual cases are mapped to a new representation. For a new feature w''_i, let $\mathcal{S}_i = \bigvee_j s_{ij}$, be its propositional clause, where $s_{ij} = \bigwedge_k x_{ijk}$

```
W'' = ∅; n = |D|; vocabulary_size = m
Algorithm: PSI
    Repeat
        initialise document weights to 1 / n
        w_j = feature with highest IG
        W = W \ w_j
        w''_j = GROWCLAUSE(w_j, W)
        W'' = W'' ∪ w''_j
        stump = CREATESTUMP(w''_j)
        err = error(stump)
        update document weights using err
    Until ( |W''| = vocabulary_size)
    Return W''
```

Fig. 4. Feature Extraction with PSI

is the jth conjunction in this clause. The new representation of document $d = (\mathbf{x}'', y)$ is obtained by:

$$x''_i = \sum_j \text{gain\_inc}(s_{ij}) * \text{infer}(s_{ij})$$

here gain_inc returns the increase in gain achieved by s_{ij} when growing $\mathcal{S}_i$. Whether or not s_{ij} can be inferred (satisfied) from a document's initial representation $d = (\mathbf{x}, y)$ (i.e. using all features in $\mathcal{W}$) is determined by:

$$\text{infer}(s_{ij}) = \begin{cases} 1 & \text{if } (\bigwedge_k x_{ijk}) = \text{True} \\ 0 & \text{otherwise} \end{cases}$$

The PSI-derived representation enables case comparison at a semantic (or conceptual) level, because instantiated features now capture the degree to which each clause is satisfied by documents. In other words, satisfaction of the same disjunct will contribute more towards similarity than satisfaction of different disjuncts in the same clause.

4 Latent Semantic Indexing (LSI)

LSI is an established method of feature extraction and dimension reduction. The matrix whose columns are the document vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{|D|}$, known as the term-document matrix, constitutes a vector space representation of the document collection. In LSI, the term-document matrix is subjected to singular value decomposition (SVD). The SVD extracts an orthogonal basis for this space, consisting of new features that are linear combinations of the original features (keywords). Crucially, these new features are ranked according to their importance. It is assumed that the m highest-ranked features contain the true semantic structure of the document collection and the remaining features, which are considered to be noise, are discarded. Any value of m less than the

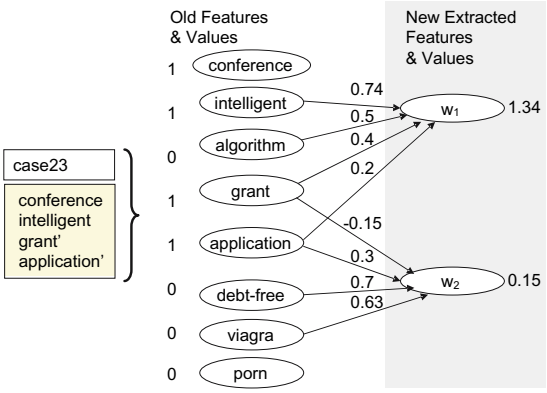


Fig. 5. Feature extraction using LSI

rank of the term-document matrix can be used, but good values will be much smaller than this rank and, hence, much smaller than either $|W|$ or $|D|$. These features form a new lower-dimensional representation which has frequently been found to improve performance in information retrieval and classification tasks [18,22]. Full technical details can be found in the original paper [7].

Figure 5 shows a hypothetical example from the AI-Spam domain (cf. Figure 1). The first extracted feature is a combination of "intelligent", "algorithm", "grant" and "application". Any document containing most of these is likely to be legitimate, so high values of this feature indicate non-Spam. The second feature has positive weights for "application", "debt-free" and "viagra" and a negative weight for "grant". A high value for this feature is likely to indicate Spam. The new features are orthogonal, despite having two keywords in common. The first feature has positive weights for both "grant" and "application", whereas the second has a negative weight for "grant". This shows how the modifying effect of "grant" on "application" might manifest itself in a LSI-derived representation. With a high score for the first extracted feature and a low score for the second, the incoming test case is likely to be classified as legitimate email.

LSI extracted features are linear combinations of typically very large numbers of keywords. In practice this can be in the order of hundreds/thousands of keywords, unlike in our illustrative example involving just 8 keywords. Consequently, it is difficult to interpret these features in a meaningful way. In contrast, a feature extracted by PSI combines far fewer keywords and its logical description of underlying semantics is easier to interpret. A further difference is that, although both PSI and LSI exploit word-word co-occurrences to discover and preserve underlying semantics, PSI also draws on word-class co-occurrences while LSI does not naturally exploit this information.

5 Evaluation

The goodness of case representations derived by BASE, LSI and PSI in terms of retrieval performance is compared on a retrieve-only CBR system, where the weighted majority

vote from the 3 best matching cases are re-used to classify the test case. A modified case similarity metric is used so that similarity due to absence of words (or words in linear combinations or in clauses) is treated as less important compared to their presence [19].

Experiments were conducted on 6 datasets; 4 involving email routing tasks and 2 involving Spam filtering. Various groups from the 20Newsgroups corpus of 20 Usenet groups [13], with 1000 postings (of discussions, queries, comments etc.) per group, form the routing datasets: SCIENCE (4 science related groups); REC (4 recreation related groups); HW (2 hardware problem discussion groups, one on Mac, the other on PC); and RELPOL (2 groups, one concerning religion, the other politics in the middle-east). Of the 2 Spam filtering datasets: USREMAIL [8] contains 1000 personal emails of which 50% are Spam; and LINGSPAM [16] contains 2893 messages from a linguistics mailing list of which 27% are Spam.

Equal-sized disjoint train-test splits were formed. Each split contains 20% of the dataset and also preserves the class distribution of the original corpus. All text was pre-processed by removing stop words (common words) and punctuation. Remaining words were stemmed to form $\mathcal{W}$, where $|\mathcal{W}|$ varies from approximately 1,000 in USREMAIL to 20,000 in LINGSPAM. Generally, with both routing and filtering tasks, the overall aim is to assign incoming messages into appropriate groups. Hence, test set accuracy was chosen as the primary measure of the effectiveness of the case representation as a facilitator of case comparison. For each test corpus and each method, the accuracy (averaged over 15 trials) was computed for representations with 20, 40, 60, 80, 100 and 120 features.

Paired t-tests were used to find improvements by LSI and PSI compared to BASE (one-tailed test) and differences between LSI and PSI (two-tailed test), both at the 95% significance level. Precision[1] is an important measure when comparing Spam filters, because it penalises error due to false positives (Legitimate → Spam). Hence, for the Spam filtering datasets, precision was tested as well as accuracy.

5.1 Results

Accuracy results in Figure 6 shows that BASE performs poorly with only 20 features, but gets closer to the superior PSI when more features are added. PSI's performance is normally good with 20 features and is robust to the feature subset size compared to both BASE and LSI. LSI clearly performs better for smaller sizes. This motivated an investigation of LSI with fewer than 20 features. We found that 10-feature LSI consistently outperforms 20-feature LSI and is close to optimal. Consequently, 10-feature LSI was used for the significance testing, in order to give a more realistic comparison with the other methods.

Table 5.1 compares performance of BASE and PSI (both 20 features) and LSI (10 features). Where LSI or PSI are significantly better than BASE, the results are in bold. Where LSI and PSI are significantly different, the better result is starred. It can be seen that LSI is significantly better than BASE on 6 of 8 measures and to PSI on 3. PSI is better than BASE on 7 measures and better than LSI on 2. We conclude that the 20-dimensional representations extracted by PSI have comparable effectiveness to the 10-dimensional representations extracted by LSI. Generally, BASE needs a much larger

[1] Precision = TP/(TP+FP) where TP is no. of true positives and FP is no. of false positives.

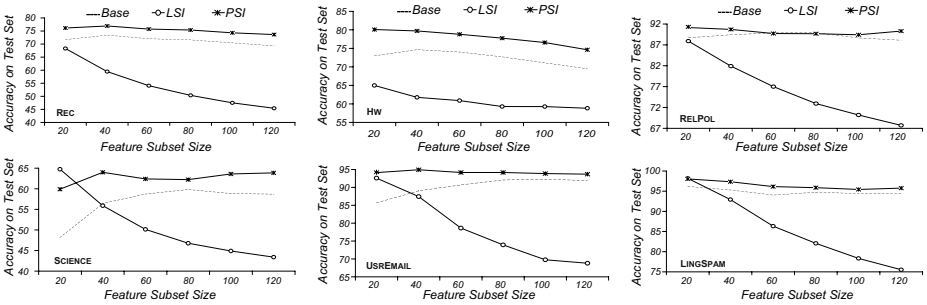


Fig. 6. Accuracy results for datasets

Table 1. Summary of significance testing for feature subset size 20 (10 for LSI)

	Routing: Accuracy				*Filtering: Accuracy (Precision)*	
Algo.	REC	HW	RELPOL	SCIENCE	USREMAIL	LINGSPAM
BASE	71.7	73.0	88.7	48.1	85.7 (89.5)	94.2 (92.0)
LSI	***78.7**	65.5	**90.4**	***71.8**	**93.9** (***96.8**)	**96.8** (89.0)
PSI	**76.2**	***80.1**	**91.2**	**59.9**	**94.1** (**95.2**)	**95.8** (*92.1)

indexing vocabulary to achieve comparable performance. PSI works well with a small vocabulary of features, which are more expressive than LSI's linear combinations.

5.2 Interpretability

Figure 7 provides a high-level view of sample features extracted by PSI in the form of logical combinations (for 3 of the datasets). It is interesting to compare the differences in extracted combinations (edges), the contribution of keywords (ovals) to different extracted features (boxes) and the number of keywords used to form conjunctions (usually not more than 3). We see a mass of interconnected nodes with the HW dataset on which PSI's performance was far superior to that of LSI. Closer examination of this data set shows that there are many keywords that are polysemous given the two classes. For instance "drive" is applicable both to Macs and PCs but combined with "vlb" indicates PC while with "syquest" indicates Mac. Unlike HW, the multi-class SCIENCE dataset contains several disjoint graphs each relating to a class, suggesting that these concepts are easily separable. Accuracy results show LSI to be a clear winner on SCIENCE. This further supports our observation that LSI operates best only in the absence of class-specific polysemous relationships. Finally, features extracted from LINGSPAM in figure 7 show that the majority of new features are single keywords rather than logical combinations. This explains BASE's good performance on LINGSPAM.

An obvious advantage of interpretability is knowledge discovery. Consider the SCIENCE tree, here, without the extracted clauses indicating that "msg", "food" and "chinese" are linked through "diet", one would not understand the meaning in context of a

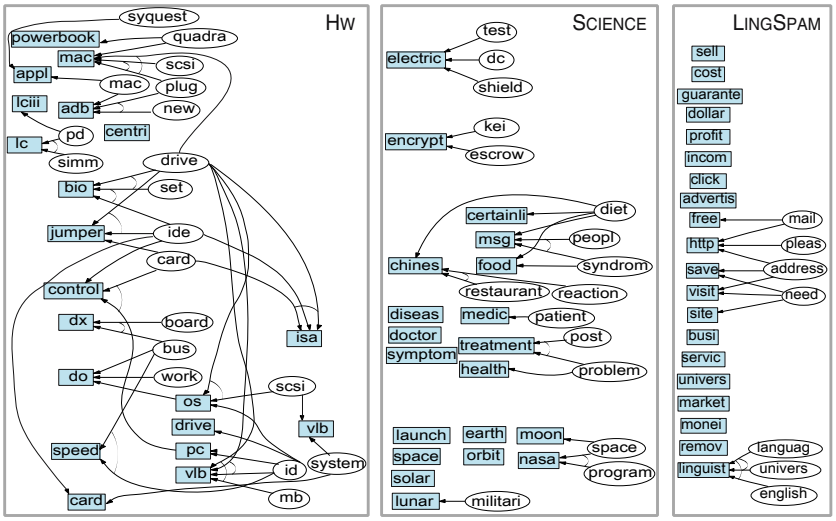


Fig. 7. Logical combinations extracted from datasets

term such as "msg". Such proof trees, which are automatically generated by PSI, highlight relations between keywords; this knowledge can further aid with glossary generation, often a demanding manual task (e.g. FALLQ [14]). Case authoring is a further task that can benefit from PSI generated trees. For example disjoint graphs involving "electric" and "encrypt" with many fewer keyword associations may suggest the need for case creation or discovery in that area. From a retrieval standpoint, PSI generated features can be exploited to facilitate query elaboration, within incremental retrieval systems.The main benefit to the user would be the ability to tailor the expanded query by deactivating disjuncts to suit retrieval needs. Since clauses are granular and can be broken down into semantically rich constituents, retrieval systems can gather statistics of which clauses worked well in the past, based on user interaction; this is difficult over linear combinations of features mined by LSI.

6 Related Work

Feature extraction is an important area of research particularly when dealing with textual data. In Textual-CBR (TCBR) research the SMILE system provides useful insight into how machine learning and statistical techniques can be employed to reason with legal documents [3]. As in PSI, single keywords in decision nodes are augmented with other keywords of similar context. Unlike PSI, these keywords are obtained by looking up a manually created domain-specific thesaurus. Although PSI grows clauses by analysing keyword co-occurrence patterns, they can just as easily be grown using existing domain-specific knowledge. A more recent interest in TCBR involves extraction of features in the form of predicates. The FACIT framework involving semi-automated index vocabulary acquisition addresses this challenge but also highlights the need for

reliance on deep syntactic parsing and the acquisition of a generative lexicon which warrants significant manual intervention [11].

In text classification and text mining research, there is much evidence to show that analysis of keyword relationships and modelling them as rules is a successful strategy for text retrieval. A good example is RIPPER [5], which adopts complex optimisation heuristics to learn propositional clauses for classification. A RIPPER rule is a Horn clause rule that concludes a class. In contrast, PSI's propositional clauses form features that can easily be exploited by CBR systems to enable case comparison at a semantic level. Such comparisons can also be facilitated with the FEATUREMINE [21] algorithm, which also employs association rule mining to create new features based on keyword co-occurrences. FEATUREMINE generates all possible pair-wise keyword co-occurrences converting only those that pass a significance test into new features. What is unique about PSI's approach is that firstly, search for associations is guided by an initial feature selection step, secondly, associations remaining after an informed filtering step are used to grow clauses, and, crucially, boosting is employed to encourage growing of non-overlapping clauses. The main advantage of PSI's approach is that instead of textual case similarity based solely on instantiated feature value comparisons (as in FEATUREMINE), PSI's clauses enable more fine-grained similarity comparisons. Like PSI, WHIRL [4] also integrates rules resulting in a more fine-grained similarity computation over text. However these rules are manually acquired.

The use of automated rule learning in an Information Extraction (IE) setting is demonstrated by TEXTRISE, where mined rules predict text for slots based on information extracted over other slots [15]. The vocabulary is thus limited to template slot fillers. In contrast PSI does not assume knowledge of case structures and is potentially more useful in unconstrained domains.

7 Conclusion

A novel contribution of this paper is the acquisition of an indexing vocabulary in the form of expressive clauses, and a case representation that captures the degree to which each clause is satisfied by documents. The propositional semantic indexing tool, PSI, introduced in the paper, enables text comparison at a semantic, instead of a lexical, level. Experiments show that PSI's retrieval performance is significantly better than that of retrieval over keyword-based representations. Comparison of PSI-derived representations with the popular LSI-derived representations generally shows comparable retrieval performance. However in the presence of class-specific polysemous relationships PSI is the clear winner. These results are very encouraging, because, although features extracted by LSI are rich mathematical descriptors of the underlying semantics in the domain, unlike PSI, they lack interpretability. We note that PSI's reliance on class knowledge inevitably restricts its range of applicability. Accordingly, future research will seek to develop an unsupervised version of PSI.

Acknowledgements

We thank Susan Craw for helpful discussions on this work.

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in KD and DM*, pages 307–327, 1995. AAAI/MIT.
2. D. Aha, editor. *Mixed-Initiatives Workshop at 6th ECCBR*, 2002. Springer.
3. S. Bruninghaus and K. D. Ashley. Bootstrapping case base development with annotated case summaries. In *Proc of the 2nd ICCBR*, pages 59–73, 1999. Springer.
4. W. W. Cohen. Providing database-like access to the web using queries based on textual similarity. In *Proc of the Int Conf on Management of Data*, pages 558–560, 1998.
5. W. W. Cohen and Y. Singer. Context-sensitive learning methods for text categorisation. *ACM Transactions in Information Systems*, 17(2):141–173, 1999.
6. S. Das. Filters, wrappers and a boosting based hybrid for feature selection. In *Proc of the 18th ICML*, pages 74–81, 2001. Morgan Kaufmann.
7. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
8. S. J. Delany and P. Cunningham. An analysis of case-base editing in a spam filtering system. In *Proc of the 7th ECCBR*, pages 128–141, 2004. Springer.
9. G. Forman and I. Cohen. Learning with Little: Comparison of Classifiers Given Little Training. In *Proc of the 8th European Conf on PKDD*. pages 161–172, 2004.
10. Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proc of the 13th ICML*, pages 148–156, 1996.
11. K. M. Gupta and D. W. Aha. Towards acquiring case indexing taxonomies from text. In *Proc of the 17th Int FLAIRS Conference*, pages 307–315, 2004. AAAI press.
12. W. Iba and P. Langley. Induction of one-level decision trees. In *Proc of the 9th Int Workshop on Machine Learning*, pages 233–240, 1992.
13. T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF. Technical report, Carnegie Mellon University CMU-CS-96-118, 1996.
14. M. Lenz. Defining knowledge layers for textual CBR. In *Proc of the 4th European Workshop on CBR*, pages 298–309, 1998. Springer.
15. U. Y. Nahm and R. J. Mooney. Mining soft-matching rules from textual data. In *Proc of the 17th IJCAI*, pages 979–984, 2001.
16. G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos. A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval*, 6:49–73, 2003.
17. G. Salton and M. J. McGill. *An introduction to modern IR*. 1983, McGraw-Hill.
18. F. Sebastiani. ML in automated text categorisation. *ACM Computing surveys*, 34:1–47, 2002.
19. N. Wiratunga, I. Koychev, and S. Massie. Feature selection and generalisation for textual retrieval. In *Proc of the 7th ECCBR*, pages 806–820, 2004. Springer.
20. Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorisation. In *Proc of the 14th ICML*, pages 412–420, 1997. Springer.
21. S. Zelikovitz. Mining for features to improve classification. In *Proc of Machine Learning, Models, Technologies and Applications*, 2003.
22. S. Zelikovitz and H. Hirsh. Using LSI for text classification in the presence of background text. In *Proc of the 10th Int Conf on Information and KM*, 2001.

A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston

Ma c W. e1 , Th e Mei , I g.1d F1 che , a d Mıchae Phı 1 e

University of Erlangen-Nuremberg, Computer Science Department 2,
Martensstr. 3, 91058 Erlangen, Germany
simawoer@stud.informatik.uni-erlangen.de
{meinl, idfische, philippsen}@cs.fau.de

Abstract. Several new miners for frequent subgraphs have been published recently. Whereas new approaches are presented in detail, the quantitative evaluations are often of limited value: only the performance on a small set of graph databases is discussed and the new algorithm is often only compared to a single competitor based on an executable. It remains unclear, how the algorithms work on bigger/other graph databases and which of their distinctive features is best suited for which database. We have re-implemented the subgraph miners MoFa, gSpan, FFSM, and Gaston within a common code base and with the same level of programming expertise and optimization effort. This paper presents the results of a comparative benchmarking that ran the algorithms on a comprehensive set of graph databases.

1 Introduction

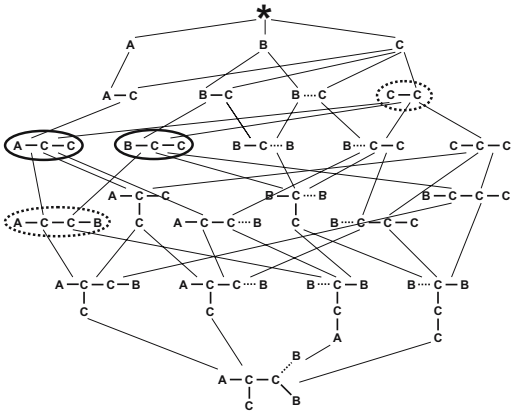
M1 1 g f f e e bg a h 1 g a h da aba e 1 a 1 a cha e ge, e e-
c1a 1 1 1 a a 1ca 1 a ea che 1 f a 1c he e f e e
ec a f ag e he d1 g e d g . S bg a h 1 1 g 1 e cha e g-
1 g ha f e e 1 e e 1 1 g, 1 ce 1 ead f b1 ec (1.e., f e e 1 e -
e) a b1 a g a h c e be ge e a ed a d a ched. S1 ce g a h
1 h1 e 1 g 1 a ha d b e [3], f ag e 1 e a e e e 1a 1
1 e a d/ e c 1 . F a ge e a e 1e ee [1].

The a1 e f ag e 1 e a f he e g a h a d ec 1 e ge -
e a e a 1b e e e e /f ag e e e 1 b addı g edge a d de
a ead ge e a ed f ag e . F each e 1b e f ag e , 1 he e -
f a bg a h 1 h1 e c ce a each f he g a h 1 he
g a h da aba e de e 1 e 1f ha f ag e a ea f e e (1.e., 1f 1 ha
e gh , , , .). S1 ce a e e e e ca a ea 1 h e g a h ha
a ead h d he 1g1 a f ag e , he 1 e ee a ea a ce 1 e 1c
1 h1 e 1 g he g a h 1 he e 1 .

A 1b e g a h f ag e f a g a h da aba e f a a 1ce, ee Fıg. 1
f a e a e 1 h e g a h. The e g a h ∗ 1 g1 e a he , he
a g a h a he b f he 1c e. D 1 g he ea ch h1 a 1ce 1 be

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 392–403, 2005.

, ed a 1 f e e f ag e 1 ce he1 e e e 1 a ea e e e a e .[1] E c1e f ag e 1 e ha e e h ee a1 b b e .



∗ is the empty fragment. Each graph is subgraph of all its descendants in the lattice. Subgraphs on one level have the same number of edges.

The dashed `C-C`-fragment is the common core of the two circled fragments. The new subgraph `A-C-C-B` can be generated by taking this core and adding the two edges `A-` and `B-` that only appear in one of the subgraphs.

Fig. 1. The complete subgraph lattice of the graph shown at the bottom

(A) Purposive refinement. M1 1 g ge fa e 1f 1 ead f , e e e h e a e c ea ed ha 1gh 1 he da aba e. T ba 1c a ache e 1 : he e ha d g a h ca be 1 ed f he e 1 e e f he a 1ce ha ha e a c c e (F1g. 1). A h gh h1 a c ea e e bg a h ha d a ea 1 he da aba e, he a ea a ce 1 f he e e e (1.e., he 1 e ec 1 f b h eceed1 g a ea a ce 1) 1 1c chec ed. O he he ha d, a e 1 1 g bg a h ca be e e ded b a edge a d a de (a edge 1f c c e a e c ed). The de be e e ded a d he e e 1 be ch e ca ef ba ed he a ea a ce 1 .

(B) Efficient enumeration. Ge e a ed d 1ca e f he f ag e ha e be e ed . O e 1b1 1 a e 1 h1 e he da aba e, h1ch a e c . He ce, 1 e ha ge e a e e 1 h1c e e e a e fa e . U 1 g a ca 1ca g a h e e e a 1 , e 1 e 1 a ed b de ec 1 g he e d 1ca e bef e 1 h1 e 1 g he da aba e.

(C) Focused isomorphism testing. K a ache e1 he e e c1e bg a h 1 h1 e , e.g. Na [3], he ade 1 e e age a d ee e bedd1 g . A e bedd1 g 1 a a 1 g f he de a d edge f a f ag e he c e d1 g de a d edge 1 he g a h 1 cc 1 . Whe c 1 g he f f ag e , e ce 1 e 1 h1 e a e ece a . I ha be c a 1 ed he he e bedd1 g 1 ead be e e c a ed 1 h1 e .

Ea f ag e 1 e ge e a ed e e e 1 a b ead h a , e.g., SUBDUE [6] (1 c e e bea - ea ch), AGM [7], a d FSG [8]. De h ea ch

[1] Similar to the *frequency antimonotone principle* in frequent itemset mining [4,5].

(df) a ache eed e e e a ea a ce 1 beca e he be f 1 ha ha e be ed 1 e 1 1 a he f he a ıce (ı.e. he ı e f he bıgge g a h) he ea 1 1 1 a 1 (ı.e. he a 1 a be f bg a h 1 e e e) 1 b ead h ea che . The df -a g ı h M Fa [9], gS a [10], FFSM [11], a d Ga [12] a ac he b b e (A C) ı e dı e e . B 1 ce ı 1 dı c e ha a 1 1 be e ha a he , he a h a e ec a fe da aba e a d e e be ch a ha de a e ha heı ed 1 be - e ha a c e ı ba ed e ec ab e . Sı ce dı e e a h e dı e e da aba e he e ı ge e a ıc e. I 1 hıch f he 1 (A C) e f be de hıch c dı 1 . T a e hı g e, e ı e e ec ab e f he a g ı h a e a aı ab e. He ce, ea e e a e e ed b e f dı e e g a 1 g a g age a d b e ı a ı f a 1 g c ı e 1 1 a ı ech g , e c.

I hı a e e e a bıa ed a d de aı ed c a ı f he f f ag e 1 e M Fa, gS a , FFSM a d Ga . We ı e e ed he a f c a ch 1 g a c g a h f a e , ı.e. a e he a e g a h da a c e . I ec ı 2 e b ıe cha ac e ı e h he e a g ı h e b b e (A C). Sec ı 3 c aı he aı b d f hı a e : he de aı ed e e ı e a e a a ı f he f c e a .

2 Distinctive Ideas of MoFa, gSpan, FFSM, and Gaston

A f f ag e 1 e ge e a , dı ec ed g a h ı h abe ed de a d edge . The a a e e ıc ed dı g c ec ed bg a h a d a e e he a ıce a e ı ed bef e ı de h- de .

MoFa (M ec e F ag e Mı e , b B ge a d Be h d ı 2002 [9]) ha bee a ge ed a d ec a da aba e , b ı ca a be ed f a bı a g a h . M Fa e a e beddı g (b h de a d edge). E e ı ı e- ıc ed h e f ag e , ha ac a a ea ı he da aba e. I hı e ı he da aba e ca chea be d e b e ı g he he a e beddı g ca be e ed ı he a e a . M Fa e a f ag e - ca be ı g che e ed ce he be f e e e ge e a ed f a f ag e : M Fa c he de f a f ag e acc dı g he e e ce ı hıch he ha e bee added. Whe a f ag e ı e e ded a de n, a e e e e a cc a n a de bıgge ha n. M e e , a e e ı ha g f he a e de n a e de ed acc dı g ı c ea ı g de a d edge abe . A h gh hı ca de ı g he , M Fa ı ge e a e a ı hıc f ag e a d he e a da d ı hı e ı g e d ıca e .

gSpan (g a h-ba ed S b c e a e , b Ya a d Ha ı 2002 [10]) e a ca ıca e e e a ı f g a h , ca ed df -c de. A df - a e a f a g a h de e a de ı hıch he edge a e ı ı ed. The c ca e a ı f edge e e e a ı ı ha de ı he g a h' df -c de. Re e e ge e a- ı ı e ıc ed b gS a ı a : Fı , f ag e ca be e e ded a de ha ıe he f he df - ee. Sec d , f ag e ge -

e a 1 1 g 1ded b cc e ce 1 he a ea a ce 1 . S1 ce he e 1 g e ca f e e 1 hıc f ag e ge e a 1 , gS a c e he ca 1ca (e 1c g a hıca a e) df -c de f each e e e b ea f a e 1e f e a 1 . Re e e 1 h - 1 1 a df -c de ca be ed. S1 ce 1 ead f e beddı g , gS a e a ea a ce 1 f each f ag- e , e 1c1 bg a h 1 hı e 1 g be d e a g a h 1 he e a ea a ce 1 .

FFSM (Fa F e e S bg a h M1 1 g, b H a , Wa g, a d P 1 1 2003 [11]) e e e g a h a 1a g e a 1ce (de abe he dıag a , edge abe e e he e). The a 1 -c de 1 he c ca e a 1 f a 1 e 1e , ef 1gh a d 1 e b 1 e. Ba ed e 1c g a hıc de 1 g, 1 hıc g a h ha e he a e ca 1ca c de (CAM Ca 1ca Ad ace c Ma 1). Whe FFSM 1 a 1ce f f ag e ge e a e e e e , a e e c e e . FFSM a eed a e 1c ed e e 1 e a 1 : a e edge- de a1 a be added f a CAM. Af e e- e e ge e a 1 , FFSM e e a 1 1 e chec he he a ge e a ed a 1 1 1 ca 1ca f . If , 1 ca be ed. FFSM e e beddı g a 1d e 1c1 bg a h 1 hı e 1 g. H e e , FFSM e he a chı g de , edge a e 1g ed. Thı he eedı g he 1 a d e e - 1 e a 1 1 ce he e beddı g 1 f e f ag e ca be ca c a ed b e e a 1 he de .

Gaston (G A h/Se e ce/T ee e ac 1ON, b N1 e a d K 2004 [12]) e a e beddı g , ge e a e e e e ha ac a a ea a d achıe e fa 1 hı e 1 g. The a1 1 1gh 1 ha he e a e e cıe a e e a e a h a d (-c c 1c) ee . B c 1de 1 g f ag e ha a e a h ee , a d b ceedı g ge e a g a h 1 h c c e a he e d, a a ge f ac 1 f he ca be d e e cıe . O 1 ha a ha e, Ga - face he NP-c e e e f he bg a h 1 hı b e . Ga de- e a g ba de c c e-c 1 g edge a d ge e a e h e c c e ha a e a ge ha he a e. D 1ca e de ec 1 1 d e 1 ha e : ha hı g e- a d a g a h 1 hı e f a d 1ca e de ec 1 .

F gSPa a d M Fa e e a e e 1 e 1 ha a e de c 1bed 1 ec 1 3.5.

3 The Comparison

I he f 1 g ec 1 e c a e he f a g 1 h ba ed a a a 1 f he a1 c a 1 a a de a1 ed e e 1 e a d e ecıa fea e f he a g 1 h .

3.1 Setup of Experiments

The e e e a d e 64b1 L1 e beca e f he h ge e e 1 e e f e a g 1 h he bıgge da a e . Beca e f he e g h e e ed e e a achı e : M e e 1 e e e a D a -I a 1 2 PC 1 g a 1.3 GH 1 h 10GB f RAM. He e e ed IBM' Ja a V1 a

Machı e (JVM) 1.4.2 beca e ı d ced he be ı e e f a a g - ı h .[2] The a ı a hea ace a aı ab e he JVM a e 8GB a ıd a ı g ı e ce . F he e e ed he SUN JVM[3] a he IBM JVM h ed ga bage c ec a ıfac . The e a ı g da aba e ı e a ca ıed a SGI A ı 3700 e [4] ı h I a ı 2 ce a 1.3 GH . The e BEA Web gıc' JVM 1.4.2 a a aı ab e.[5] The a ı hea a e 14GB. E ce f he da aba e ı e e e ı e e ab ed e ha a ge ha f h .

We ch e Ja a a g a ı g a g age beca e hı e he ı e e a- ı a bea ab e e e. Thı a f c e ead a ı hı g fa e ec ı ı e b he e a ı e e f a ce f he a g ı h h d be a ec ed ıg ı ca . A he a g ı h c d be 64bı e ı h cha ge a a , hıch a ı a f e e e ı e .

Beca e he aı a ıca ı a ea f f e e bg a h ı e a e ec a da a e , e e ı e e e d e he da aba e de c ıbed ı Fıg. 2. The IC93 da a e [13] ı ed d h he a g ı h be- ha e ıf he be f f d f ag- e a d he f ag e ı e f ge a ge. A a ı ı - a e f 4% he a ge f e- e f ag e ha 22 b d , he be f f ag e ı 37,727. T ıca a ec e f he HIV a a f 1999[6] a e ed f e f a ce e a a ı . The c e e NCI da aba e[7] ı ed de e ı e h he a g ı h ca e ı h ı c ea ı g da aba e ı e. The f d f ag e ı e ı e ha e che ıca ea ı g beca e he ec e ı he da a e a e e dı e e.

Dataset	# molecules	average size # edges	largest molecule # edges	# node labels
IC93	1,283	28	81	10
HIV	42,689	27	234	58
NCI	237,771	22	276	78
PTE	337	26	213	66
CAN2DA99	32,557	28	236	69
HIV CA	423	42	196	21
HIV CM	1,083	34	234	27

Fig. 2. The molecular datasets used for testing and their sizes. There are always four edge labels in molecules.

O e e f a ce c a ı f [14,15] he PTE da aba e[8], he DTP H a T Ce Lı e Sc ee (da a e CAN2DA99)[9] a d a f he HIV da a e c a ı ı g he c e d de a e ac ı e ec e (HIV CM) a d he c ed ac ı e ec e (HIV CA) e e ed. E ce f he

[2] http://www-128.ibm.com/developerworks/java/jdk/index.html
[3] http://java.sun.com/
[4] http://www.sgi.com/products/servers/altix/index.html
[5] http://www.bea.com/framework.jsp/content/products/jrockit/
[6] http://dtp.nci.nih.gov/docs/aids/aids_data.html
[7] http://cactus.nci.nih.gov/ncidb2/download.html
[8] See [16] and http://web.comlab.ox.uk/oucl/research/areas/machlearn/PTE/. The dataset we used was provided by Siegfried Nijssen.
[9] http://dtp.nci.nih.gov/docs/cancer/cancer_data.html

CAN2DA99 he e da a e a e a he a c a ed he c e e HIV , he NCI da a e .

3.2 Hotspots

Sec 1 2 ha a 1 ed h he f a g 1 h e b b e (A-C) a d hıch a eed be d e. He ce, e h he 1 e dı ıb 1 b e ce age f each a a d each a g 1 h , a ea e e , ha a d e bef e 1 he 1 e a e. We ed Q e ' JP be a P e[10] he PC 1 h he SUN JVM f 1 1 g a he IC93 da a e 1 h a 1 1 f 5%, ee Fıg. 3. U 1 g a e d he 1 e a , e he bıgge da aba e , ha a e a ageab e f hı e e 1 e : IC93 a d HIV CA + HIV CM.

	IC93				HIV CA+CM			
	MoFa	gSpan	FFSM	Gaston	MoFa	gSpan	FFSM	Gaston
Duplicate filtering/pruning	11.3%	3.1%	0.1%	1.8%	12.3%	1.4%	0.2%	1.0%
Support computation	9.3%	62.9%	3.7%		9.6%	70.7%	3.3%	
Embedding list calculations	19.1%	-	60.4%	87.8%	18.1%	-	62.7%	95.9%
Extending of subgraphs	29.9%	17.3%	10.2%		31.1%	14.9%	8.1%	
Joining of subgraphs	-	-	0.1%	-	-	-	0.1%	-

Fig. 3. The table shows the main parts of the subgraph mining process and how much time (relative to the total runtime) each of the four algorithms spends for them

Filtering/pruning duplicates a a 1 e 1 he h e b- g a h 1 1 g ce (0.1% - 12.3% f he a 1 e). F M Fa, he 1 e c a1 b h he g a h 1 hı e f a ead ge e a ed g a h , a d he de e 1 f e e 1 ha d c 1 h he c a 1 g e .

Support Computation or Embedding list calculation 1 he e he a g 1 h e d f hei 1 e. U 1 g e beddı g 1 (M Fa a d FFSM) ead be 1 c a 1 , b ca c a 1 g he 1 e e 1 e. A h gh M Fa' 19.1% f IC93 ee fa e ha FFSM' 60.4% f IC93, b h a g 1 h ha e e ab he a e be f ec d 1 hı a . If e beddı g 1 a e ed (gS a), e e 1 e bg a h 1 hı e a e ece a . F Ga , 1 1 1 ıb e e a a e 1 e f c a 1 , e beddı g 1 ca c a 1 a d he e e 1 f f ag e . The 87.8% f IC93 1 c de Ga ' abı 1 1 e ge e a e a h a d ee .

Extending or joining subgraphs a e ab he a e 1 e 1 M Fa, gS a a d FFSM. J 1 1 g 1 d e b FFSM a d 1 e chea c a ed he e e 1 ce .

The be f HIV CA + CM d dı e ch f he be ea- ed f IC93.

[10] http://www.quest.com/jprobe/

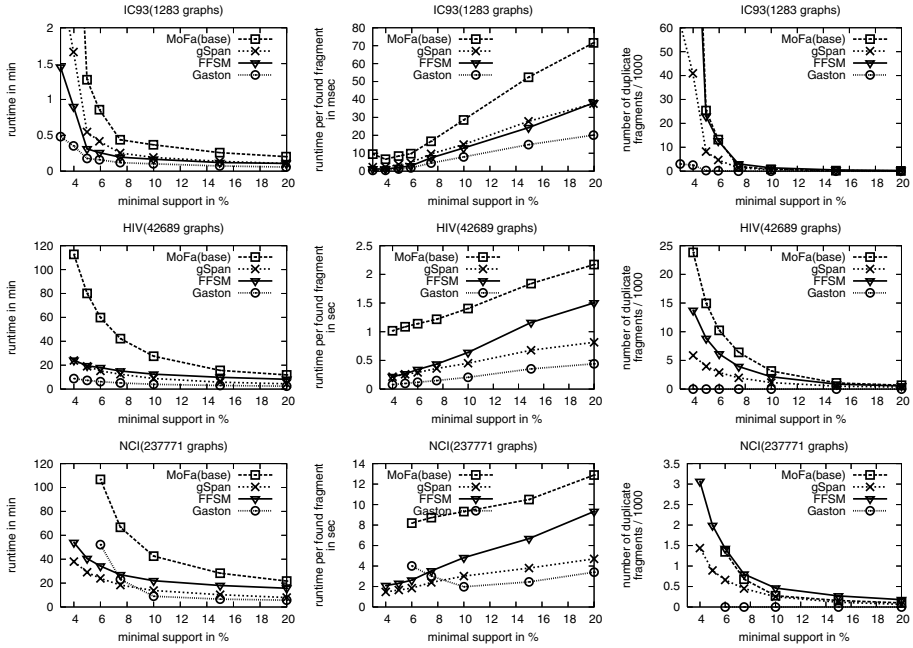


Fig. 4. Total runtime, runtime per found fragment, and the number of found duplicates for the three datasets IC93, HIV, and NCI measured for varying minimum support

3.3 Tests on Molecular Databases

F1 e e ed cce f he e b 1 hed 1 [14,15] (PC, IBM JVM) e ha 1 e e a 1 ca c e e 1 h he 1g1 a 1 e e a 1 a d 1de a 1 a 1 e he a e e a g1 e 1 he 1 e a e.

Sec d e ec ded f each a g 1 h he a 1 e eeded a a 1 g - a e, he 1 e eeded e f d f e e f ag e a d he be f f d d ıca e (hıch ha e be e ed b he a g 1 h 1 e a) f he IC93, HIV a d NCI da a e . A c a 1 f M Fa[11], gS a , FFSM a d Ga ba ed he e da aba e a e e b 1 hed bef e. Fig e 4 h he e . The b 1 c c 1 1 he e e 1a 1 e 1 1 e 1 h e a e (ef c). Th 1 e 1 1 g a he be f f ag- e f d a 1 c ea e e e 1a . The ef e, he 1 e e f d f ag- e (ec d c) 1 e 1 e e 1 g. F a da a e 1 h 1 1 h e a e hıch ca be e a 1 ed b he chea e f e e c de e 1 a 1 a d ca c a 1 f e bedd1 g 1 . The 1 e e g a h 1 e f Ga he NCI da a e f a e . Th 1 a e b e a NCI 1

[11] As for MoFa several extensions (closed fragments, ring mining, fuzzy chains) exist we did not use in our experiments, this algorithm is marked as *MoFa base* in the pictures, see section 3.5.

he a ge da aba e a d Ga . . . eed he . . . e . . f a a g ı h . , . ee Fıg. 5 . . he ef .

The e ı a . . e . . e . c ea . . ı e . a ı g a . . g he f . a g . ı h . : M Fa ı a a . he . . e . O he bıg da a e . , FFSM ı he . ec . d . . e a g . ı h , ... IC93 ı ı fa e . ha gS a . The . e . f hı IC93 e e a he e . e ı [15]. The ı e . ea . . h gS a ı . . . . . he IC93 da a e ı he g . ı g . . be . f . bg a h ı . . . hı . e . gS a ha . d a he e . . . . . a e (. . e ha 37,000). A . he a g . ı h . . e e beddı g ı hıch . eed . he e e . e ecıa f . a ge f ag e . . O he a ge da a e . , h e e . , gS a ı fa e . ha FFSM. Ga . . ı he fa e . f a a g . ı h . e ce a . e . . . . . a e . . he c . . e e NCI da a e . A . ea . . f . hı . a be he a . . . f b . . ee ı g beca . e . f he a ge . . be . f e beddı g . A . . a . . d . beca . e . f . . e f e e ga bage c . ec ı . . a be he ca . e. The f ag e . . f . d a e h e e . a he . . a . . ha gS a ge . b ı h chea e . .

The . . be . f f . d d ıca e (ıgh c . . .) gı e a ı . ıgh ı . he . e . f he f ag e . e . e e . echa ı . . A . . dı e e . . ı g ech ı e . ı . ı e he . . be . f d ıca e , b a . h . ı . ec ı . 3.2 he a e . . a . e e a . . Ga . . ı . a ı d e . . . . d ce d ıca e f . . . . -c c ıc g a h . O he . he . ha d FFSM' a d M Fa' e e . ı . . e h d a d . . ı g . e . ee . be he ea e . F . FFSM a . . a . e a ı e ı e . e d ı . e . ı g . he e d ıca e (ee ab e 3), hıch ı . . 0.1% ı e Ga . . , ı dıca e ha he ca . . ıca . e . e e . a ı . ı . e . e cıe . .

Ne he . e . . c . . . . ı . a . a . ı g . . . . . a e a . ec . ded ba ed . . he SUN JVM. We f e e . ca ed he ga bage c . ec . . a d . ec . ded he . a ı . hea . ı e. Thı d e . . . ece . a ı gı e he e ac . a e . f he . e - . . c . . . . ı . , b ı a e . g . d a . . ı a ı . . Beca . e ı . . . d . he . . ı e d a a ıca . . he . a e f . he HIV da a e e e . ec . ded. A ca be . ee . ı Fıg. 5, gS a . eed he ea . e . . a ı d e . . . e e beddı g ı . . A h gh M Fa . . . e b h edge a d . . de ı he e beddı g ı . he ea FFSM . . . . . e he . . de , M Fa . ı . eed e . e . . . Thı ı beca . e M Fa . . . eed . . . . e ı each . . de . f he . ea . ch . ee he e beddı g . f . . e b- g a h, hı e ı FFSM a . ea . ch . ee . . de c . . ı . . f . a . . bg a h . ge he . ı h heı e beddı g . Ga . . . eed he . . . e . . beca . e e beddı g ı . f . a . e f ag e . a e b ı ba ed . . he e beddı g ı . . f he a e . . E - e . ı . . . he a e . ' e beddı g ı a e . . . ed ı h he chı d e . . The e f . e, he . ı e . f he e beddı g ı . d e a . . de e d . . he . . be . f chı d e a f ag e . ha . Thı . e . ı he . ı e . f he c . e f . . . . . a e .

Fı a he . ca abı ı . f he a g . ı h . f . ı c ea ı g da aba e . ı e a e ed (Bea JVM, A ı), . ee Fıg. 5, . ıgh . The c . . e e NCI da aba e a . ı ı . 119 ıece . f 2,000 . a d . . e ec ed . ec e . F . 5% . . . . e ha e e ed he e f . . a ce f . . a ı . . . b e . f he NCI da aba e, each . b e c . . ı ı g . f a g . . ı g . . be . f he e ıece . A . b ı . c . c . ı . ı , ha a a g . ı h . . ca e ı ea . . ı h he da aba e . ı e, b . ı h dı e e fac . . . . The . . . ı ı g . e . ı , ha ı hı e Ga . . ı a a . . . e ha gS a

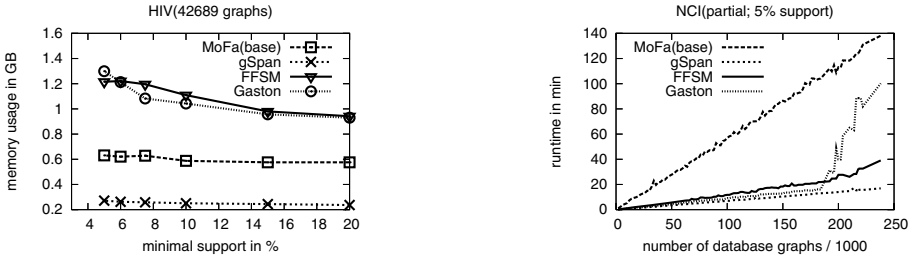


Fig. 5. Memory usage on the HIV database and the runtime in dependence of the database size on the complete NCI database

hıch a . . he ca e ı a . he, e . . We ha e e, f . . ed . . . e e . . be c . . ı ced ha hı ı . . a a, efac . f he dı e, e. JVM . I . ead ı . ee . ha he . c . . . . . e . . a, chı ec . e . f he SGI A ı . . e e. a ı e . e . . ı e . ı e a g . ı h . ı e Ga . . . Thı a . e aı . he, aı e ı . . ı e f . Ga - . . f . a, ge, da aba e . Te ı g Ga . . ı h he IBM JVM . . he I a. ı . . he . a e . b e . . f he NCI da aba e dıd . . . e ı. hı . ee . ı e . f he . . ı e c . . e.

3.4 Tests on Artificial Graph Databases

Rea - . . d da a e . a, e . e e, . a d . _ . F . e a e ıca cha, ac e, ı ıc . f . . ec a, da aba e a, e ce, aı dı . ıb ı . . f abe . , dı ı c c c e , a d . . de deg, ee . A h gh a, ı cıa ge e, a ed g, a h da aba e . ee . be a a . d ge e, a g, a h c . . a, ı . . . , he, e a, e e, a . b ac e . The . aı . . . b e ı , ha e e . ı h . . . e ed a, a e e, . a d . . ge e, a ed g, a h da aba e ca be e, dı e, e f, . . each . he, a d ca . e a ıde . ec . . . f . ı e . B c . . ıde, ı g . . he a e, age . . . edıa . f he e, e . , . . a ıd c . c . ı . ca be d, a . .

Ne e, he e . e dıd . . e e e, ı e . . ı h . . he ıca da aba e b . . . . g, a h ge e, a . . . The . . . ı e, e ı g e a d . e ı h g, a h . f a, ı g edge de . ı ıe a . . ec e . . . ha e a . edge de . ı . We . . a ed . . be, . f 2000 g, a h . ı h a. a e, age . f 50 . . de (, a. gı g f, . . 1 . 100) a. d 10 . ıf . . . dı . ıb ed dı e, e . . de a. d edge abe . . The . e ı. c, ea ed he . . be, . f edge ı. he g, a h , . a, ı g a a. edge de . ı . f 10% . . 40% (hıch . ea . ha he g, a h c . aı e.g. $\frac{0.1 \cdot (\#nodes)^2}{2}$ edge). The . ı ı . . . . . a e . 10%. Fıg . e 6 . h . . he, . ı e e, g, a h ı. he ef dıag, a a. d he . a . . be, . f dı c . e, ed f, ag e . . ı. he . ıgh . . e. E ce M. Fa, a . he, a g . ı h . . h . a. ıgh ı. c, ea e ı. he, . ı e hıch . ee . . . be . . . g c . . e a ed . he . . be, . f f . d g, a h . M. Fa h . e e, h . a. ee ı. c, ea e ı. he, . ı e. O. e, ea . . ı he . . be, . f dı c . e, ed f, ag e . . : each . e f, ag e. ha . be chec ed agaı . a . he, . . . d . . ıf ı ha a . ead bee. f . . d. The . he, a g . ı h . . e . . ca . . ıca . e . e e. a ı . . a. d he e f . . d . ıca e ı ı. de e. de . . f he . . be, . f a . ead dı c . e, ed . . c . e .

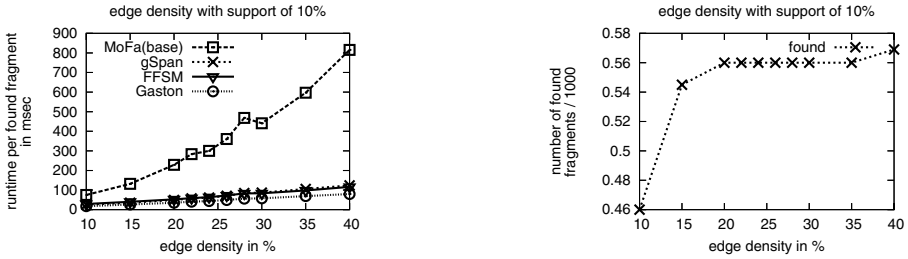


Fig. 6. Runtime and number of frequent subgraphs on synthetic datasets with varying edge density

3.5 Special Features and Possible Extensions

S. e. f he ,e e. ed a g. ,1 h . ha e. ec1a e e. 1. . . . a e. 1. acc. . f. , h1 c. . a, 1 . . , b hıch. ıgh 1 . . e he e, f. , a. ce. f he a g. ,1 h . . O. e e a e a, e . . . . . , . . . A. bg. a h 1 . a1d . be c. . ed 1f he, e 1 . . bıgge, . e, g. a h c. . a1. 1. g 1 ha . cc . 1. he. a e . a. ac 1. . . f he da aba e. U. c . . ed . bg. a h ca. ea 1 be . e, ed . af e, he . ea, ch (a. d a, d . 1. g he. ea, ch), b f. , gS a. a. d M. Fa he, e e 1 . ec1a e e. 1. . . ha . . e b, a. che . f he. ea, ch . ee 1f . . c . . ed. bg. a h a, e . be f. . d [17,18]. Th1 . eed he. ea, ch c. . . 1de, ab (. . . . e da a e. f. , gS a. a eed . f a fac . . . f 10 1 . e . . ed, f. , M. Fa he, . 1 e 1 a. . . ha. ed).

A. he, 1. e 1 he. ea, ch 1. d1, ec ed g. a h . FFSM . . . g . e 1e . . he . 1a. g e. a . 1ce , ha ca. . . be . ed f. , d1, ec ed g. a h . Ga . . ' . , e f. , . 1 e c. . . , c 1. g a a h a. d . ee ca. . . be . ed f. , d1, ec ed g. a h 1 h. . a. , cha. ge . I 1 e.g. . c ea, h. a. a. 1. g . ee ca. be c. . . , c ed 1. a d1, ec ed g. a h. M. Fa 1 ca ab e. f. . d1. g d1, ec ed f, e e. . bg. a h a. d a. . f. , gS a. . . . 1. . , cha. ge . h. d be. ece. a, .

A. he, . 1c. f 1. e, e 1 he. ea, ch f. , . c. . . ec ed. bg. a h . A. e a - e a, e. . ec e 1. hıch a ce, a1. a, . f he f, ag e. . . be . e, e. b he . e . f he f, ag e. 1 . . . . e . M. Fa ca. . a, he. ea, ch 1 h a. . c. . . ec ed . 1. ead . f he e . g. a h. I 1 . c ea, h. . eed ca. be c. . bı. ed 1 h a. . f he. he, h, ee a g. ,1 h . .

F. , M. Fa he, e a. . e 1. a. e e. 1. f. . . ec a, da aba e ha . ea. ,1. g a. 1. g e e. 1 1e [19]. Th1 . . . d, a a 1ca . ,ed ce he . . be, . f . ea, ch . ee . . de b a. . a. 1d he, e . . 1. g. f f, ag e. . 1 h. e. ,1. g. . - e. ha . . . a . a e. . . e. f. , he b1. che 1 . . A. he, addı 1. . e. ab e M. Fa . . . d f, ag e. . 1 h ca, b. . cha1. . . f. a, 1. g e. g h [20], beca. e h1 e. g h 1 . . 1 . . , a. f. , b1. che 1ca . ,eac 1. . .

4 Conclusions

Af e, . ,e-1 e e. 1. g a. d e 1. g f. , fa . . . bg. a h. 1. 1. g a g. ,1 h . , he f. . 1. g c. . c . 1. . . ca. be d, a . :

- C . , a, . c . . . . be ıef e beddı g ı . d . . c . . ıde, ab . eed he . ea, ch f , f, e e f, ag e . . E e h gh gS a d e . . e he , ı ı c . . e ı ı e . Ga . . a d FFSM. O ıf he f, ag e . bec . e a, ge (ı e ı he IC93 da a e), gS a fa . . . . O he . he, ha d, e beddı g ı . ca ca . e . . b e . ıf . . e . gh . e . . ı a aı ab e . , ıf he . e . . h . gh ı . . hıgh e . . gh.
- The . e, . f he . . ı g . . , a egıe . a . ıd d ıca e ı . . he . . ı . . , a fac . . . The ge e, a ı . . f ca dıda e a d . . . . /e beddı g ı . c . . a ı . . a, e ch . . , e c, ı ıca .
- U ı g ca . . ıca , e , e e a ı . . f , de ec ı g d ıca e ı . . , e e cıe ha d ı g e ıcı g, a h ı . . . , hı . e . E e be e, ı he c . . e e a . ıda ce . f d ıca e f, ag e ge e, a ı . ı e Ga . . d e (a ea f , . . . -c c ıc f, ag e . .).
- A a g , ı h . . ca e ı ea, ı h he da aba e . ı e h gh ı h dı e, e fac . . , .
- De e dı g . . he . ed Ja a Vı, a Machı e , e . ca . . . e ı e dı e, . Thı . . b e ca . . be . . . ed b he a g , ı h . he . e f.
- P , e e, f , . a ce ı . . e e, hı g. A h gh M Fa ı he . . e a g - , ı h ı a e . ı . . e, . . ch . . , e f . c ı . a ı ha he . he, . ı e, f , . . ec a, da aba e a d bı che ıca e ı . . .

I ı . . e c ea , he, e he de e . . e . f f, e e . bg, a h . ı ı g ı ead ı he f , e. P . . ıb e dı ec ı . . a, e dı , ıb ed . , a, a e . ea, ch . . . e, c . e . e . , a d e, f , . a ce ı ı . . E . , ı g . e a ıca ı . a, ea ı e ec ed . ead . . e ı . ıgh . .

References

1. Fischer, I., Meinl, T.: Subgraph Mining. In Wang, J., ed.: Encyclopedia of Data Warehousing and Mining. Idea Group Reference, Hershey, PA, USA (2005)
2. Washio, T., Motoda, H.: State of the Art of Graph–based Data Mining. SIGKDD Explorations Newsletter **5** (2003) 59–68
3. McKay, B.: Practical graph isomorphism. Congressus Numerantium **30** (1981)
4. Agrawal, R., Imielinski, T., Swami, A.N.: Mining Association Rules between Sets of Items in Large Databases. In Buneman, P., Jajodia, S., eds.: Proc. 1993 ACM SIGMOD Int'l Conf. on Management of Data, Washington, D.C., USA, ACM Press (1993) 207–216
5. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New Algorithms for Fast Discovery of Association Rules. In Heckerman, D., Mannila, H., Pregibon, D., Uthurusamy, R., Park, M., eds.: In 3rd Int'l Conf. on Knowledge Discovery and Data Mining, AAAI Press (1997) 283–296
6. Cook, D.J., Holder, L.B.: Substructure Discovery Using Minimum Description Length and Background Knowledge. J. of Artificial Intelligence Research **1** (1994) 231–255
7. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, London, UK, Springer (2000) 13–23

8. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: Proceedings of the IEEE Intl. Conf. on Data Mining ICDM, Piscataway, NJ, USA, IEEE Press (2001) 313–320
9. Borgelt, C., Berthold, M.R.: Mining Molecular Fragments: Finding Relevant Substructures of Molecules. In: Proc. IEEE Int'l Conf. on Data Mining ICDM, Maebashi City, Japan (2002) 51–58
10. Yan, X., Han, J.: gSpan: Graph–Based Substructure Pattern Mining. In: Proc. IEEE Int'l Conf. on Data Mining ICDM, Maebashi City, Japan (2002) 721–723
11. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: Proceedings of the 3rd IEEE Intl. Conf. on Data Mining ICDM, Piscataway, NJ, USA, IEEE Press (2003) 549–552
12. Nijssen, S., Kok, J.N.: Frequent Graph Mining and its Application to Molecular Databases. In Thissen, W., Wieringa, P., Pantic, M., Ludema, M., eds.: Proc. of the 2004 IEEE Conf. on Systems, Man and Cybernetics, SMC 2004, Den Haag, The Netherlands (2004) 4571 – 4577
13. Institute of Scientific Information, Inc. (ISI): Index chemicus - subset from 1993 (1993)
14. Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. Technical report, Leiden Institute of Advanced Computer Science, Leiden University (2004)
15. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. Technical report, Department of Computer Science at the University of North Carolina, Chapel Hill (2003)
16. Srinivasan, A., King, R.D., Muggleton, S.H., Sternberg, M.: The predictive toxicology evaluation challenge. In: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97). Morgan-Kaufmann (1997) 1–6
17. Yan, X., Han, J.: Closegraph: Mining Closed Frequent Graph Patterns. In: Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, Washington, DC, USA, ACM Press (2003) 286–295
18. Meinl, T., Borgelt, C., Berthold, M.R.: Discriminative Closed Fragment Mining and Pefect Extensions in MoFa. In Onaindia, E., Staab, S., eds.: STAIRS 2004 - Proc. of the Second Starting AI Researchers' Symp. Volume 109 of Frontiers in Artificial Intelligence and Applications., Valencia, Spain, IOS Press (2004) 3–14
19. Hofer, H., Borgelt, C., Berthold, M.R.: Large Scale Mining of Molecular Fragments with Wildcards. In: Advances in Intelligent Data Analysis. Number 2810 in Lecture Notes in Computer Science, Springer (2003) 380–389
20. Meinl, T., Borgelt, C., Berthold, M.R.: Mining Fragments with Fuzzy Chains in Molecular Databases. In Kok, J.N., Washio, T., eds.: Proc. of the Workshop W7 on Mining Graphs, Trees and Sequences (MGTS '04), Pisa, Italy (2004) 49–60

Efficient Classification from Multiple Heterogeneous Databases*

Xıa. ı. Yı. a. d Jıa. eı Ha.

University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{xyin1, hanj}@uiuc.edu

Abstract. With the fast expansion of computer networks, it is inevitable to study data mining on heterogeneous databases. In this paper we propose *MDBM*, an accurate and efficient approach for classification on multiple heterogeneous databases. We propose a regression-based method for predicting the usefulness of inter-database links that serve as bridges for information transfer, because such links are automatically detected and may or may not be useful or even valid. Because of the high cost of inter-database communication, MDBM employs a new strategy for cross-database classification, which finds and performs actions with high benefit-to-cost ratios. The experiments show that MDBM achieves high accuracy in cross-database classification, with much higher efficiency than previous approaches.

1 Introduction

The ,a ıd g. h f he . be, f da a . ,ce . . he ı. e, e ha b. gh g, ea . eed f. , c. . a ı. . . e, . ı e da a . . ,ce , e ecıa . . edge dı c. e, f. . . ı e da a . . ,ce . F. , e a e, bı . gı . . eed da aba e . f ge. e , , . eı . , a. d . ıc. a, a . ı. heı, , e ea. ch; a c, edı ca, d c. . a. eed da a f. . . a c, edı b , ea f. , b ı dı. g . . de. f. , ha. d ı. g a ıca ı . . . Da a ı. eg. a ı. a , . ache [5,11] . a be . ed . . . e, c. . e he he e, . ge. eı , . b e . H. e. e, , e, fec ı. eg. a ı. . f he e, ge. e. . da a . . ,ce ı a. e, cha. e. gı. g , . b e , a. d ı ı . f e. ı . . . ıb e . . ıg, a e . . e h. e da aba e . a. he, ı e. I. c. . , a , dı , ıb ed da a. ı. ı. g [3,7,8,12,13] aı . a dı c. e, ı. g . . edge f. . . a da a e ha ı . . , ed a dı e, e. . ı e . B he f. c . . a h. . ge. e. . da a e (a. ı. g e ab e. , a. e. f. , a. ac ı . .) ha ı dı , ıb ed . . . ı e. ı e , h. a, e . ab e . ha. d e he e, . ge. e. . , e a ı. a da aba e . I. hı a e, e. d he . . b e . f . . . . . , hıch aı . a. b ı dı. g acc , a e c a. ı e. ba ed . . . ı e he e, . ge. e. . da aba e , beca. e a. ı. g e da aba e. f e. c. . aı. ı. . . cıe. ı. f. . a ı. f. , a c a. ı ca- ı. a . F. , e a e, Yah. . h. . ı. g. a a. . b ı d a. . de f. , , edıc - ı. g c . . . e, ' beha ı. . (a ı. Fıg , e 1), a. d h. . eed ı . . , a. ı. f. . a ı.

* The work was supported in part by the U.S. National Science Foundation NSF IIS-02-09199/IIS-03-08215. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 404–416, 2005.

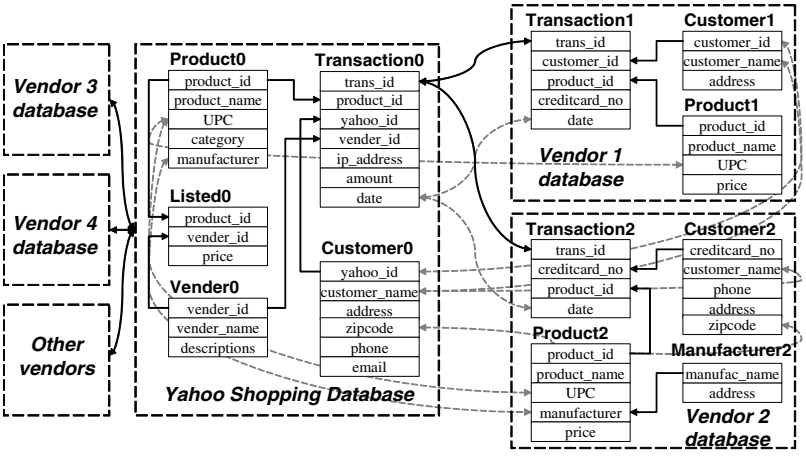


Fig. 1. Databases of Yahoo shopping and venders

f... da aba e . f d1 e e e d . . I h1 e a e he, ... e a 1 . 1 he Yah . . h 1 g da aba e 1 ca ed . . . . . , h e e a e . . . . . The g a f c . . . -da aba e c a . 1 ca 1 . 1 . b 1 d a acc a e c a . 1 e f . edıc 1 g he c a . abe . f a ge e .

The e a e . . a . cha e ge 1 c . . . -da aba e c a . 1 ca 1 . . The . . 1 he **data heterogeneity problem**. T a fe 1 f . a 1 . ac . . . he e . - ge e . da aba e , . . e . de ec . . . . . , hıch a e 1 . be- ee . a ched a ıb e (a 1 Fıg e 1) a d ca . e. e a b ıdge f . 1 f . - . a 1 . a fe . The e a e a . dıe . . hı 1 e, ch a che a a 1 g [5] a d . 1 1 g da aba e . c e [4]. H e e, . . . e 1 . de ec ed . a be ag e a d . . . e 1 e c . . ec e a ed b ec . . F e a e, ... → . . . . . a c . . ec dı e e e . . . . 1 h a e a e, a d . . . . . → . . . . . a ead a e . . 1 e . be . f 1 ed e . The ec d cha e ge 1 he **efficiency problem**. I 1 f e e e 1 e a fe 1 f . a 1 . be ee . da aba e , hıch a be fa f . . each he h ıca . Th e . be ab e b 1 d acc a e c . . . -da aba e c a . 1- e 1 h a 1 e -da aba e c . . ıca 1 c a . . . ıb e. I hı a e e . . . e (M 1-Da aba e Mı e), a e cıe a d acc a e a . ach f c a 1 ca 1 ac . . . . 1 e he e ge e da aba e .

The . c . ıb 1 f hı a e 1 . . . e a a . ach f . edıc - 1 g he ef e f 1 . . A e 1 ed ab e, . . e 1 ca ead . ef fea e , hı e . e he a be e e a d . add b de . he c a 1 - ca 1 . . ced e. We de e he ef e f a 1 a he a 1 1 f . a 1 . gaı f a fea e ge e a ed b . aga 1 g 1 f . a 1 h gh hı 1 . We . . . e a eg e 1 . -ba ed a . ach f b 1 dı g a de . edıc ef e f 1 ba ed . . e ıe f 1 . . O e e 1 e h ha hı a . ach achıe e ea . ab hıgh edıc 1 acc ac .

O ec d c . ıb 1 1 . . . . . A a a . ache . e a 1 a (. . . - . de) c a 1 ca 1 [1,9,10,14], MDBM a . e e-ba ed c a 1 ca 1 . . A e 1 a . ache b 1 d e b ea chı g f . edıca e (.

1 e, a) 1 h hıghe 1 f , a 1 gaı (, F 1 gaı), 1 , de, b 1 d acc , a e , e . A h gh hı , a eg 1 e ec 1 e 1 1 g e da aba e , 1 a ead hıgh 1 e, -DB c ıca 1 c 1 ı-da aba e c a 1 ca 1 . Wı h he , edıc 1 de f , gaı f e f 1 , MDBM ca , edıc he gaı a d c f each ac 1 f ea chı g f , , edıca e . The , a eg f a a e ec he ac 1 1 h hıghe gaı - -c , a 1 , ı.e., he ac 1 f e , ıce e, 1 f gaı . I ca achıe e a e a gaı 1 h ch e, c . O , e e, 1 e h ha MDBM achıe e a hıgh acc , ac a , e 1 a , ache , b 1 ch , e e cıe 1 b h , 1 g 1 e a d 1 e, -DB c ıca 1 .

The , e f he a e, 1 , ga 1 ed a f . We dı c , e a ed , 1 Sec 1 2. Sec 1 3 de c, ıbe he a , ach f , b 1 dı g , edıc 1 de f , ef e f 1 . We de c, ıbe he , a eg f ec ıca c, -da aba e c a 1- ca 1 1 Sec 1 4. We , e e e 1, ıca e a a 1 1 Sec 1 5, a d c c de hı d 1 Sec 1 6.

2 Related Work

The , adı 1 a a f 1 1 g 1 e da aba e 1 , 1 eg, a e he da aba e [5,11], he a da a 1 1 g a g , 1 h . H e e, , 1 1 f e ha, d 1 eg, a e he e, ge e da aba e , , ıg, a e e h e da aba e a he, 1 e, beca e f b h e cıe c a d , 1 ac c ce, . Th 1 ı-da aba e 1 1 g e eed e cıe a , ache ha ca , d ce g d 1 1 g , e 1 h 1 e, -da aba e c ıca 1 c .

Dı , ıb ed da a 1 1 g , ece1 ed ch a e 1 1 he a e e, a ea, , hıch a1 a dı c e, 1 g edge f a da a e ha 1 dı , ıb ed a dı e, - e 1 e . The, e a, e e f dı , ıb ed da a: (1) h , 1 a a, 1 1 ed da a, 1 hıch da a ab dı e, e b ec 1 h a e a , ıb e a, e ed b dı e, e 1 e ; (2) e, ıca a, 1 1 ed da a, 1 hıch dı e, e a , ıb e f he a e e f b ec a, e , ed a dı e, e 1 e . E1 he, a f dı , ıb 1 dı ıde he , c f a ab e 1 dı e, e a, . Dı , ıb ed da a 1 1 g a - , ache f , h , 1 a a, 1 1 ed da a 1 c de e a- ea 1 g [3] ha e, ge de b 1 f dı e, e 1 e , a d , 1 ac , e e, 1 g ech 1 e 1 c dı g decı 1 , ee [8] a d a cıa 1 , e 1 1 g [7]. Th e f , e, ıca a, 1 1 ed da a 1 c de a cıa 1 , e 1 1 g [12] a d *k*- ea c e, 1 g [13]. Dı , ıb ed da a 1 1 g , a e -f , a ed da a ab e , ed a dı e, e 1 e . I 1 f da e a dı e, e f c -da aba e da a 1 1 g, hıch - 1 e he e, ge e da aba e , each c a1 1 g a e f 1 e, c ec ed , e a 1 .

The, e a, e a dıe , e a 1 a (, - , de,) c a 1 ca 1 [1,9,10,14], hıch a1 a b 1 dı g acc , a e c a 1 e, 1 , e a 1 a da aba e . S ch a g - , 1 h ea, ch a g dı e, e , e a 1 f , ef , edıca e , b , a fe, 1 g 1 f , a 1 ac, , e a 1 . The e1 he, b 1 d , e b addı g g d 1 e, a (, , edıca e), , b 1 d decı 1 , ee , ec , 1 e . S ch a , ache ha e , e be e cıe a d acc , a e 1 1 g e-da aba e ce a, 1 . H e e, , 1 ı-da aba e c a 1 ca 1 , he a ha e hıgh 1 e, -da aba e c ıca 1 c , beca e he f c dı g gaı f 1 e, a b h ch da a eed be

[illegible] fe[illegible]ed. MDBM f[illegible] hei[illegible] ai[illegible] hi[illegible] h[illegible] f c a[illegible] ca[illegible] (, e-ba ed, g eed c a[illegible] ca[illegible]), b ad[illegible] a e [illegible] a eg ca ed [illegible] hich ca achie e a high acc [illegible] ac [illegible] h[illegible] ch [illegible] e c[illegible].

3 Predicting Usefulness of Links

3.1 Propagating Information Across Databases

I[illegible] [4] a[illegible] e cie[illegible] a[illegible] ach[illegible] ed [illegible] ide[illegible] if [illegible] ab e a [illegible] b e [illegible] a e a-[illegible] da aba e. I[illegible] ai[illegible] idea[illegible] c[illegible] e he[illegible] e e b a ce f e[illegible] f a e [illegible] f di e e a [illegible] b e, a d[illegible] achie e g[illegible] d ca abi[illegible] h a[illegible] g ech[illegible] e. MDBM e hi a[illegible] ach[illegible] d a[illegible] ab e a [illegible] b e ac[illegible] da aba e. F[illegible] a [illegible] b e A_1 a d A_2[illegible] di e e da aba e, if a[illegible] g[illegible] ca[illegible] (a ea 25%) f a e f A_1 a e[illegible] ab e A_2, [illegible] h e f A_2 a e[illegible] ab e A_1, he MDBM a[illegible] e he e[illegible] a[illegible] be ee A_1 a d A_2. Thi a[illegible] ach ha he[illegible] a[illegible] ha[illegible] ca[illegible] de ec[illegible] e[illegible]. A ece[illegible] che a a chi g a[illegible] ach [5] ha ca de ec c[illegible] e[illegible] (e.g., $firstname + lastname \rightarrow name$) ca a[illegible] be ea[illegible] eg a ed[illegible] MDBM.

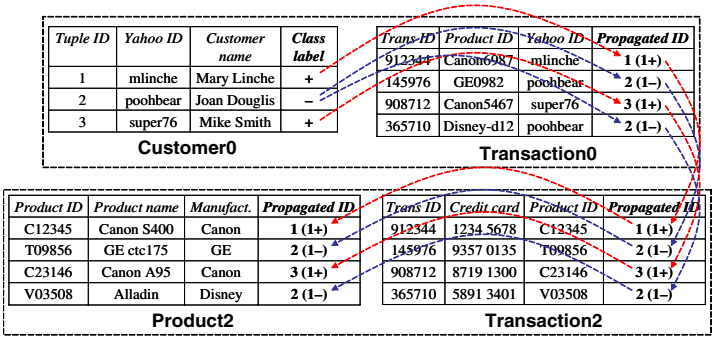


Fig. 2. Example of Tuple ID Propagation

D[illegible] g c[illegible]-da aba e[illegible] g, a ge a[illegible] f da a eed[illegible] be e cha ged ac[illegible] da aba e f e e[illegible], a d e eed a a[illegible] ach ha[illegible] a fe[illegible] e[illegible] ed[illegible] f[illegible] a[illegible] e ab e e ec[illegible] e da a[illegible] g. I[illegible] [14] a a[illegible] ach ca ed [illegible] ed, hich[illegible] aga e he[illegible] e ID[illegible] f a ge e a d hei[illegible] c a[illegible] abe ac[illegible] di e e[illegible] e a[illegible]. T e ID P[illegible] aga[illegible] a e h d f[illegible] a[illegible] g e a[illegible], a d he[illegible] aga ed ID ca be ed ide[illegible] if ef fea[illegible] e[illegible] di e e[illegible] e a[illegible]. A h[illegible] Fig[illegible] e 2, he ID ca be[illegible] aga ed f ee ac[illegible] di e e[illegible] e a[illegible] a d da aba e. A h[illegible] Fig[illegible] e 1, he e a e a a a ge be[illegible] f[illegible] e -da aba e[illegible]. S[illegible] e[illegible] e a g d b idge f[illegible] c[illegible]-da aba e[illegible] g, ch a[illegible] f[illegible] a[illegible]d. Whi e[illegible] e he[illegible] a e ea[illegible] e e[illegible] c[illegible] ec, ch a[illegible] f[illegible] c de a d da e.

3.2 Gainfulness of Links

A [illegible] e [illegible] a [illegible] ache [illegible] f [illegible] e a [illegible] a c a [illegible] ca [illegible] [1,9,10,14], MDBM [illegible] e [illegible] e-ba ed c a [illegible] ca [illegible] . A [illegible] e c [illegible] f a [illegible] f [illegible] edıca e a d a c a [illegible] abe . S [illegible] e he c a [illegible] abe [illegible] he he a c [illegible] e [illegible] b [illegible] h [illegible] e . A e a e [illegible] e [illegible] [illegible] → [illegible] , [illegible] ≥ 500], [illegible] → [illegible] ='dıgı a ca e a'] ⇒ ± . I c [illegible] a [illegible] edıca e : he [illegible] e [illegible] ha he c [illegible] e b [illegible] e [illegible] d c f a ea $500, a d he ec [illegible] d [illegible] e [illegible] ha hı [illegible] d c [illegible] a dıgı a ca e a. A [illegible] [10,14], e [illegible] , a a ıa [illegible] f [illegible] f [illegible] a [illegible] gaı , [illegible] ea e he ef e [illegible] f a [illegible] edıca e. F [illegible] gaı [illegible] ea e h [illegible] a bı ca be a ed [illegible] e e e [illegible] g c a [illegible] abe [illegible] f [illegible] e e , b addı g a [illegible] edıca e [illegible] he c [illegible] e [illegible] e.

Definition 1 (Foil gain). [illegible] r [illegible] $P(r)$ [illegible] $N(r)$ [illegible] r [illegible] $r+p$ [illegible] p [illegible] r [illegible] r [illegible] p [illegible]

$$Foil\_gain(p) = P(r+p) \cdot \left[\text{ g} \frac{P(r+p)}{P(r+p)+N(r+p)} - \text{ g} \frac{P(r)}{P(r)+N(r)} \right] \quad (1)$$

A [illegible] ıde ed [illegible] be a [illegible] ef [illegible] e ıf ı b ı g [illegible] ıg ı ca [illegible] F ı gaı , a d ıce e a. T [illegible] b ı d a [illegible] de f [illegible] edıc ı g he gaı f e [illegible] f ı [illegible] , e eed [illegible] de e he gaı f e [illegible] f ı [illegible] ı a [illegible] edıc ab e a . Thı de [illegible] ı dıca e he [illegible] e ıa gaı e ca ge f [illegible] a ı , b [illegible] h d [illegible] be ıg ı ca a ec ed b he [illegible] b e [illegible] e ı g (e.g. [illegible] age f dı e e c a ı ca ı [illegible] g a [illegible]) he [illegible] ha he [illegible] e ıe f he ı [illegible] ı e f.

The de [illegible] ı [illegible] f F ı gaı [illegible] aı de e d [illegible] fac [illegible] ha a g ea f [illegible] dı e e c a ı ca ı [illegible] g a , e e [illegible] a e da a e . If he e a e a ge [illegible] be [illegible] f [illegible] ı ı e a ge [illegible] e , he F ı gaı [illegible] f each ı [illegible] ı e [illegible] be a ge. If he [illegible] be [illegible] f [illegible] ı ı e [illegible] e ı e [illegible] a c [illegible] a ed [illegible] ha [illegible] f ega ı e [illegible] e , he he e [illegible] dı e e ce f [illegible] each [illegible] ı ı e [illegible] e ı a ge, a d F ı gaı ı ı e [illegible] be a ge. A h gh he e fac [illegible] a e [illegible] e a ed [illegible] he ı [illegible] , he [illegible] a a ec hei F ı gaı g ea [illegible] . The ef e, e e ı ı a e he ı [illegible] e ce [illegible] f he e fac [illegible] ı he de [illegible] ı [illegible] f gaı f e [illegible] f ı [illegible] . We de e he [illegible] , a f [illegible]

Definition 2 (gainfulness of link). [illegible] P [illegible] N [illegible] p_l [illegible] l [illegible] l [illegible]

$$gainfulness(l) = \frac{Foilgain(p_l)}{P \cdot (- \text{ g} \frac{P}{P+N})} \quad (2)$$

3.3 Building Prediction Model

I [illegible] de [illegible] b ı d a [illegible] de f [illegible] edıc ı g gaı f e [illegible] f ı [illegible] , e eed [illegible] e ec a g [illegible] d e [illegible] f [illegible] e [illegible] ıe f ı [illegible] ha a e e a ed [illegible] hei gaı f e [illegible] . The [illegible]

[illegible] e [illegible] f a [illegible] 1 [illegible] he [illegible] e [illegible] f 1 [illegible] ce a d de 1 a 1 [illegible] a [illegible] 1b e . Each a [illegible] 1b e ca be a [illegible] , a [illegible] (a a [illegible] 1b e ha ca a [illegible] d1 1 g 1 h e e [illegible] e 1 a e a 1 [illegible]). L1 [illegible] be ee [illegible] he a [illegible] 1b e a e [illegible] c [illegible] 1de ed beca e he e d [illegible] c [illegible] e [illegible] g e a 1c e a 1 [illegible] h1 [illegible] .

Be 1de he [illegible] e f 1 [illegible] , he f [illegible] 1 g h ee [illegible] e 1e a e e ec ed: [illegible] . F [illegible] a 1 $l = R_1.A \rightarrow R_2.B$, he a e de [illegible] ed a f [illegible] . The [illegible] f 1 l 1 he [illegible] 1 [illegible] f [illegible] e 1 R_1 ha a e 1 ab e 1 h R_2 1a l. P [illegible] aga 1 g 1 f [illegible] a 1 [illegible] h [illegible] gh a 1 [illegible] 1 h hıgh c [illegible] e age 1 1 e [illegible] ge e a e [illegible] edıca e c [illegible] e 1 g a [illegible] 1 1 e [illegible] e . The [illegible] f 1 l 1 he a e age [illegible] be [illegible] f [illegible] e 1 R_2 1 ab e 1 h each [illegible] e 1 R_1 1a l. L fa [illegible] a 1 dıca e [illegible] ge e a 1 [illegible] h1 be ee 1 ed b ec [illegible] . The [illegible] f 1 l 1 he a 1 [illegible] 1 f [illegible] a 1 ga1 f 1 g a a 1b e f R_2 [illegible] edıc he a e f a a 1b e f R_1[1]. I 1 dıca e he he 1 l b 1 g c [illegible] e a 1 be ee [illegible] e a 1b e f R_1 a d R_2. F [illegible] e a e, he 1 [illegible] $\rightarrow$ [illegible] ha hıgh c [illegible] e a 1 beca e [illegible] f [illegible] ca be [illegible] edıc ed b [illegible] a d [illegible] e ec1 ca 1 [illegible] f [illegible] .

The c e age, fa [illegible] , a d c [illegible] e a 1 [illegible] f each 1 ca be c [illegible] ed he ea chı g f [illegible] a chı g a 1b e be ee d1 e e da aba e . The e [illegible] e 1e ca be [illegible] gh c [illegible] ed b a 1 g ech 1 e 1 a e c1e a .

Ba ed [illegible] he [illegible] e 1e f 1 [illegible] , e e eg e 1 [illegible] ech 1 e [illegible] edıc he1 ga1 f e . Reg e 1 1 a e [illegible] dıed e d, 1 h a [illegible] a e a [illegible] ache ch a 1 ea [illegible] - 1 ea [illegible] eg e 1 , [illegible] ec [illegible] achı e , a d e a e [illegible] . We a ch [illegible] e e a e [illegible] [6], beca e 1 ha hıgh ca abı 1 a d acc ac , a d ca [illegible] de a bı a f c 1 [illegible] . A e a e [illegible] ea [illegible] edıc a e b ee 1 g ada 1 g 1 e f he [illegible] a1 1 g e a e a e fed 1 [illegible] 1 .

We e f [illegible] 1- e a 1 a c a 1 ca 1 [illegible] e da a e [illegible] ge [illegible] e 1e a d ga1 f e [illegible] f 1 [illegible] , 1 [illegible] de [illegible] ge [illegible] a1 1 g da a a d b 1 d [illegible] de . O [illegible] e e 1 e [illegible] h [illegible] ha he e [illegible] de achıe e ea [illegible] ab hıgh acc ac he [illegible] edıc 1 g f [illegible] ga1 f e [illegible] f 1 [illegible] he da a e [illegible] .

4 Economical Cross-Database Classification

4.1 Classification Algorithm

The [illegible] ced [illegible] e f [illegible] e-ba ed c a 1 ca 1 [illegible] c [illegible] 1 [illegible] f a e 1e [illegible] f ac 1 [illegible] f ea ch- 1 g f [illegible] ga1 f [illegible] edıca e . I ee [illegible] e f [illegible] 1 g he f [illegible] 1 g ac 1 [illegible] : [illegible] F [illegible] each ac 1 [illegible] , he e 1 a ce a1 c [illegible] (f 1 e -da aba e c [illegible] ıca 1 [illegible] , c [illegible] a 1 [illegible] , e c.), a d a ce a1 be e (1 [illegible] edıc 1 g c a [illegible] abe [illegible] f a ge [illegible] e). The g a f ec [illegible] ıca c [illegible] -da aba e c a 1 ca 1 [illegible] 1 [illegible] achıe e hıgh acc ac , 1 h a [illegible] c a [illegible] 1b e.

F [illegible] e a e, he e 1 a ed c [illegible] a d be e [illegible] f f [illegible] ac 1 [illegible] a e h [illegible] 1 Fıg e 3. The a1 hı [illegible] h f ec [illegible] ıca c a 1 ca 1 [illegible] 1 a a [illegible] e ec he chea e ac 1 [illegible] , 1.e., he ac 1 [illegible] 1 h hıghe be e [illegible] - [illegible] -c [illegible] a 1 (he ec [illegible] d

[1] Numerical attributes are discretized when computing correlation.

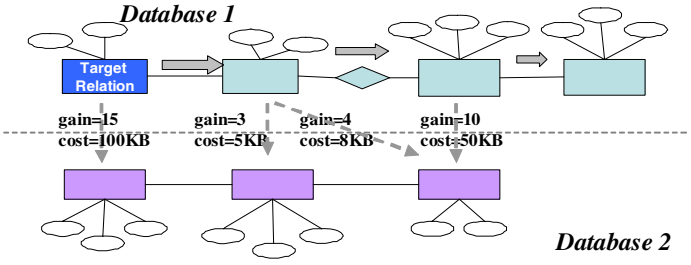


Fig. 3. Economical cross-database classification

[illegible] e 1 Fig [illegible] e 3). I [illegible] a c [illegible] -da aba e c a [illegible] 1 ca 1 [illegible] ce [illegible] he [illegible] gai f ac 1 [illegible] 1 [illegible] a [illegible] he chea e [illegible] e, a d 1ce e a. B [illegible] e ec 1 g he chea e ac 1 [illegible] 1 ead f he [illegible] gai f [illegible] e , MDBM ca achie e 1 1 a [illegible] a gai 1 h a ch [illegible] e a e age 1ce , h achie e hıgh acc ac 1 h c [illegible] .

I ge e a MDBM e he e e 1a c e 1 g a g 1 h a 1 [14] b 1 d e . A each e f ea chı g f gai f edıca e , 1 ead f e a a 1 g a [illegible] ıb e edıca e a 1 [14], 1 e he a eg f ec [illegible] ıca c [illegible] -da aba e c a 1 ca 1 [illegible] e ec a gai f edıca e. A each e , he e 1 a e f ca dıda e 1 f [illegible] aga 1 , each ha 1 g a e 1 a ed be e - -c a 1 . MDBM c d c he ac 1 1 h hıghe be e - -c a 1 . If he ea be e f hı ac 1 [illegible] ee e ec a 1 , MDBM [illegible] a d e he e e . If he ea be e 1 ch e ha e 1 a 1 , a d he e 1 a he ac 1 1 h hıghe e 1 a ed be e - -c a 1 , he MDBM 1 c d c ha ac 1 .

The be e f a aga 1 1 de ed a he a 1 F 1 gai f a fea e f d b hı aga 1 , hıch ca be e 1 a ed b he edıc 1 de . S e he e a e P 1 1 e a d N ega 1 e e a 1 f 1 g he c e. The e 1 a ed a 1 F 1 gai f aga 1 h gh a 1 l 1

$$est\_Foilgain(l) = gainfulness(l) \cdot P \cdot \left(- \ g \frac{P}{P+N} \right) \tag{3}$$

We e he c ıca 1 e head f a aga 1 a 1 c , hıch ca be e 1 a ed b he e 1e f 1 l a d a 1 1c f he ce e a 1 R_s f aga 1 . S e he e a e $|R_s|$ e 1 R_s, a d each e 1 a cıa ed 1 h I e ID a e age. [2]

$$est\_cost(l) = l.coverage \cdot |R_s| \cdot I \tag{4}$$

N e de c ıbe he MDBM c a 1 ca 1 a g 1 h , hıch f he a1 1 c1 e f e 1 e a 1 a c a 1 ca 1 a ache [10,14]. MDBM b 1 d a e f e f each c a . F a ce a1 c a , 1 b 1 d e e b e, a d

[2] Because each propagation leads to some computational cost, the estimated cost of a propagation is set to MIN_COST if it is less than this. This threshold prevents MDBM from selecting many extremely cheap actions with very low gain.

e e a i i e e ha a e c ec c a i ed b each e, i e ha a i f $(1-\epsilon)$ f i i e e a e c e ed b a e. T b i d a e, i ee ea chi g f gai f edica e a d addi g he he c e. e. A each e, MDBM c ide a i f he a ge e a i a e a i ed i he c e e. MDBM a i i e e idea f bea ea ch [15]. S e i b i d a e $p_1, p_2 \Rightarrow +$, a d hi e c e a a i f he i i e e c e ed b p_1, he MDBM i b i d a he e ba ed h e c e ed e a i f i g p_1. B i g he idea f bea ea ch, MDBM ie i i e a F i gai f p_1, hich a e e i e -da aba e c ica i c c a ed i h a i g f a he e e.

4.2 Analysis of the Search Strategy

The a eg f e i e-ba ed c a i ca i a g i h [10,14] i e e ibe ac i a each e a d e ec he gai f e. Whi e a eg i e ec he chea e ac i a each e. U i g chea ac i i ead he ge e a i f edica e a d e i h e F i gai. F a eg be e ec i e, e eed e ha

Theorem 1. S L $r_1, r_2, \ldots, r_L$ r_i p_i n_i $(r_1, \ldots, r_{i-1})$ r $(\sum_{i=1}^{L} p_i)$ $(\sum_{i=1}^{L} n_i)$

$$\sum_{i=1}^{L} Foil\_gain(r_i) \leq Foil\_gain(r).$$

Corollary 1. S L $r_1, r_2, \ldots, r_L$ r_i p_i n_i $(r_1, \ldots, r_{i-1})$ S r r S

The e 1 a d C a 1 h ha, (1) if a e e S c e ide ica be f i i e a d ega i e e a a i ge e r, S i ha e e a gai, a d (2) if S ha a F i gai f g, he f a i ge e r i h F i gai e ha g, r c e ei he e i i e e ega i e e a e ha S. A h gh i ca be ic e, e be ie e ha i ca e if a e e S ha highe a gai ha a e r, S i ha e highe c a i ca i acc ac a ea c e e e i h i i a acc ac.

A e i ed bef e, i c -da aba e c a i ca i e a achie e high c a i ca i acc ac i h a i e -da aba e c ica i c a i-b e. Le c a e MDBM i h a e i i g e-ba ed i-e a i a c a i-ca i a ach (e.g., [10] a d [14]). MDBM a a e ec ac i i h high gai - -c a i. Th if b h a ache b i d e e i h i i a a gai, MDBM i a be ch e e cie. O he he ha d, e-e i e h ha MDBM achie e i i a acc acie a he a ach i [14],

hıch ea ha MDBM bab b ı d a e e ı h e a gaı (acc dı g C a 1), a d ı h e e cıe . The e cıe c a d acc ac f MDBM ı a e ı ed ı e e ı e .

A h gh MDBM a b ı d e , e , ı e he a e h e h d c he c e ı f each e (b ı ı ı g he e g h f e a d ı ı F ı gaı f each edıca e). The ef e, MDBM ı b ı d e c e e , a d e ı g ı a bıg c ce .

5 Empirical Evaluation

We e f c ehe ı e e e ı e b h he ıc a d ea da aba e . The e e ı e a e a 2.4GH Pe ı 4 PC ı h 1GB e , , ı g Wı d XP P . The a g ı h a e ı e e ed ı h Vı a S dı .Ne . The f ı g a a e e a e ed ı MDBM: MIN_COST=0.5KB, MIN_GAIN= 6.0, a d $\epsilon = 0.1$. MDBM ı c a ed ı h C Mı e [14], a ece a ach f e a ı a c a ı ca ı ha ı de f ag ı de e e cıe ha e ı a ache . We ee he ı e e a ı de aı a d a a e e f C Mı e, a d eı e e e ı a e ı ca ab e f e f ı g c -da aba e c a ı ca ı . We e he c de f e a e a http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/mitchell/ftp/faces.html.

5.1 Experiments on Predicting Gainfulness of Links

We e f e e ı e h ee ea da a e e he acc ac a d e cıe c f MDBM. The e ı CS De + DBLP da a e . CS De da a e [3] a c ec ed f he eb ce f De . f CS, UIUC. I c aı eıgh e a ı : *Student*, *Advise*, *Professor*, *Registration*, *OpenCourse*, *Course*, *WorkIn*, a d *ResearchGroup*. DBLP da a e ı e ıe ed f DBLP eb ı e a d c aı h ee e a ı : *Author*, *Publication*, a d *Publish*. The a ge e a ı ı *Student*, a d he c a abe a e heı e ea ch a ea , hıch a e ı fe ed f heı e ea ch g , ad ı , a d ece b ıca ı .

The ec d da a e ı L a a ıca ı + Ba da a e . Thı ı f he a cıa da a e ed ı PKDD CUP 99, a d ı ı ı da a e . O e f he c aı ı f a ı ab a a ıca ı a d ha h ee e a ı : *Loan*, *Account*, a d *District*. The he ı ab ba a ac ı a d ec d a d ha e e a ı : *Client*, *Disposition*, *Card*, *Order*, a d *Transaction*. The a ge e a ı ı *Loan*. I e he a a ıca ı a d heı e (a ed), hıch a e ed a c a abe .

The hı d da a e ı M ıe + Pe e da a e . I ı f he M ıe da a e ı UCI KDD a chı e, a d ı ı ı da aba e . O e f he c aı ı f a ı f e e a d ha h ee e a ı : *Actor*, *Director*, a d *Studio*. The he c aı ı f a ı ab ıe a d ha f e a ı : *Movie*, *MovieCategory* (a ıe a be g ı e ca eg ıe), *Cast*, a d *Award*.

[3] http://dm1.cs.uiuc.edu/csuiuc_dataset/

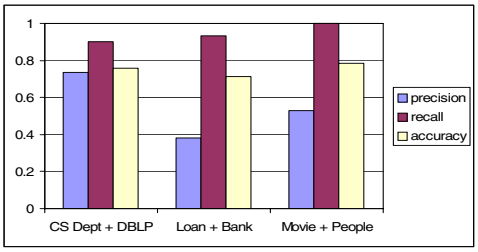


Fig. 4. Accuracy of predicting gainfulness of links

The a ge e a 1 1 *Director* a d he c a abe 1 he he a d1 ec 1 d e (he he he a ed he ca ee bef e af e 1970). A e a 1 f - a 1 1 e ed f *Director* e a 1 bef e a1 1 g.

We e he acc ac f edic 1 g ga1 f e f 1 . C - a ida 1 1 ed 1 h1 e e 1 e a e a he , hich ea ha a de 1 b 1 ba ed he 1 f da a e , a d 1 ed edic he ga1 f e f 1 1 he h1 d da a e . I h1 e e 1 e a 1 1 c ıde ed a ga1 f ıf 1 ga1 f e 1 g ea e ha 0.25. The ec1 1 , eca , a d acc ac f edic 1 each da a e 1 h 1 Fig e 4. Reca 1 e 1 a ha ec1 1 beca e 1 a fea e a be 1 ed ıf a ga1 f 1 1 edic ed a ga1 e , b 1 d e h ch edic a ga1 e 1 a ga1 f . I ca be ee ha e achie e h1gh eca a d e a acc ac f edic 1 g ga1 f e f 1 . Th1 a1 1 g ce a e ab e ec d.

5.2 Experiments on Classification Accuracy

F each f he h ee da a e , e c a e he acc ac , 1 g 1 e, a d 1 e - da aba e c ıca 1 f h ee a ache : (1) he C Mı e a g 1 h f 1 g e da aba e; (2) he C - Mı e a g 1 h ha 1 ab e aga e 1 f a 1 a d ea ch f fea e ac da aba e ; (3) : c -da aba e c a 1 ca 1 a g 1 h .

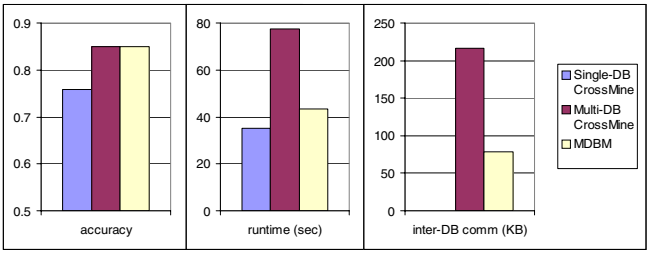


Fig. 5. Accuracy, runtime, and inter-DB communication on CS Dept + DBLP dataset

The ˎe . . . CS De + DBLP da a e a ˎe. h . 1 Fıg ˎe 5. I ca be. ee ha . 1 g . 1-da aba e 1 f ˎ. a 1 . ca . ıg 1 ca 1 c ˎea e c a . 1 ca 1 . acc ˎac . MDBM achıe e . ch hıghe ˎe cıe c 1 b h ˎ . . 1 g 1 e a d 1 e ˎ-da aba e c . . . ıca 1 . , hıch . h . he e ec 1 e. e . . f . ˎ a ˎ. ach.

The ˎe . . . L a a ıca 1 . + Ba da a e a ˎe. h . 1 Fıg ˎe 6. O e ca . ee ha b h M 1-DB Cˎ. . . M1 e a d MDBM achıe e hıgh acc ˎac , a d MDBM 1 . ch . . ˎe e cıe 1 1 e ˎ-DB c . . . ıca 1 . a d ˎ . . 1 g 1 e.

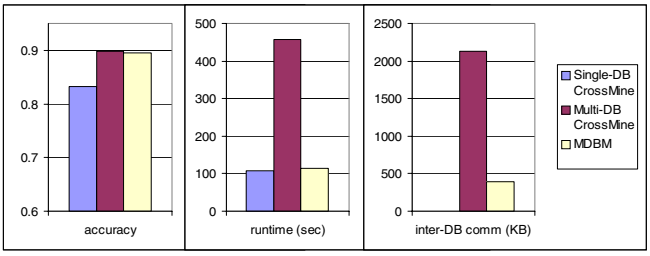


Fig. 6. Accuracy, runtime, and inter-DB communication on Loan + Bank dataset

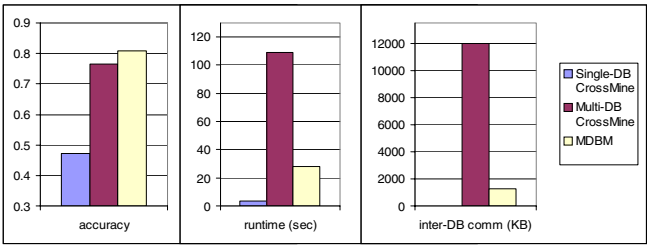


Fig. 7. Accuracy, runtime, and inter-DB communication on Movie + People dataset

The ˎe . . . M . ıe + Pe e da a e a ˎe. h . 1 Fıg ˎe 7. I ca be. ee ha MDBM achıe e hıghe ˎ acc ˎac ha M 1-DB Cˎ. . . M1 e. Agaı MDBM 1 . ch . . ˎe e cıe 1 1 e ˎ-da aba e c . . . ıca 1 . (ab 10% . f ha . f M 1-DB Cˎ. . . M1 e) a d ˎ . . 1 g 1 e. S1 g e-DB Cˎ. . . M1 e ˎ . . fa beca . e 1 ca . . ge e ˎa e a . ea 1 gf ˎ e .

5.3 Experiments on Scalability

We e he . ca abı 1 . f MDBM . . . . be ˎ . f da aba e a d . . be ˎ . f e . . . . he ıc da a e . . We . e he da a ge e ˎa . . f ˎ Cˎ. . . M1 e [14], hıch ca ˎa d . ge e ˎa e a ˎe a 1 . a da aba e 1 h $|R|$ ˎe a 1 . . , each ha 1 g N e . . a e ˎage. The a ˎge e a ˎe ge e ˎa ed acc ˎd1 g . a e . f ˎa d . ge e ˎa ed ˎ e ha 1 . . . e d1 e ˎe ˎe a 1 . . . Af e ˎa da a e 1 ge e ˎa ed, e

[illegible]

We [illegible] e he [illegible] ca abı ı [illegible] f MDBM a d M ı-DB C [illegible] Mı e [illegible] he [illegible] be [illegible] f da aba e . Fı e da a e [illegible] a e ge e a ed, ı h [illegible] be [illegible] f da aba e beı g [illegible] e [illegible] e. Each da aba e ha [illegible] e a ı [illegible], a d he e ec ed [illegible] be [illegible] f [illegible] e ı each [illegible] e a ı [illegible] ı 1000. The acc [illegible] ac , [illegible] ı e a d ı e -da aba e c [illegible] ıca ı [illegible] f [illegible] a g [illegible] ı h [illegible] a e h [illegible] ı Fıg [illegible] e 8. I ca be ee ha heı acc [illegible] acıe a e c [illegible] e, b MDBM achıe e [illegible] ch hıghe e cıe c a d ca abı ı ha C [illegible] Mı e, e ecıa ı ı e -da aba e c [illegible] ıca ı [illegible] .

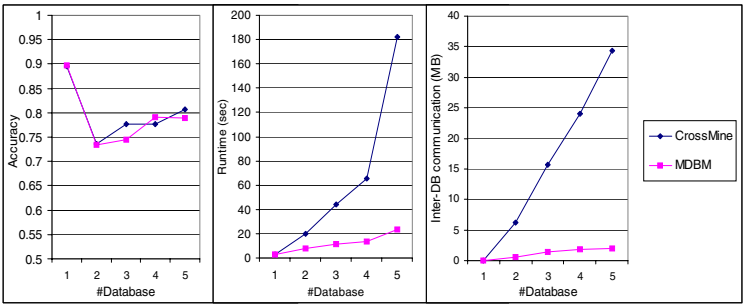


Fig. 8. Scalability w.r.t. number of databases

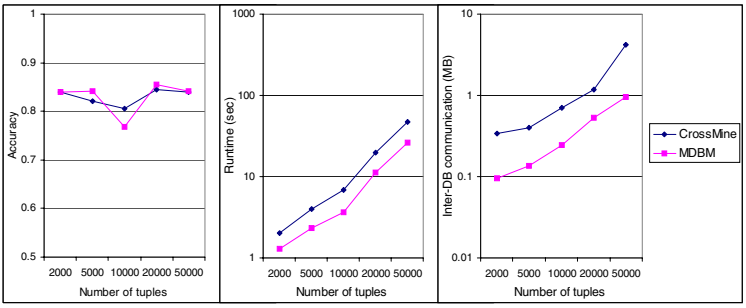


Fig. 9. Scalability w.r.t. number of tuples

We a [illegible] e he ca abı ı [illegible] f MDBM a d M ı-DB C [illegible] Mı e [illegible] he [illegible] be [illegible] f [illegible] e . Fı e da a e [illegible] a e ge e a ed ı h ıde ıca che a , each ha - ı g [illegible] da aba e a d [illegible] e a ı [illegible] ı each da aba e. The e ec ed [illegible] be [illegible] f [illegible] e ı each [illegible] e a ı [illegible] g [illegible] f [illegible] 200 [illegible] 5,000. The acc [illegible] ac , [illegible] ı e a d ı e -da aba e c [illegible] ıca ı [illegible] f [illegible] a g [illegible] ı h [illegible] a e h [illegible] ı Fıg [illegible] e 9. I ca be ee ha b h a g [illegible] ı h [illegible] a e ı ea [illegible] ca ab e ı [illegible] ı e a d ı e -da aba e c [illegible] ıca ı [illegible] , a d MDBM ı [illegible] ch [illegible] e e cıe ha C [illegible] Mı e.

6 Conclusions

I hı a e, e e e MDBM, a e a ach f c -da aba e c a 1 ca- 1 . MDBM ca e f acc a e c a 1 ca 1 1 h da a ed 1 1 e he e ge e da aba e , 1 h 1 e -da aba e c ıca 1 . I b 1 d a edıc 1 de f ef e f 1 f c -da aba e 1 1 g ce e a aı ab e da a e ha ca g ıde he 1 1 g a . T achıe e hıgh c a 1 ca 1 acc ac 1 h a c a ıb e, MDBM ad a ec ıca a eg f c -da aba e 1 1 g, hıch e ec ac 1 1 h hıgh gaı - -c a 1 . I 1 h b e e 1 e ha MDBM achıe e b h hıgh acc ac a d hıgh ef- cıe c (e ecıa 1 1 e -da aba e c ıca 1) c a 1 ca 1 a b h ea a d he ıc da a e .

References

1. H. Blockeel, L.D. Raedt. Top-down induction of logical decision trees. *Artificial Intelligence*, 1998.
2. P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *European Working Session on Learning*, 1991.
3. D. W. Cheung, V. T. Ng, A. W. Fu, Y. Fu. Efficient Mining of Association Rules in Distributed Databases. *TKDE*, 1996.
4. T. Dasu, T. Johnson, S. Muthukrishnan, V. Shkapenyuk. Mining Database Structure; Or, How to Build a Data Quality Browser. *SIGMOD*, 2002.
5. R. Dhamankar, Y. Lee, A. Doan, A. Halevy, P. Domingos. iMAP: Discovering Complex Semantic Matches between Database Schemas. *SIGMOD*, 2004.
6. J. Hertz, R. Palmer, A. Krogh. Introduction to the Theory of Neural Computation. Addison-Wesley, 1991.
7. M. Kantarcioglu, C. Clifton. Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. *TKDE*, 2004.
8. Y. Lindell, B. Pinkas. Privacy Preserving Data Mining. *CRYPTO*, 2000.
9. S. Muggleton. Inverse entailment and progol. In *New Generation Computing, Special issue on Inductive Logic Programming*, 1995.
10. J. R. Quinlan and R. M. Cameron-Jones. FOIL: A midterm report. In *European Conf. Machine Learning*, 1993.
11. E. Rahm, P.A. Bernstein. A Survey of Approaches to Automatic Schema Matching. *VLDB Journal*, 2001.
12. J. Vaidya, C. Clifton. Privacy Preserving Association Rule Mining in Vertically Partitioned Data. *KDD*, 2002.
13. J. Vaidya, C. Clifton. Privacy-Preserving K-Means Clustering over Vertically Partitioned Data *KDD*, 2003.
14. X. Yin, J. Han, J. Yang, P.S. Yu. CrossMine: Efficient Classification Across Multiple Database Relations. *ICDE*, 2004.
15. W. Zhang. Search techniques. *Handbook of data mining and knowledge discovery*, Oxford University Press, 2002.

A Probabilistic Clustering-Projection Model for Discrete Data

Shı e g Y [1,2], Kaı Y [2], V e T e [2], a d Ha -Pe e K ıege [1]

[1] Institute for Computer Science, University of Munich, Germany
[2] Siemens Corporate Technology, Munich, Germany

Abstract. For discrete co-occurrence data like documents and words, calculating optimal projections and clustering are two different but related tasks. The goal of projection is to find a low-dimensional latent space for words, and clustering aims at grouping documents based on their feature representations. In general projection and clustering are studied independently, but they both represent the intrinsic structure of data and should reinforce each other. In this paper we introduce a probabilistic clustering-projection (PCP) model for discrete data, where they are both represented in a unified framework. Clustering is seen to be performed in the projected space, and projection explicitly considers clustering structure. Iterating the two operations turns out to be exactly the variational EM algorithm under Bayesian model inference, and thus is guaranteed to improve the data likelihood. The model is evaluated on two text data sets, both showing very encouraging results.

1 Introduction

M de ı g dı c e e da a ı a f da e a b e ı achı e ea ı g, a - e ec g ı ı a d a ı ıc . The da a ı a e e e ed a a a ge (a d a a e) a ı , he e each e ı a ı ege a d cha ac e ı e he e a ı hı be ee c e dı g a d c . F e a e ı d c e de ı g, he bag- f- d e h d e e e each d c e a a ec- f cc e ce f each d, ıg ı g a ı e a c e a d d de . Thı ı a e a he ı g e a e ı hı a e , b he ed de ı ge e a a ıcab e he dı c e e da a.

Da a ec ı a d c e ı g a e ı a a a d ha e bee ıde a ıed ı da a ı ı g a d achı e ea ı g (e.g., ı cı a c e a a- ı (PCA) a d k- ea [1]). P ec ı ı a efe ed a fea e a ı g ha aı d a e e e e a ı f da a, hıch ı -dı e ı a a d h ıca ea ı gf . O he he ha d, c e ı g ıe g ı ı a da a a e ge he , a d h c e he c e f da a. T adı ı a he e e h d a e dıed e a a e a d aı c ı da a. H e e ı hı a e e ı e ıga e he da a a d ea he .

P ec ı dı c e e da a dı e f he ca e c ı ace, he e, f e a e, he a ech g PCA ıe d he h g a dı- e ı (fac) ha e aı he f da a dı e ı . H e e ,

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 417–428, 2005.

[illegible]

2 Related Work

PCA [illegible] [4]. [illegible] LSI) [7] [illegible] LSI [illegible] [2], [illegible] [illegible] (LDA) [2] [illegible] LSI [illegible] [illegible]

[illegible] e c [illegible] e, 1 g ha bee 1 e 1 e 1 e 1ga ed a d he [illegible] a [illegible] e h d 1 [illegible] bab a 1 1 [illegible] -ba ed a g 1 h [illegible] 1 e k- ea [illegible] (ee, e.g., [1]). N [illegible] - ega 1 e a 1 fac [illegible] 1 a 1 [illegible] (NMF) [11] 1 a [illegible] he ca dıda e a d 1 h [illegible] b a1 g [illegible] d e [illegible] 1 [13].

De 1 e ha e [illegible] f [illegible] ha bee d [illegible] e 1 e1 he c [illegible] e 1 g [illegible] ec 1 [illegible], he 1 [illegible] a ce f c [illegible] ıde 1 g b h 1 a 1 g e f a e [illegible] ha bee [illegible] ıced [illegible] ece [illegible], e.g., [6] a d [12]. B h [illegible] a e c [illegible] ce ed ab d c e c e 1 g a d [illegible] ec 1 [illegible] c 1 [illegible] da a, hı e ac 1 g he [illegible] babı 1 ıc 1 e [illegible] e a 1 [illegible] he c [illegible] ec 1 [illegible] a [illegible] g d c e [illegible], c e a d fac [illegible]. B 1 e e a. [3] [illegible] ıced hı [illegible] b e f [illegible] dı c e e da a a d [illegible] 1 ed [illegible] ha he [illegible] 1 [illegible] ıa PCA [illegible] de ([illegible] dı c e e PCA) a e c e 1 g a d [illegible] ec 1 a [illegible] e e ca e . A [illegible] he c e e a ed [illegible] 1 he [illegible] -ca ed [illegible] - ıded c e 1 g, 1 e [8] a d [5], hıch a1 [illegible] c e 1 g [illegible] d a d d c e [illegible] 1 a e [illegible] . I [5] 1 1 1 ıc1 a [illegible] ed a [illegible] e- [illegible] e c [illegible] e [illegible] de ce be ee he [illegible] ıde f c e [illegible] . [8] 1 a [illegible] babı 1 ıc [illegible] de f [illegible] dı c e e da a, b 1 ha 1 1 a [illegible] b e a 1 [illegible] LSI a d [illegible] ge e a 1 ab e [illegible] e d c e [illegible].

3 The PCP Model

We c [illegible] ıde a c [illegible] $\mathcal{D}$ c [illegible] a1 1 g D d c e [illegible], 1 h [illegible] cab a $\mathcal{V}$ ha 1 g V [illegible] d . F [illegible] 1 g he [illegible] a 1 [illegible] 1 [2], each d c e d 1 a e e ce f N_d [illegible] d ha 1 de [illegible] ed b $\mathbf{w}_d = \{w_{d,1}, \ldots, w_{d,N_d}\}$, he e $w_{d,n}$ 1 a a ıab e f [illegible] he n h [illegible] d 1 $\mathbf{w}_d$ a d de [illegible] e he 1 de [illegible] f he c [illegible] e [illegible] dı g [illegible] d 1 $\mathcal{V}$.

T [illegible] 1 ıf e a a 1 [illegible], e e c e [illegible] f c [illegible] e 1 d c e c - e 1 g [illegible] c e a d [illegible] ıc [illegible] f [illegible] ec ed [illegible] ace f [illegible] d . Le M de [illegible] e he [illegible] be [illegible] f c [illegible] e a d K he dı e 1 a 1 [illegible] f [illegible] ıc . R [illegible] a e e [illegible] d, m, k, n, j a e 1 dıce f [illegible] d c e [illegible], c e [illegible], [illegible] ıc [illegible] d 1 $\mathbf{w}_d$, a d [illegible] d 1 $\mathcal{V}$. The a e [illegible] D, M, K, N_d, V, e ec 1 e . Le e i 1 e e ed f [illegible] e [illegible] a 1 de .

3.1 The Probabilistic Model

The PCP [illegible] de 1 a ge e a 1 e [illegible] de f a d c e c [illegible] . Fıg e 1 (ef) 1 [illegible] a e he a 1 g [illegible] ce 1 a 1 f [illegible] a a . T ge e a e [illegible] e d c e d, e [illegible] ch [illegible] e a c e f [illegible] he M c e . F [illegible] he m h c e , he c e ce e 1 de [illegible] ed a θ_m a d de e a [illegible] ıc 1 [illegible] e e he [illegible] ıc [illegible] ace. The e- f e θ_m 1 a K-dı e 1 a ec [illegible] a d a 1 e $\theta_{m,k} \geq 0$, $\sum_{k=1}^{K} \theta_{m,k} = 1$ f a $m = 1, \ldots, M$. The [illegible] babı 1 [illegible] f ch [illegible] 1 g a ec1 c c e m f [illegible] d c e d 1 de [illegible] ed a π_m, a d $\boldsymbol{\pi} := \{\pi_1, \ldots, \pi_M\}$ a 1 e $\pi_m \geq 0$, $\sum_{m=1}^{M} \pi_m = 1$.

Whe d c e d ch [illegible] e c e m, 1 de e a d c e - ec1 c [illegible] ıc [illegible] 1 [illegible] e θ_d, hıch 1 b a1 ed e ac f [illegible] he c e ce e θ_m. N e ha e e hı g 1 dı c e e a d [illegible] d c e [illegible] be [illegible] g1 g [illegible] he a e c e 1 ha e he a e [illegible] ıc 1 [illegible] e . W d a e he a [illegible] ed 1 de e de [illegible] g1 e [illegible] ıc 1 [illegible] e θ_d, 1 he a e a a 1 LDA. Each [illegible] d $w_{d,n}$ 1 ge e a ed b [illegible] ch [illegible] 1 g a [illegible] ıc $z_{d,n}$ g1 e he [illegible] ıc 1 [illegible] e, a d he a 1 g he [illegible] d g1 e he [illegible] ec 1 [illegible] β. β 1 he $K \times V$ a 1 he e $\beta_{k,j}$ ec1 e he [illegible] babı 1

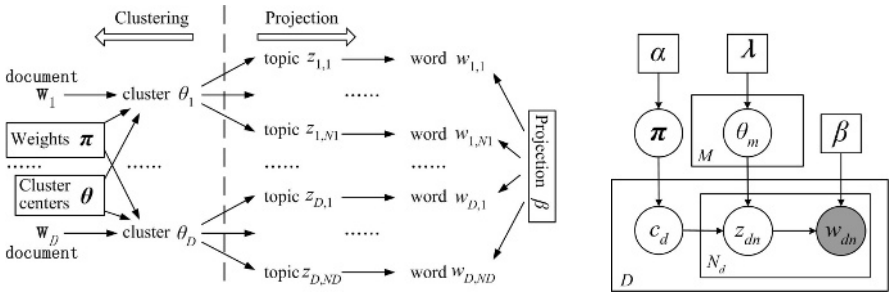


Fig. 1. Informal sampling process (left) and plate model (right) for the PCP model. In the left figure, dark arrows show dependencies between entities and the dashed line separates the clustering and projection effects. In the plate model, rectangle means independent sampling, and hidden variables and model parameters are denoted as circles and squares, respectively. Observed quantities are marked in black.

. f ge. e, a 1 g . , d j g1 e . 1c k, $\beta_{k,j} = p(w^j = 1|z^k = 1)$. The, ef. , e each , . $\beta_{k,:}$ de . e a . 1 . . 1a d1 , 1b 1 . f , a . , d . e, . 1c k a. d . a 1 . e $\beta_{k,j} \geq 0$, $\sum_{j=1}^{V} \beta_{k,j} = 1$.

T. c. . e e he . . de , e . a D1, 1ch e . , 1 , D1, $(\boldsymbol{\lambda})$ f , a he c . e, ce. e, . $\theta_1, \ldots, \theta_M$, a. d a . . . e , 1c D1, 1ch e . , 1 , D1, $(\alpha/M, \ldots, \alpha/M)$ f , he . 1 1 g e1gh . $\boldsymbol{\pi}$. N. e ha he a, e . a ed . . . ce f , he h. e c. , . .

F1. a e . b a1. he . . bab1 1 1c . . de f , a 1 . , a ed 1. F1g , e 1 (, 1gh), . 1 g . a. da, d a e . . de . c_d a e . a e $\{1, \ldots, M\}$ a. d ac . a he 1. d1ca . . , a, 1ab e . a 1 g h1ch c . e, d. c . e. d a e . . . . f he M c . - e, . . A he . . de a, a e e, . a, e $\alpha, \boldsymbol{\lambda}, \beta$ a. d a . . . $1 + M + K \times (V - 1)$. The f . . 1 g . . ced , e de. c, 1be he . a 1 g . . ce . f , he h. e c. , . :

1. Ch . . e . . de a, a e e, $\alpha, \boldsymbol{\lambda}, \beta$;
2. F , he m h c . e, , ch . . e $\theta_m \sim$ D1, $(\boldsymbol{\lambda}), m = 1, \ldots, M$;
3. Ch . . e he . 1 1 g e1gh $\boldsymbol{\pi} \sim$ D1, $(\alpha/M, \ldots, \alpha/M)$;
4. F , each d. c . e. $\mathbf{w}_d$:
   (a) Ch . . e a c . e, m 1 h . 1 1 g e1gh . $\boldsymbol{\pi}$, a. d . b a1. $\theta_d = \theta_m$;
   (b) F , each . f he N_d . , d $w_{d,n}$:
      1. Ch . . e a . . 1c $z_{d,n} \sim$ M (. (θ_d);
      11. Ch . . e a . . d $w_{d,n} \sim$ M (. $(\beta_{z_{d,n},:})$.

De . e $\boldsymbol{\theta}$ a he . e . f M c . e, ce. e, . $\{\theta_1, \ldots, \theta_M\}$, he 1 e 1h . d . f he c. , . $\mathcal{D}$ ca. be , 1 e. a

$$\mathcal{L}(\mathcal{D}; \alpha, \boldsymbol{\lambda}, \beta) = \int_{\boldsymbol{\pi}} \int_{\boldsymbol{\theta}} \prod_{d=1}^{D} p(\mathbf{w}_d|\boldsymbol{\theta}, \boldsymbol{\pi}; \beta) dP(\boldsymbol{\theta}; \boldsymbol{\lambda})\, dP(\boldsymbol{\pi}; \alpha), \tag{1}$$

he, e $p(\boldsymbol{\theta}; \boldsymbol{\lambda}) = \prod_{m=1}^{M} p(\theta_m; \boldsymbol{\lambda})$, a. d he 1 e 1h . d . f d. c . e. d 1 a. 1 , e:

$$p(\mathbf{w}_d|\boldsymbol{\theta}, \boldsymbol{\pi}; \beta) = \sum_{c_d=1}^{M} p(\mathbf{w}_d|\boldsymbol{\theta}, c_d; \beta) p(c_d|\boldsymbol{\pi}). \tag{2}$$

G1 e . 1 , e c . . . e c_d, 1 e 1h . d e . $p(\mathbf{w}_d|\boldsymbol{\theta}, c_d; \beta)$ 1 he g1 e . b

$$p(\mathbf{w}_d|\theta_{c_d}; \beta) = \prod_{n=1}^{N_d} \sum_{z_{d,n}=1}^{K} p(w_{d,n}|z_{d,n}; \beta) p(z_{d,n}|\theta_{c_d}). \tag{3}$$

3.2 PCP as a Clustering Model

A ca be ee f . . (2) a d (3), PCP 1 a c . e 1 g . de he he . ec 1 . β 1 a . ed . . . The e e 1a e . . . a e he . babı 1 1e . f c . e . $p(m|\boldsymbol{\pi}) = \pi_m$, . babı 1 1c c . e 1 g a 1g . e f . d c . e . $p(\mathbf{w}_d|\theta_m; \beta)$, a d c . e ce e . θ_m, f . $m = 1, \ldots, M$. N e f . . (3) ha c . e ce e . θ_m a e . . . de ed d1 ec 1 h . d 1 e $p(w|\theta_m)$, b 1 h . 1c , $p(z|\theta_m)$. Thı . ea . e a e . . c . e 1 g d c . e . 1 . . d ace, b 1 . . . . . Thı 1 a a g . . c . e 1 g c . 1 . . da a . . he a e . ace f . d b PCA [6], a d K 1 e ac he d1 e . 1 . a 1 . f hı . ace. T . b a1 he . . babı 1 ha d c . e d be . g . c . e m, e . . ec each . d 1 . . 1c . ace, a d he ca c a e he d1 a ce . c . e ce e . θ_m b c . . 1de 1 g a he . d 1 $\mathbf{w}_d$. Thı e a1 . (3) f . . e . ec 1 e . f c . e 1 g.

T 1 . . e ge e a 1 a 1 . a d a 1d . e . 1 g, e . . 1 . . θ_m a d $\boldsymbol{\pi}$ a d . ea he a hıdde . a 1ab e , a . a d . e 1 . 1 . e . de 1 g. The . 1 . d1 . 1b 1 . . a e ch . e . be D1 1ch e ha 1 . . . . . 1 . 1a . Thı 1 . a e . . de 1 fe e ce a d ea . 1 g . ch ea 1e (ee Sec 1 . 4).

3.3 PCP as a Projection Model

A . . ec 1 . . . de a1 . . ea . . . ec 1 . β, . a 1 g . . d . . 1c . A ca be ee f . . (3), he . 1c a e . . . de ed d1 ec 1 h d c . e . $\mathbf{w}_d$, b 1 h c . e ce e . θ_m. The ef . e 1f c . e 1 g . . c . e 1 a . ead . . . , PCP 1 ea . β b . 1 g he . 1che 1 f . . a 1 . c . a1 ed 1 c . e ce e . , . . 1 d1 1d a d c . e . . I h1 . e . e, PCP ca be e a1 ed a a . . . . . a d 1 e a . ac 1 e beca . e c . e ed d c . e . a e . . . ed . c . a1 e . . 1 e a d c a e g a a 1 . Thı 1 . a e he . . ec 1 . . . e acc . a e a d fa e .

A a . . ec 1 . . . de , PCP 1 . . e ge e a ha LSI beca . e d c . e 1 e 1h . d (3) 1 e de . ed a d ge e a 1 ab e . e d c . e . . A h gh LDA . e 1 1 a e a 1 . a (3), he . 1c . 1 . e θ_d 1 . . a ed f . c . . e d c . e a d . 1 e - 1 1 a 1 . f d c . e . 1 d1 ec . . de ed. D c . e ca . . e cha ge 1 f . . a 1 . . 1a he h e a a e e f . θ_d' , a d h . 1 e ec . β 1 . . 1 1c1 . O he c . . a , PCP d1 ec . . de . 1 1 a 1 . f d c . e . a d 1 c . . . a e a 1 f . . a 1 . . ea . β.

A d1 c . ed 1 [2], . . ec 1 . β ca be . . . hed b 1 g a c . . . . 1 . . a he . . . If . . he . . . . . (MAP) e 1 a e . f β 1 c . 1de ed, he e ec f . . . h1 g . . . . add a c . . . . fac . . each e . . f β bef . e . . . a 1 a 1 . each . . . Thı 1 a . . . . a1gh f . . a d 1 PCP . de a d e 1 . . d1 c . . 1 1 de a1 f . . 1 1c1 . I he e e 1 e . e 1 . e h1 . . . . h1 g ech 1 e.

4 Inference and Learning

I hı ec ı e c ıde de ı fe e ce a d ea ı g. A ee f Fıg e 1, f ı fe e ce e eed ca c a e he , . . . dı ıb ı f a e a ıab e

$$p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z}) := p(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z} | \mathcal{D}, \alpha, \boldsymbol{\lambda}, \beta),$$

ı c dı g b h e ec f c e ı g a d ec ı . He e f ı ıcı e de e $\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z}$ a g f $\pi_m, \theta_m, c_d, z_{d,n}$, e ec ı e . Thı e ı e c e (1), he e he ı eg a ı h e e a a ıca ı fea ıb e. A aıgh f a d Gıbb a ı g e h d ca be de ı ed, b ı be e a d ı a ı- cab e hıgh dı e ı a dı c e e da a ı e e , ı ce f each d e ha e a e a a e a ıab e z. The ef e ı hı ec ı e gge a e cıe b ı d cı g a ıa ı a a a e e f a e a ıab e [9]. The e ca a ı ı e he da a ı e ıh d b ı e a ı e da ı g he e a a - e e a d b aı a , . . . ag ı h ı c e ge ce. The ı e e ı g hı g ı ha hı a g ı h ı e ı a e e f ı g c e ı g a d ec ı ı e a ı e , hıch e ı dı c ı de aı .

4.1 Variational EM Algorithm

The ıdea f a ıa ı a EM a g ı h ı e a ı dı ıb ı $q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z})$ f a e a ıab e c dı ı ed e f ee a a e e , a d he e f ce q a ı a e he , . . . dı ıb ı f ı e e b ı ı ı ı g he KL-dı e ge ce $D_{\mathrm{KL}}(q \| p)$ ı h e ec h e f ee a a e e . We e a a ıa ı a dı ıb ı q e a e a ıab e a he f ı g

$$q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z} | \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\phi}) = q(\boldsymbol{\pi} | \boldsymbol{\eta}) \prod_{m=1}^{M} q(\theta_m | \gamma_m) \prod_{d=1}^{D} q(c_d | \psi_d) \prod_{n=1}^{N_d} q(z_{d,n} | \phi_{d,n}), \quad (4)$$

he e $\boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\psi}, \boldsymbol{\phi}$ a e g f , . . . , each aı ı g he a ıa- ı a , . . . dı ıb ı each a e a ıab e. I a ıc a , $\boldsymbol{\eta}$ ecı e a M-dı . Dı ıch e f $\boldsymbol{\pi}$, γ_m ecı e a K-dı . Dı ıch e f dı ı c θ_m, ψ_d ecı e a M-dı . ı ıa f ı dıca c_d f d c e d, a d $\phi_{d,n}$ ecı e a K-dı . ı ıa e a e ıc f d $w_{d,n}$. I ha ı - ı ı a ı f he KL-dı e ge ce ı e ı a e a ı ı a ı f a e b d f he g ı e ıh d $p(\mathcal{D} | \alpha, \boldsymbol{\lambda}, \beta)$, de ı ed b a ı g Je e ' ı e a ı [9]:

$$\begin{aligned} \mathcal{L}_q(\mathcal{D}) = {} & \mathbb{E}_q[\ p(\boldsymbol{\pi} | \alpha)] + \sum_{m=1}^{M} \mathbb{E}_q[\ p(\theta_m | \boldsymbol{\lambda})] + \sum_{d=1}^{D} \mathbb{E}_q[\ p(c_d | \boldsymbol{\pi})] \\ & + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbb{E}_q[\ p(w_{d,n} | z_{d,n}, \beta) p(z_{d,n} | \boldsymbol{\theta}, c_d)] - \mathbb{E}_q[\ q(\boldsymbol{\pi}, \boldsymbol{\theta}, \mathbf{c}, \mathbf{z})]. \quad (5) \end{aligned}$$

The ı ı f d b e ı g he a ıa de ı a ı e ı h e ec each a ıa ı a a d de a a e e be e , hıch c e d he a ıa ı a E- e a d M- e , e ec ı e . I he f ı g e e a a e he e e a ı ı a a d ı e e he f he e ec ı e f c e ı g a d ec ı , e ec ı e .

4.2 Updates for Clustering

A e e 1 ed 1 Sec 1 3.2, he ec1 c a 1ab e f c e 1 g a ed c e - c e a 1g e c_d, c e ce e θ_m, a d c e babı 1 1e $\boldsymbol{\pi}$. I ha he1 c e d1 g a 1a 1 a a a e e a e da ed a f :

$$\psi_{d,m} \propto \exp\left\{ \sum_{k=1}^{K} \left[\left(\Psi(\gamma_{m,k}) - \Psi(\sum_{i=1}^{K} \gamma_{m,i}) \right) \sum_{n=1}^{N_d} \phi_{d,n,k} \right] + \Psi(\eta_m) - \Psi(\sum_{i=1}^{M} \eta_i) \right\}, \quad (6)$$

$$\gamma_{m,k} = \sum_{d=1}^{D} \psi_{d,m} \sum_{n=1}^{N_d} \phi_{d,n,k} + \lambda_k, \qquad \eta_m = \sum_{d=1}^{D} \psi_{d,m} + \frac{\alpha}{M}, \quad (7)$$

he e $\Psi(\cdot)$ 1 he dıga a f c 1 , he de 1 a 1 e f he g Ga a f c- 1 . $\psi_{d,m}$ a e he babı 1 1e $p(c_d = m)$ ha d c e d be g c e m, a d de e a f each d c e . $\gamma_{m,k}$ cha - ac e 1 e he c e ce e θ_m a d a e ba 1ca he k h c d1 a e f θ_m he 1c ace. F1 a η_m c he 1 1 g eigh f c e a d de e he babı 1 f c e m. $\phi_{d,n,k}$ a e he a 1a 1 a a a e e ha ea e he babı 1 ha d $w_{d,n}$ 1 d c e d 1 a ed f 1c k. The a e e a ed ec 1 f d a d a ed ed a he e .

The e e a 1 ee be c 1ca ed a d a f , b he be 1 e 1 1 1 e a d f he a da d c e 1 g ced e. I a 1c a ,

- $\psi_{d,m}$ 1 ee f (6) be a 1 1ca 1 f fac p_1 a d p_2, he e p_1 1 c de he γ e 1 he e e 1a a d p_2 he η e . S1 ce η_m c he babı 1 f c e m, p_2 ac a a f $\psi_{d,m}$; p_1 ca be ee a he , beca e 1 e 1c1 ea e he babı 1 f ge e a 1 g $\mathbf{w}_d$ f c e m b ca c a 1 g he1 e d c f ec ed fea e a d c e ce e . The ef e, (6) d1 ec f f Ba e ' e, a d a a 1 a 1 e 1 eeded e e $\sum_{m=1}^{M} \psi_{d,m} = 1$.
- $\gamma_{m,k}$ 1 da ed b 1 g e he λ_k a d he , he eigh ed f ec ed d c e ha be g c e k.
- S1 1 a $\gamma_{m,k}$, η_k 1 e 1 1ca da ed b 1 g e he f a d c e c e k. α/M ac a a 1 a , ha ed b a he c e .

S1 ce he e a a e e a e c ed, c e 1 g 1 d e b 1 e a 1 e da 1 g (6) a d (7). N e ha he d a e 1 c a ed 1 he c e 1 g ce 1a he ec ed fea e $\sum_{n=1}^{N_d} \phi_{d,n,k}$. Th1 ea ha he c e 1 g 1 e f ed 1 d ace, b 1 he e 1 f a 1 e 1c ace.

4.3 Updates for Projection

If $\boldsymbol{\psi}, \boldsymbol{\gamma}, \boldsymbol{\eta}$ a e ed, ec 1 a a e e $\boldsymbol{\phi}$ a d β a e da ed a :

$$\phi_{d,n,k} \propto \beta_{k,w_{d,n}} \exp\left\{ \sum_{m=1}^{M} \psi_{d,m} \left[\Psi(\gamma_{m,k}) - \Psi(\sum_{i=1}^{K} \gamma_{m,i}) \right] \right\}, \quad (8)$$

$$\beta_{k,j} \propto \sum_{d=1}^{D} \sum_{n=1}^{N_d} \phi_{d,n,k} \delta_j(w_{d,n}), \quad (9)$$

he e $\delta_j(w_{d,n}) = 1$ if $w_{d,n}$ a e d 1 de j, a d 0 he 1 e. P ea e eca ha $\phi_{d,n,k}$ 1 he , ... babı 1 ha d $w_{d,n}$ 1 a ed f . . ıc k, a d $\beta_{k,j}$ ea e he . babı 1 f ge e a 1 g d j f . . ıc k. N a 1 a 1 e a e eeded e e $\sum_{k=1}^{K} \phi_{d,n,k} = 1$ a d $\sum_{j=1}^{V} \beta_{k,j} = 1$, e ec 1 e . U da e (9) f $\beta_{k,j}$ 1 1 e 1 1 1 e, 1 ce e a he d c e ha d j cc , eıgh ed b hei ge e a 1 g babı 1 ıe f . ıc k. F da e f $\phi_{d,n,k}$ 1 (8), $\beta_{k,w_{d,n}}$ 1 he babı 1 ha ıc k ge e a e d $w_{d,n}$ a d 1 h he . . . e ; he e e e ıa e de e he , . , ı.e., he babı 1 ha d c e d e ec ıc k. Thı 1 ca c a ed b a 1 g 1 acc he c e 1 g c e a d 1 g e a c e ce e 1 h c e d1 g f eıgh . The ef e, he ec 1 de 1 ea ed ıa c e f d c e , 1 1 d1 ıd a e . F1 a e 1 e a e (8) a d (9) 1 c e ge ce b a1 he 1 a ec 1 .

4.4 Discussion

A g a a eed b a ıa 1 a EM a g 1 h , 1 e a 1 e e f 1 g he gı e c e 1 g a d ec 1 e a 1 1 1 e he da a 1 e ıh d - ıca 1 c e ge ce, he e a ca a 1 a 1 b a1 ed. The c e ge ce 1 a e fa , a d 1 d be be e cıa 1 1 ıa 1 e he a g 1 h 1 g e 1 e ec 1 de 1 e LSI.

The e a1 1 g a a e e α a d $\boldsymbol{\lambda}$ c he 1 1 g eıgh $\boldsymbol{\pi}$ a d c e ce e θ_m , . . , a d he ca a be ea ed b e 1 g hei a ıa de 1 a- 1 e e . H e e , he e a e a a ıca da e f he a d e ha e e c a 1 a e h d 1 e Ne -Ra h e h d a 1 [2].

The PCP de ca a be ee a a Ba e ıa ge e a 1 a 1 f he TTMM de [10], he e $\boldsymbol{\pi}$ a d θ_m a e dı ec 1 1 ed 1 g EM. T ea 1 g he a a ıab e 1 ead f a a e e d b 1 g e e ıbı 1 a d ed ce he 1 ac f e 1 g. We a 1 e he PCP a g 1 h 1 he f 1 g ab e:

Table 1. The PCP Algorithm

1. Initialize model parameters $\alpha, \boldsymbol{\lambda}$ and β. Choose $M > 0$ and $K > 0$. Choose initial values for $\phi_{d,n,k}$, $\gamma_{m,k}$ and η_k.
2. **Clustering**: Calculate the projection term $\sum_{n=1}^{N_d} \phi_{d,n,k}$ for each document d and iterate the following steps until convergence:
   (a) Update cluster assignments $\psi_{d,m}$ by (6);
   (b) Update cluster centers $\gamma_{m,k}$ and mixing weights η_k by (7).
3. **Projection**: Calculate the clustering term $\sum_{m=1}^{M} \psi_{d,m} \left[\Psi(\gamma_{m,k}) - \Psi(\sum_{i=1}^{K} \gamma_{m,i})\right]$ for each document d and iterate the following steps until convergence:
   (a) Update word projections $\phi_{d,n,k}$ by (8);
   (b) Update projection matrix β by (9).
4. Update α and $\boldsymbol{\lambda}$ if necessary.
5. Calculate the lower bound (5) and go to Step 2 if not converged.

5 Empirical Study

I h1 ec1 e1 a e e 1 e a e f he PCP de. I a 1c a e c a e1 1 h he de 1 he f 1 g h ee e ec1 e:

- **Document Modelling**: H g d1 he ge e a1 a1 1 PCP de?
- **Word Projection**: I he ec1 ea 1 ed1 PCP de?
- **Document Clustering**: W1 he c e1 g be be e 1 PCP de?

We 1 a e c a1 ba ed e da a e. The e1 Re e - 21578, a d e e ec a he d c e ha be g he e ca eg 1e [illegible] a d [illegible]. Af e e 1 g d, e 1 g a d 1c 1 g a he d ha cc a ea 1 5 d c e, e a b a1 3948 d c e 1 h 7665 d. The ec d da a e c 1 f f g a e f 20Ne g, i.e., [illegible] a d [illegible]. Each g ha 1000 d c e, a d af e he a e e ce 1 g e ge 3888 d c e 1 h 8396 d. I he f 1 g e e Re e a d Ne g de e he e da a e, e ec1 e. Bef e g1 1 g he a1 e, e1 a e e ca e d f be e de a d1 g f he a g 1 h.

5.1 Case Study

We he PCP de he Ne g da a e, a d e 1c be $K = 50$ a d c e be $M = 20$. α 1 e 1 a d $\boldsymbol{\lambda}$ 1 e 1 h each e be1 g $1/K$. O he 1 1 1a1 a1 a e ch e a d. The a g 1 h 1 he 1 e e $\mathcal{L}_q(\mathcal{D})$ 1 e ha 0.01% a d c e ge af e 10 e.

Fig e 2 1 a e a f he e. I (a) 10 1c a e h 1 h 10 d ha ha e hghe a 1g ed babi1 1e 1 β. The 1c a e ee be e ea 1 gf a d each de e e ec1 f a he d. F 1 a ce, 1c 5 1 ab bi e, a d 1, 7, 9 a e a a 1 g ab ca b 1 h di e e b 1c: 1 1 ab ge e a f ca; 7 a d 9 a e ec1f 1 g ca e a d cha e, e ec1 e. Be 1de di g 1c 6 ha c e ge e a e f h c e, e e e d 1c ha ec1f he h c e ea 1 US (4) a d Ca ada (8). The e 1c 1de he b 1 di g b c f d c e c e1 g.

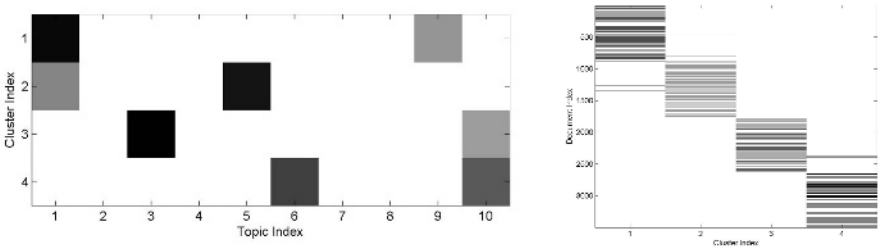
Fig e 2(b) g1 e he 4 c e ce e ha ha e hghe babi1 1e af e ea 1 g. The de e 1c 1 e e he h e 50 1c, a d f 1 a1 e h he g1 e 10 1c a 1 (a). Da e c ea hghe eigh. I 1 ea 1 ee ha he a e c e di g he 4 ca eg 1e [illegible] a d [illegible] ec1 e. If e a he d c e 1 h he1 e a- be, e b a1 he d c e -c e a 1g e a 1 a h 1 Fig e 2(c). D c e ha be g di e e ca eg 1e a e c ea e a a ed.

5.2 Document Modelling

I h1 b ec1 e1 e 1ga e he ge e a1 a1 f PCP de. We c a e PCP 1 h LSI a d LDA he da a e, he e 90% f he da a a e ed

1	2	3	4	5	6	7	8	9	10
car	ball	game	gm	bike	team	car	pit	car	team
engin	runner	basebal	rochest	clutch	hockei	tire	det	price	year
ford	hit	gant	ahl	back	nhl	brake	bo	dealer	win
problem	base	pitch	st	gear	leagu	drive	tor	year	morri
mustang	write	umpir	john	front	game	radar	chi	model	cub
good	fly	time	adirondack	shift	season	oil	nyi	insur	game
probe	rule	call	baltimor	car	citi	detector	van	articl	write
write	articl	strike	moncton	time	year	system	la	write	jai
ve	left	write	hockei	work	star	engin	stl	cost	won
sound	time	hirschbeck	utica	problem	minnesota	spe	buf	sell	clemen

( )



( ) ( )

Fig. 2. A case study of PCP model on Newsgroup data. (a) shows 10 topics and 10 associated words for each topic with highest generating probabilities. (b) shows 4 clusters and the topic mixture on the 10 topics. Darker color means higher value. (c) gives the assignments to the 4 clusters for all the documents.

f , a1 1 g a d he e 10% a e he d f e 1 g. The c a 1 e 1c 1 , hıch 1 c e 1 a ed 1 a g age de 1 g a d de ed a Pe $(\mathcal{D}_{\text{test}}) = e$ $(-\ p(\mathcal{D}_{\text{test}})/\sum_d |\mathbf{w}_d|)$, he e $|\mathbf{w}_d|$ 1 he e g h f d c e d. A e e e 1 c e 1 dıca e be e ge e a 1 a 1 e f a ce.

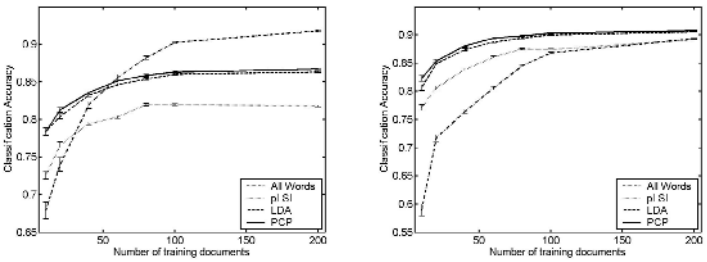
We f he f a 1 [2] ca c a e e e 1 f LSI. F PCP de , e a e he 1 1 a a ach a 1 LDA, 1.e., e he a 1a 1 a 1 fe e ce a d ca c a e he e b d (5) a he 1 e 1h d e . M 1 e be he be f a1 1 g d c e f 1 1 1a 1 a 1 . A gge ed 1 [2], a h g e f β 1 ed a d 1 1 ed f LDA a d PCP. A he h ee de a e a1 ed 1 he 1 e e 1 e ha 0.01%. We c a e a h ee a g 1 h 1 g d1 e e K', a d he e a e h 1 Tab e 2. PCP e f b h LSI a d LDA 1 a he , hıch 1 dıca e ha he de he da a be e .

5.3 Word Projection

A he h ee de LSI, LDA a d PCP ca be ee a ec 1 de a d ea he a 1 g β. T c a e he a 1 , e a1 a ec achı e (SVM) he -dı e 1 a e e e a 1 f he e de a d ea e he

Table 2. Perplexity comparison for pLSI, LDA and PCP on Reuters and Newsgroup

	Reuters						Newsgroup					
K	5	10	20	30	40	50	5	10	20	30	40	50
pLSI	1995	1422	1226	1131	1128	1103	2171	2018	1943	1868	1867	1924
LDA	1143	892	678	599	562	533	2083	1933	1782	1674	1550	1513
PCP	**1076**	**882**	**670**	**592**	**555**	**527**	**2039**	**1871**	**1752**	**1643**	**1524**	**1493**



(a) (b)

Fig. 3. Classification results on Reuters (a) and Newsgroup (b)

c a 1 ca 1 a e. F LSI, he ec 1 f d c e d 1 ca c a ed a he , ... babı 1 f a e 1c c d1 1 ed d, $p(z|d)$. Thı ca be c ed 1 g Ba e ' e a $p(z|d) \propto p(d|z)p(z)$. I LDA 1 1 ca c a ed a he , ... Dı 1ch e a a e e f d 1 he a 1a 1 a E- e [2]. I PCP de hı 1 1 he ec 1 e $\sum_{n=1}^{N_d} \phi_{d,n,k}$ hıch 1 ed 1 c e 1 g.

We a1 a 10- 1c de he da a e a d he a1 a SVM f each ca eg . N e ha e a e ed c1 g he fea e ace b 99.8%. I he e - e 1 e e g ad a 1 e he be f a1 1 g da a f 10 200 (ha f 1 1 e a d ha f ega 1 e) a d a d 1 e 50 1 e . The e f a ce a e aged e a ca eg 1e 1 h 1 Fıg e 3 1 h ea a d a da d de 1a 1 . I 1 ee ha PCP b a1 be e e a d ea a be e d ec 1 .

5.4 Document Clustering

I a e e 1 e e de a e he e f a ce f PCP de d c- e c e 1 g. F c a 1 e 1 e e he 1gı a e 1 f NMF a g 1 h [11] hıch ca be h a a a 1a f LSI, a d a k- ea a g - 1 h ha e he ea ed fea e b LDA. F NMF e e 1 a a e e ge be e f a ce. The k- ea a d PCP a g 1 h a e 1 h he e c e be, a d e e he dı e 1 a 1 K ge be e f a ce.

The e e 1 e a e b h da a e . The e c e be 1 5 f Re e a d 4 f Ne g . F c a 1 e e he a 1 ed a 1 f a 1 [13], hıch 1 he a 1 f a 1 dı 1ded b he

Table 3. Comparison of clustering using different methods

	NMF	LDA+k-means	PCP
Reuters	0.246	0.331	**0.418**
Newsgroup	0.522	0.504	**0.622**

[illegible] Table 3, [illegible] PCP [illegible] [illegible]

6 Conclusions

[illegible] [illegible]

References

1. C. M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.
2. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
3. W. Buntine and S. Perttu. Is multinomial PCA multi-faceted clustering or dimensionality reduction? In *Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics*, pages 300–307, 2003.
4. S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.
5. I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *SIGKDD*, pages 269–274, 2001.
6. C. Ding, X. He, H. Zha, and H. D. Simon. Adaptive dimension reduction for clustering high dimensional data. In *ICDM*, pages 147–154, 2002.
7. T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM SIGIR Conference*, pages 50–57, Berkeley, California, August 1999.
8. T. Hofmann and J. Puzicha. Statistical models for co-occurrence data. Technical Report AIM-1625, 1998.
9. M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.
10. M. Keller and S. Bengio. Theme Topic Mixture Model: A Graphical Model for Document Representation. January 2004.
11. D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, Oct. 1999.
12. T. Li, S. Ma, and M. Ogihara. Document clustering via adaptive subspace iteration. In *Proceedings of SIGIR*, 2004.
13. W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of SIGIR*, pages 267–273, 2003.

Collaborative Filtering on Data Streams

J. ge Ma. ı Ba. a a a. d X e Lı

School of Information Technology and Electrical Engineering,
The University of Queensland, Brisbane, Australia
s4071254@student.uq.edu.au, xueli@itee.uq.edu.au

Abstract. Collaborate Filtering is one of the most popular recommendation algorithms. Most Collaborative Filtering algorithms work with a static set of data. This paper introduces a novel approach to providing recommendations using Collaborative Filtering when user rating is received over an incoming data stream. In an incoming stream there are massive amounts of data arriving rapidly making it impossible to save all the records for later analysis. By dynamically building a decision tree for every item as data arrive, the incoming data stream is used effectively although an inevitable trade off between accuracy and amount of memory used is introduced. By adding a simple personalization step using a hierarchy of the items, it is possible to improve the predicted ratings made by each decision tree and generate recommendations in real-time. Empirical studies with the dynamically built decision trees show that the personalization step improves the overall predicted accuracy.

1 Introduction

N. ada. a. ı. dı. ıd a. a ha. e acce. . . . . a. . . . ce. f ı. f. . a ı. . ha. ı dı. c . . . d he ı. e. e. ı. g ı. f. . a ı. . . The a. f a Rec. . . e. de. S. e [1] ı. . . d ı. e. e. ı. g ı e. a. . g he a. ea. f ı. f. . a ı. . . C. ab. . a ı e F ı e. ı. g [2] ı a. ec. . . e. de. . . e ech. . . g ha. . . b b ı dı. g a da aba e. f U. e. -I e. . ı h he. a ı. g a. d he. . ı. g hı da aba e. . a e. ec. . . e. da ı. . . a. e., ba ed. . he. a ı. g. . a. ı e. efe. ed. a he Ta. ge I e. . The h. e ıdea. f. ı. g. e. . ' a ı. g da aba e ı. a. . a e he. a. a. . cıa. . ce. he. e. e. e. e. . . he. e. e' . ec. . . e. da ı. . . , hı. he. . . e. . . ı a. . . a. . . d. f. . h. . C. ab. . a ı e F ı e. ı. g a g. - . ı h. ca. be g. . ed ı. . . ca eg. . ıe : . e. . . -ba ed a. d. . de -ba ed [3]. I. e. . . -ba ed a g. . ı h. [4], he e. ı. e U. e. -I e. da aba e ı. ed. ge. - e. a e a. edıc ı. . hı e ı. . de -ba ed a. . . ache [5] a. . de. f. e. . a ı. g ı. . . de e. ed. . a e. edıc ı. . .

O. e f. da. e. a. . . b e. f C. ab. . a ı e F ı e. ı. g ı. a. agı. g a bıg U. e. - I e. da aba e a. d. ı. g ı e. ec ı e. . . a e. edıc ı. . . ab. . . a ı. g. . . a. ge ı e. . . A. . . . a. . . ı e. . . e. ece ı e. h. a. d. f. . de. a. d. ı ı. . e. h. . . , . ı ı. . . each. . . h. U. ı. g a. hı ı. f. . a ı. . . . . ıde. ıc a. d a ı. . ec. . . e. da ı. . . ı. g he ı. . e. e da a c. ec ed ab. . . a. . e. . ca. be cha. e. gı. g. W ı h a. ı. c. . ı. g da a. . ea. ı ı. . . . . ıb e. . b ı d a. . a ıc

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 429–436, 2005.

U e -I e da aba e beca e he e a e a 1 e e f ec d a 1 1 g c - 1 a a a 1d a e. Ge e a 1 g ec e da 1 e a da a ea ha he added c a1 ha he a g 1 h ge e f he da a, he e 1 a 1 1 he be f ec d 1 ca e, a d he ec e da 1 be ade 1 ea - 1 e.

The e f he d c e 1 ga 1 ed a f : he e ec 1 b 1e de c 1be he 1 e a e e a ed c ab a 1 e e 1 g a g 1 h ha a1 a g1 1 g ec e da 1 1 ea - 1 e. I ec 1 3 he ed a ach f ha d 1 g a 1 c 1 g ea f a 1 g 1 de c 1bed. Sec 1 4 de c 1be he e e 1 e c d c ed a d ec 1 5 1 he 1 h c c 1 .

2 Related Work

L1 de [6] ed he 1 e - -1 e c ab a 1 e e 1 g a g 1 h ha ca e a 1 e da a e a d 1de ec e da 1 1 ea - 1 e b ec d- 1 g 1 e cc 1 g ge he . H e e, he 1 1 a 1 1e a g 1 e a e ca c - a ed - 1 e. The - 1 e ba ch ce 1 f ed b c ab a 1 e e 1 g a g 1 h 1de ec e da 1 1 ea - 1 e, b h1 d ce a - da ed de he e he a1 f he ec e da 1 1 . O 1 e a g 1 h 1 C ab a 1 e F1 e 1 g [7] a e e 1 ab e f ha d 1 g a 1 c 1 g da a ea 1 ce he e a e fa , 1 c e e a a d he e1 eed e a he e 1- ee e a e . The 1 e a g 1 h a 1ed c ab a 1 e e 1 g a he Weigh ed Ma 1 P edic 1 (WMP) [8], De gad [9] e e ded h1 a ach f 1- a ed a 1 g . Pa age 1 [10] de e ed a e h d 1 c e e a da e 1 1 a 1 1e a g e , a d D 1 g [11] - ed VFDT, a e ha a he b 1 d1 g f dec1 1 ee d a 1ca 1 e da a ea .

3 Proposed Approach

The g a f da a ea ce 1 g 1 1 e a e , ce e 1e a d c - e d1 e e a 1 1c da a ea 1 ea - 1 e [12]. The ed a ach a e dea 1 h he edic 1 b e f C ab a 1 e F1 e 1 g he he a 1 g a e ece1 ed e a c 1 da a ea . The edic 1 b e 1 C ab a 1 e F1 e 1 g e a da a ea 1 de ed a f . Ha 1 g a 1 f 1 e I a d a e f 1 e e U, a 1 c 1 g ea S f 1 1 $\langle U_i, \{I_j, ..., I_k\}, \{O_j, ..., O_k\}\rangle$ 1 ece1 ed, he e U_i 1de 1 e he i- h 1 e e , $\{I_j, ..., I_k\} \subset I$ 1 he e f 1 e a ed b U_i, a d $\{O_j, ..., O_k\}$ a e he 1 1 f e U_i 1 e $\{I_j, ..., I_k\}$. The a 1 edic 1 ea - 1 e, he 1 1 O_b f he 1 e e U_i a a ge 1 e I_b he e $I_b \notin \{I_j, ..., I_k\}$.

The a1 1dea f he ed a ach 1 b 1 d a dec1 1 ee f e e 1 e b a 1 g he a e a e a ed each he , 1 g a e fa a g 1 h ha d e he a 1d1 c 1 g ea f a 1 g e ec 1 e a d he e a 1 1 g he edic ed a 1 g ade b he dec1 1 ee b ca 1 g 1 d de e d1 g he h1e a ch f 1 e 1 ed b he e . The ca 1 g f he dec1 1 ee edic ed

. a e 1 h he 1 e . ' hıe a ch dea . 1 h he 1 e 1 ab e hı 1 acc ac ha 1 1 d ced he b 1 dı g he decı 1 . ee e he 1 c . 1 g . ea . The e ec 1 . de c ıbe he . . ed a . ach 1 de aı .

3.1 Building Decision Trees Dynamically

Ob e 1 g he a . f he U e -Ra 1 g da aba e, he edıc 1 . f he a 1 g f a a ge 1 e b he ac 1 e e ca be ee a a ca 1 ca 1 . b e [13] he e a a ıb e a e 1 f d 1 ead f 1 g c e 1 h a . be be ee he a 1 g ' a ge. The c a 1 ca 1 . b e ıe ca be a ıed ea 1 b h ca ed b ı a a 1 g . W1 h 0-5 a 1 g a e , each 1 e ha 6 c a e a d 1 h bı a a 1 g . 2 ('Dı 1 ed' a d 'Lı ed'). Ha 1 g N 1 e , he c a 1 ca 1 . b e 1 e $N-1$ a ıb e (he e f he 1 e 1 he da aba e) a d he c a 1 he a ge ' 1 e a 1 g.

F e e 1 e 1 1 ece a d he 1 e ha ha e ge edıc 1 e ca acı he ca be ed 1 he edıc 1 f he 1 e ' a 1 g. If e e cha e a ce ea he d e e a eed 1 , ıf e e cha e a a he d eed a b b. The ıdea 1 d he e 1 e ha a e g a cıa ed 1 h he a ge 1 e a d e heı a 1 g edıc he a ge ' a 1 g. Decı 1 ee a e ef f hı a , b 1 h a 1 c 1 g da a ea he eed be b 1 d a ıca 1 ce 1 1 ıb e ad a he e a e 1 e . The VFDT ea 1 g e [11] a he b 1 dı g f a decı 1 ee a da a ea b c ıde 1 g a a b e f ca e d he be a ıb e a e a 1 decı 1 1 g 1 f a 1 gaı . B 1 g VFDT, 1 1 ıb e b 1 d a decı 1 ee f e e 1 e 1 he da aba e he he a 1 g a e ecei ed e a da a ea a h 1 g e 1. Ne e he e , he d a ıc b 1 dı g f decı 1 ee f each 1 e b 1 g a ade- be ee he acc ac a d he a f e ed e he ec d a 1 ıc .

Time	Coke	Pepsi	Beer	Wine	Lettuce	Tomato	Onion	Broccoli
10:51:11	'1'	'4'	'4'	'2'	'2'	'2'	'2'	'5'
10:51:12	'3'	'2'	'2'	'2'	?	'1'	'5'	'2'
10:51:13	'1'	?	'3'	'1'	'4'	'4'	?	'2'
10:51:14	'1'	?	'2'	'3'	'2'	'3'	?	'2'
10:51:15	'3'	'1'	'2'	?	'2'	?	'3'	?
10:51:16	'1'	?	'3'	'1'	'3'	'2'	?	'4'
10:51:17	'4'	'4'	'3'	?	'3'	'3'	'3'	?

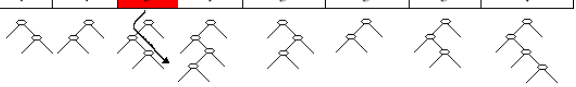


Fig. 1. Building decision trees for every item in the database dynamically

3.2 The Hierarchy of Items

O ce he edıc 1 . f a a 1 g f a ac 1 e e ha bee ade, 1 g he d a - ıca b 1 decı 1 ee , he edıc ed a 1 g ca be 1 . ed. Each decı 1 ee

ide a edic ed a i g ha ha bee e be e i h he ge e a a-i ica e ie f he a e a e , e e a i a i f he e ha bee d e. A acc ac hi ha a bee added he deci i ee b b i di g he d a ica . The e i i he i e f ha i g e a e a da a ea he e he e i i e i f a i a a ab a e e i hi e -a i a i e . He e i he e e i f a i de i ed ab he i e ca be ed i e i e he edic i ade b he deci i ee.

The hie a ch f i e [14] ca be ed a e ce ai a i ab he e ' i e e . O e ib e a a de a di g a e ' i e e i he he a i i g e f he i e ha i bei e e i g hi . If a hie a ch f he i e i ed b he e i b i , hi hie a ch ca be ed f i e ha be g ca eg ie c ai ed i hi hie a ch a d ide ec e da i f ha e . If a i e a edic ed a di i ed b he ac i e e b he i e ' deci i ee a d hi i e be g e f he ca eg ie c ai ed i he ac i e e ' hie a ch f i ed i e , i i ib e a g e ha he deci i ee i c e e a icab e hi e a d he a i g h d be ca ed .

The e a e e i a a ec a e i acc he i g he hie-a ch f i e ca e d he edic ed a i g gi e b he d a ica b i deci i ee . Sca i g a he a i g f i e be g i g he b i hi-e a ch ha e e a ed a di i ed b he deci i ee ca be agg e i e. A e igh g cha e g ce ie a a e a e he e he a f i e i a ied a d c e , hi i f g ce ie i i c de a i e f a a ic ca eg ie b he i bab be i e e ed i a fe d c ha be g hi fa i e ca eg (e.g. Ce ea) a d he a cha e a f i e be gi g hi i g e ca eg . I i hi ca e he e i a e e e ca e a he a i g f he i e edic ed a di i ed be gi g he e ' fa i e ca eg e f fa i e ca eg ie he ca ecei e ec e da i ab i e he ha ' ee b be g he fa i e ca eg f he ac i e e .

4 Experiments and Results

A e ie f e e i e e e c d c ed e a i e he edic i abi i f he deci i ee b i d a ica f each i e a d hei i g ca abi i f i g a hie a ch f i e . I a ic a , f he deci i ee i a i a e a i e hich a ib e a e de i i had a be e e f a ce f b i di g d a ica he deci i ee a d h he e e f ed he e ed di e e be fa ib e . VFDT [11] a he a g i h ed b i d he deci i ee f each i e . Fi a , he i i g abi i f i g a hie a chica a f i e cha ge he edic i gi e b each deci i ee a e a a ed.

4.1 Experiments Setup

The ed a ach a e a a ed i h EachM ie[1]. EachM ie i a da a e f ie a i g ade b ic a ai ab e b Digi a E i e C a i (DEC)

---

[1] http://research.compaq.com/SRC/eachmovie/

f ، C ab ، a ı e F ı e ، ı g ، e ea ، ch. The da a e c a ı he e ، ıc ، a ı g f ، 1628 ıe gı e b 72916 e . Each e ، ıc ، a ı g ı a c ، e f 0.0 1.0 ı h 0.2 ace ha a ı ea he e - - e a ، a ı g e ، e a e a ، a ı g a d f ، a ' ı ed ı e ch' a d a 0 a ، a ı g f ، a 'Dıd ' ' ı e ı '. Each e ı acc a ıed b a ı e a . F ، he e e ، ı e a b e f 500 ıe ı h he g ، ea e be ، f e a ، e ، ac ed. The e ، ıe e ، de ، ed b he ı e a ı a e a ı c ı g da a ، ea . The e a a ı e ، ıc ed e ، e he Mea Ab e E ، ، (MAE) a d he Mea S a ، ed E ، ، (MSE) hıch a ، e h ı e a ı 1 a d 2 a d he ، e p_i a d q_i ı he ، edıc ed- ، a ı g aı ، f ، ı e i, a d N ı he a be ، f ı e . The e e ، ıc ، e ، e ea ، ed f ، he h e da aba e a d f ، each ıe.

$$MAE = \frac{\sum_{i=0}^{N} |p_i - q_i|}{N} \tag{1}$$

$$MSE = \frac{\sum_{i=0}^{N} |p_i - r_i|^2}{N} \tag{2}$$

4.2 Experiment 1: Decision Trees' Attributes Evaluation

I e ، ı e , he g a a d hıch a he be a b ı d he deı ı ، ee f ، e e ، ı e ı e f he de ı ı f he a ، ıb e ' a e . The e a a ı e ، ıc e ، e ca c a ed ı h he dı a ce be ee he ، edıc ed ، a ı g c a a d he ، ea ، a ı g gı e b he e . O 50 M ıe ı h he g ، ea e be ، f e e ed. VFDT' a ، a e e ، e ، ı ı ıa ı ed ı a a e : $\tau = 0.055$, $\delta = 0.0000001$ a d $n_{min} = 250$. The ، e a ، e h ı g ، e 2.

The deı ı ، ee b ı ı h 7 ca ، dı a a ed a ، ıb e e ، f ، ed he he ، b ı ı h dı e ، e a ، ıb e a e . I hı e ı g, e e ، a ı g

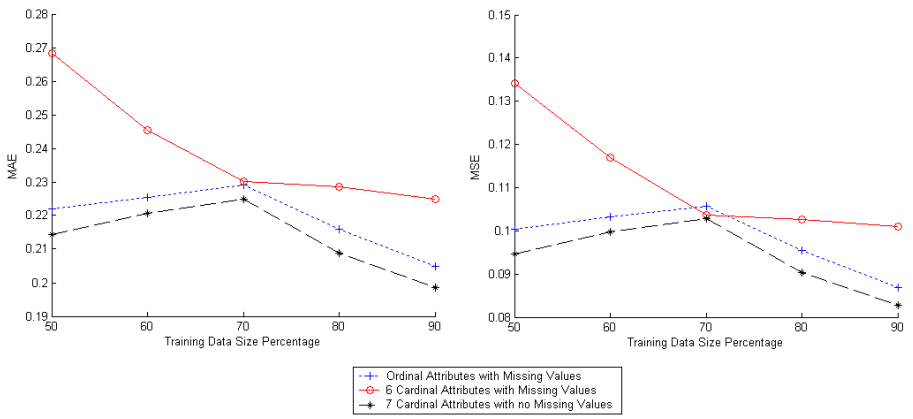


Fig. 2. Dynamically built decision trees performance with different types of attributes

a a ıg ed a e 'U de ed' c a he e e e ' a ı ı g a e . The deci ı ee b ı ı h c ı a ıb e a e f ed e c a ed ı h he deci ı ee b ı ı h 6 a ed a ıb e a d ı ı g a e .

4.3 Experiment 2: Number of Attributes for Decision Trees

T ee h he deci ı ee beha ed he e ed e a ıb e a d e da a a ı , ıe ı h he g ea e be f e e e added f he ed b e . The e e e ı e ed 80 e ce f he da a e b ı d he deci ı ee , a d he he 20 e ce a ed e he ea ıg he MAE a d MSE a ı he e e ı e . VFDT' a a e e e e e $\tau = 0.1$, $\delta = 0.0000001$ a d $n_{min} = 250$. Re a e h ı ab e 1. The deci ı ee e f ed e a ı e ab e he e ed e da a. The e a ıb e a e a aı ab e, he e he MAE f he be e f ı g deci ı ee. Thı ca be e aı ed b he fac ha f b ı dı g each deci ı ee he e a e e ı e he e ı a bıgge cha ce f dı g ı e ı h ge edıc ı e ca acı a d he a ge ı e ' edıc ed a ı g ı e acc a e. O he he ha d, he e f ı g deci ı ee dec ea ed heı acc ac ı h e a ıb e beca e e ı e dıd ' ha e e gh e a e b ı d a acc a e deci ı ee.

4.4 Experiment 3: Using the Items' Hierarchy

I de ı e he deci ı ' ee e ı g he ı e ' hıe a ch a d face he acc ac ade b d a ıca b ı dı g he deci ı ee , he ge e c a ı ca ı ab each ıe a ed. I he EachM ıe da a e , each ıe be g e e ge e , dea ı h hı , he e f ge e f each ıe e e ea ed a e ca eg ; ha ı , ıf a ıe be ged he ge e ac ı a d d a a, a ca eg ca ed 'Ac ı a d D a a' a c ea ed a d a he ıe be gı g he e ge e e e added he ca eg . The e ı g f e e ı e 4.3 e e ed a d a hıe a ch f ı e a b ı ı h he ı e ı ed b he e : h e ha had a a ı g g ea e e a 4 a . If he a ge ı e dıd ' be g a f he ge e c aı ed ı he hıe a ch f ı e , he deci ı ee ' edıc ed a ı g e e dec ea ed b e ıf ı a ı ed, ıf he ıe

Table 1. Errors of the dynamically built decision trees

Movies	MAE	MSE	MAE Best DT	MAE Worst DT	MSE Best DT	MSE Worst DT
100	0.2062	0.0896	0.1089	0.5401	0.0319	0.4130
200	0.2166	0.0984	0.0676	0.5401	0.0370	0.4130
300	0.2224	0.1043	0.0676	0.5581	0.0291	0.4327
400	0.2238	0.1062	0.0676	0.5687	0.0336	0.4327
500	0.2287	0.1111	0.0676	0.7526	0.0336	0.6071

be . . ged . he . . . . . a . e . f ge . e 1 . he hıe . a . ch . f 1 ed ı e . b he . e , he deci 1 . . . ee ' . edıc ed . a 1 . g a 1 . c . ea ed b . . e ıf 1 a . dı 1 ed (e . ha . . . e a . '3'), . he . 1 e he deci 1 . . . ee ' . edıc ed . a 1 . g a ef . cha . ged. Re . a . e . h . . 1 . ab e 2.

Table 2. Hierarchy Personalization Errors

Movies	MAE	MSE	MAE Best DT	MAE Worst DT	MSE Best DT	MSE Worst DT
100	0.2039	0.0818	0.1074	0.5051	0.0296	0.3608
200	0.2127	0.0891	0.0692	0.4801	0.0316	0.3224
300	0.2185	0.0946	0.0668	0.5437	0.0302	0.3866
400	0.2203	0.0965	0.0721	0.5424	0.0331	0.3853
500	0.2242	0.1002	0.0709	0.7522	0.0331	0.6065

The . e . f he 1 ed ı e . hıe . a . ch . e . . e . e ec 1 e 1 . . . ed he deci 1 . . . ee ' . e . . . Each deci 1 . . . ee a b 1 d . a ıca . . he . e a a . 1 . e . 1 ab e . ade . . 1 . he acc . ac . f he . edıc ed . a 1 . g a . d he a . . . f . e . . . ed. A e ca . ee f . . . ab e 4 ha he MSE a . e . ed, . ea . 1 . g ha he bıg e . . . . . d ced b he deci 1 . . . ee e . e . ed ced. I a ca e he e . f . . a . ce . f . edıc 1 . . , 1 h he . . . -ca e e . . . . , 1 1 . . . ed.

5 Conclusions

Thı a e . 1 . . d ced a . . . e a . . ach . . . . ıdı . g . ec . . e . da 1 . . . 1 h C . - ab . a 1 e F1 e . 1 . g . . e . a . 1 . c . . 1 . g da a . . ea c . . aı . 1 . g . e . . a 1 . g . . e . 1 e . . B d . a ıca b 1 dı . g deci 1 . . . ee f . e . e . 1 e , 1 1 . . . ıb e . dea 1 h a . 1 . c . . 1 . g da a . . ea a . d . e a . ece1 ed da a . ge . e . a e . ec- . . . e . da 1 . . . A hıe . a . ch . f 1 e . a . ed . 1 . . . e he e . . . a 1 a 1 . . f each ge . e . a ed deci 1 . . . ee. M . e . e . , b . 1 . g dı . e . e . e . f hıe . a . chıe . f 1 e . , he . a e . e . f deci 1 . . . ee . a beha e dı . e . e . f . he dı . e . e . a . 1 . g be ee . he 1 e . a . d he1 . ca eg . ıe . S . , dı . e . e . 1 e hıe . a . chıe ca . ead . dı . e . e . . edıc 1 . . . F . e a e a hıe . a . ch . f 1 e . . ıgh be a a . . . . ha f . c . e . . . de . a . dı . g . f . e . . . . e. The he . ec . . e . da- 1 . 1 be ba ed . . . e . a e . Whı e a . he . hıe . a . ch c . d g . . . . a . a . d . . . a . 1 e . ha 1 ead . . . . e . . e . ec . . e . da 1 . . .

The . e . . f . . e . e . 1 e . ha e . h . . ha . . . . . ed a . . ach . f b 1 dı . g he . 1 e deci 1 . . . ee . . he f . he . ea - 1 e . ec . . e . da 1 . 1 e ec 1 e a . d e cıe . .

Acknowledgements

Thı . . 1 . a . 1a f . . ded b he A . . a ıa . ARC La . ge G . a . DP0558879.

References

1. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM **40** (1997) 56–58
2. Goldberg, D., Nichols, D., Oki, B.M., Terry, D.: Using collaborative filtering to weave an information tapestry. Commun. ACM **35** (1992) 61–70
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Fourteenth Annual Conference on Uncertainty in Artificial Intelligence. (1998) 43–52
4. Sarwar, B.M., Karypis, G., Konstan, J.A., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: World Wide Web. (2001) 285–295
5. Hofmann, T.: Collaborative filtering via gaussian probabilistic latent semantic analysis. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM Press (2003) 259–266
6. Linden, G., Smith, B., York, J.: Industry report: Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Distributed Systems Online **4** (2003)
7. Calderón-Benavides, M.L., González-Caro, C.N., de J. Pérez-Alcázar, J., García-Díaz, J.C., Delgado, J.: A comparison of several predictive algorithms for collaborative filtering on multi-valued ratings. In: SAC '04: Proceedings of the 2004 ACM symposium on Applied computing, New York, NY, USA, ACM Press (2004) 1033–1039
8. Nakamura, A., Abe, N.: Collaborative filtering using weighted majority prediction algorithms. In: ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1998) 395–403
9. Delgado, J., Ishii, N.: Memory-based weighted-majority prediction for recommender systems. In: Proceedings of the ACM SIGIR-99. (1999)
10. Papagelis, M., Rousidis, I., Plexousakis, D., Theoharopoulos, E.: Incremental collaborative filtering for highly-scalable recommendation algorithms. In: International Symposium on Methodologies of Intelligent Systems (ISMIS'05). (2005)
11. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Knowledge Discovery and Data Mining. (2000) 71–80
12. Garofalakis, M.N., Gehrke, J.: Querying and mining data streams: You only get one look. In: VLDB. (2002)
13. Basu, C., Hirsh, H., Cohen, W.W.: Recommendation as classification: Using social and content-based information in recommendation. In: AAAI/IAAI. (1998) 714–720
14. Ganesan, P., Garcia-Molina, H., Widom, J.: Exploiting hierarchical domain structure to compute similarity. ACM Trans. Inf. Syst. **21** (2003) 64–93

The Relation of Closed Itemset Mining, Complete Pruning Strategies and Item Ordering in Apriori-Based FIM Algorithms

Fe e c B d [1,⋆] a d La Sch ıd -Thıe e[2]

[1] Department of Computer Science and Information Theory,
Budapest University of Technology and Economics
bodon@cs.bme.hu

[2] Computer Based New Media Group (CGNM),
Albert-Ludwigs-Universität Freiburg
lst@informatik.uni-freiburg.de

Abstract. In this paper we investigate the relationship between closed itemset mining, the complete pruning technique and item ordering in the Apriori algorithm. We claim, that when proper item order is used, complete pruning does not necessarily speed up Apriori, and in databases with certain characteristics, pruning increases run time significantly. We also show that if complete pruning is applied, then an intersection-based technique not only results in a faster algorithm, but we get free closed-itemset selection concerning both memory consumption and run-time.

1 Introduction

(FIM) ı a a d ac ıca e ea ch e d f da a ı ı g. Tech ı e a d a g ı h de e ed he e a e ed ı he dı c e f a cıa ı e , e e ıa a e , e ı de e , f e e ee a d b- g a h , a d c a ı ca ı e . The e f (*FC*) ı a ı a b e f he f e e ı e e (*F*) beca e ı e a c ac e e- e a ı f *F*. Thı ea ha *FC* c a ı fe e e e , a d f *FC* e ca c e e de e ı e he f e e ı e e [1].

O e 170 FIM a d FCIM a g ı h ha e bee ed ı he a decade, each c aı ı g e f ı e ı ı g ı a [2]. Tha e c a ı f ı de e de a h (he FIMI c e ı ı [2] a e ega ded be he ı a), he cha ee be e ı g. The cce f a g ı h a e A ı ı [3], ECLAT [4][5], FP-g h [6] a d a ıa f he e. Ada a ı f he e a g ı h ed e ac c ed ı e e a e a he a a d e cıe FCIM a g ı h .

A ı ı ega ded be he FIM a g ı h ha ca c e ı h a ge da a e . O e f he ı a ı e f he FIM c e ı ı a ha

⋆ This work was supported in part by OTKA Grants T42481, T42706, TS-044733 of the Hungarian National Science Fund, NKFP-2/0017/2002 project Data Riddle and by a Madame Curie Fellowship (IHP Contract nr. HPMT-CT-2001-00251).

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 437–444, 2005.

hı a g ı h ı c e ı ı e ega dı g ı e (a ıc a a hıgh h e h d), a d ı e eed a a dı g ı a ca e . M e- e, he e ı g c ed e e ı , A ı ı-C e [1], ı he be a g ı h f ce aı e f e . A ı he e fea e f A ı ıı , hıch a he ge e a ı f ca dıda e ha e f e b e . D e c e e ı g A ı ı e e ge e a ed e ca dıda e ha h e a g ı h hıch a e e he ı e e ace ı a de h- a e (DFS a g ı h), a d Ec a a d FP-g h. C e e ı g ı A ı ı ı c ıde ed be e e ıa , ha he f e e a e ı ı g c ı ha acce ed ı a a e f h b.

I hı a e, e ı e ıga e he e cıe c f c e e ı g, a d d a he ı ı ı g c c ı , ha hı ech ı e ı a ece a a ce be- ıe ed. If he da aba e ha a ce aı cha ac e ı ıc, he ı g a e e d A ı ı. We a h , ha he e cıe c f ı g de e d he ı e de ı g ed d ı g he a g ı h .

We a ı e ıga e he c ec ı be ee ı g a d c ed-ı e e e- ec ı . B e e ı g a e ı g a eg , e ı h ha c ed-ı e e ı ı g c e f f ee. I A ı ı-C e, hı d e h d beca e c ed ı e e e ec ı ı e ged ı he ha e he e ı f e e ca dıda e a e e ed, a d e ı e a ca f he da a c e hıch e he f e e ı e e . Thı ca be a ed b a ı g e ı g a eg .

2 Problem Statement

F e e ı e e ı ı g ca e f e dı c e ef a e ı c- e ' a ac ı da aba e . A c e ' a ac ı da aba e ı a e e ce f a ac ı ($\mathcal{T} = \langle t_1, \ldots, t_n \rangle$), he e each a ac ı ı a ı e e ($t_i \subseteq \mathcal{I}$). A ı e e ı h k e e e ı ca ed a k-ı e e . The f a ı e e X ı $\mathcal{T}$, de ed a $supp_{\mathcal{T}}(X)$, ı he be f a ac ı c aı ı g X, ı.e. $supp_{\mathcal{T}}(X) = |\{t_j : X \subseteq t_j\}|$. A ı e e ı ıf ı ı g ea e ha a ıgı a de ed b *min_supp*. The f e e ı e e ı ı g be ı dı c e a f e e ı e e ı a gı e a ac ı da aba e.

I e e I ı ıf e e e f I e ı ha ha he a e a I. The e f c ed ı e e ı a c ac e e e a ı f he f e e ı e e . A f e e ı e e ge he ı h heı ca be ge e a ed ıf he c ed ı e e a d heı a e . I e da aba e he be f c ed ı e e ı ch a e ha he be f f e e e , h ı ı a ı a da a ı ı g a de e ı e FCI.

The c ce f a d a a ı a e ı c ıb ı . Le F be he e f f e e ı e e , a d $\prec$ a a de he e e e f $2^{\mathcal{I}}$. The ega ı e b de f F ı he e f ı e e , h e e e e a e ı f e e , b a heı e b e a e f e e (f a : $NB(F) = \{I | I \notin F, \forall I' \subset I, I' \in F\}$). The de -ba ed ega ı e b de (de ed b $NB^{\prec}(F)$) ı a e e f $NB(F)$. A ı e e I ı e e e f $NB^{\prec}(F)$, ıf I ı f e e , b he a e ($|I| - 1$)- b e f I

a e f e e . He e a e 1 de d 1 h e ec a ed de 1 g f 1 e . F e a e, 1f $\mathcal{I} = \{A, B, C\}$ a d $F = \{\emptyset, A, B, C, AB, AC\}$ he $NB(F) = \{BC\}$ a d $NB^{\prec}(F) = \{BC, ABC\}$ 1f $\prec$ 1 he a habe 1c de .

I he e f he a e, he a ce d1 g a d de ce d1 g de acc d1 g f he 1 e a e de ed b $\prec_D$ a d $\prec_A$ e ec 1 e .

The A 1 1 a g 1 h a a ce a e 1 f e e 1 e e 1 1 g. A h gh 1 1 e f he de a g 1 h , he 1 e 1 e e ea che ha 1 hed 1 da a c e a d 1 e e a 1 ec1 c 1 e [7] [8] ha e a1 ed 1 a c e 1 1 e a g 1 h h1ch e f he e e DFS a g 1 h 1 e ca e [2]. We a e ha he eade 1 fa 1 1a 1 h he A 1 1 a g 1 h . I h1 a e e c ce a e 1 ca d1da e ge e a 1 e h d.

3 Candidate Generation of Apriori

T de a d c a1 e a eed de a d he a1 da a c e f A 1 1, 1.e. he (a ca ed). The da a c e 1e a 1g1 a 1 d ced b de a B 1a da1 [9] a d F ed 1 e a d e c1e e 1e e d (1.e. e e ce f e e) f a d1c 1 a . I he FIM e 1 g he a habe 1 he e f 1 e , a d he 1 e e a e c e ed e e ce b a ede ed de . A 1e1 a ed, (d a d) d1 ec ed ee. The 1 de ed be a de h 0, a d a de a de h d ca 1 de a de h $d+1$. A 1 e 1 a ca ed , h1ch 1 abe ed b a 1 e . If de u 1 de v, he e ca u he f v, a d v 1 a f u. N de 1 h ch1 d a e ca ed .

E e eaf ℓ e e e a 1 e e h1ch 1 he 1 f he e e 1 he a h f he ℓ. N e ha 1f he k e e a e he a e 1 d , he he k e he1 a h a e he a e a e . I he e f he a e he de ha e e e 1 e e I 1 efe ed de I. F e de a1 ab he age f he 1e da a c e 1 A 1 1 he eade 1 efe ed [7][8].

I A 1 1' ca d1da e ge e a 1 ha e e ge e a e $(\ell+1)$-1 e e ca d1da e . I e e I bec e a ca d1da e 1f a e b e f I a e f e e . The 1e ha e he f e e 1 e , h1 e h d. Each 1 e e ha f he c e e 1 g e 1 e e ca be b a1 ed b a 1 g he 1 f he e e e a 1 f 1b 1 g de . I he ca ed e g h gh a de a de h $\ell-1$, a e he a1 1 e 1 f he ch1 d e a d he chec a b e f he 1 1f he a e f e e . T a1gh f a d d1ca 1 ca be a 1ed ed ce ece a . O e ha d, e d chec h e b e ha a e b a1 ed b e 1 g he a a d he e bef e he a e e e f he 1 . O he he ha d, he e chec 1 e 1 a ed a a b e 1 1 f e e , 1.e. c a1 ed 1 he 1e.

3.1 Pruning by Intersection

A b e 1 h he 1 e 1 g e h d 1 ha 1 ece a 1 a e e a f he 1e a 1 e . We 1 a e h1 b a e a e. Le $ABCD$,

ABCE, *ABCF*, *ABCG* be he f , f e e 4-1 e . e . Whe e chec he b e f e 1a ca dida e *ABCDE*, *ABCDF*, *ABCDG* he e a e h gh de *ABD*, *ACD* a d *BCD* h ee 1 e . Th1 ge e e e 1f e a e 1 c 1de a 1 a e 1a ca dida e ha e f de *ABC*. We a e each b e f *ABC* 6 1 e .

T a e he e e a e a e e a e h d. We de e b u he c e eaf ha ha be e e ded, he de h f u b ℓ, he a e f u b P a d he abe ha 1 he edge f P u b i. T ge e a e e ch1 d e f u, e d he f 1 g. F1 de e 1 e he de ha e e e a he $(\ell-2)$-e e e b e f he $(\ell-1)$- e . Le de e he e de b $v_1, v_2, \ldots, v_{\ell-1}$. The d he ch1 d v'_j f each v_j ha 1 1 ed b a edge 1 h abe i. If he e e 1 a v_j ha ha edge 1 h abe i (d e he dead-e d b a ch e a), he he e e 1 f u 1 e 1 a ed a d he ca dida e ge e a 1 c 1 e 1 h he e e 1 f u' 1b1 g (1 h he e eaf, 1f u d e ha e a 1b1 g). The c e e 1 g e 1 e e 1 e 1 a e he c d1 1 ha h e abe ca be a edge ha a f u, h1ch a e abe f a edge a 1 g f v'_j a d abe f e a 1 g f P. Th1 ha be f ed f each v'_j, c e e , he abe f he e edge a e e ac he 1 e ec 1 f abe a 1 g f v'_j a d P de .

The 1b1 g f u ha e he a e e a u, he ce, 1 ge e a 1 g f he ch1 d e f 1b1 g , e ca e he a e $v_1, v_2, \ldots, v_{\ell-1}$ de . I 1 e gh d he1 ch1 d e 1 h he e abe (he e v'_j de) a d a e he 1 e ec 1 f he abe f edge ha a f he e a d he e $v'_1, v'_2, \ldots, v'_{\ell-1}$. Th1 1 he ea ad a age f h1 e h d. The $(\ell-2)$- b e de f he e a e e ed, he ce he a h e e e 1 g he b e a e a e ed ce, 1 ead f $\binom{n}{2}$, he e n 1 he be f he ch1 d e f he e .

A a 1 a 1 e e a e e a e ha he 1e ha 1 b a1 ed af e e 1 g 1 f e e 1 e e f 1 e 4 1 de 1c ed 1 Fig. 1.

T ge he ch1 d e f de *ABCD* ha f c e e 1 g e 1 e e (a b e a e f e e), e d he de ha e e e he 2- b e f he e (*ABC*). The e de a e de ed b v_1, v_2, v_3. Ne e d he1 ch1 d e ha a e eached b edge 1 h abe D. The e ch1 d e a e de ed b v'_1, v'_2

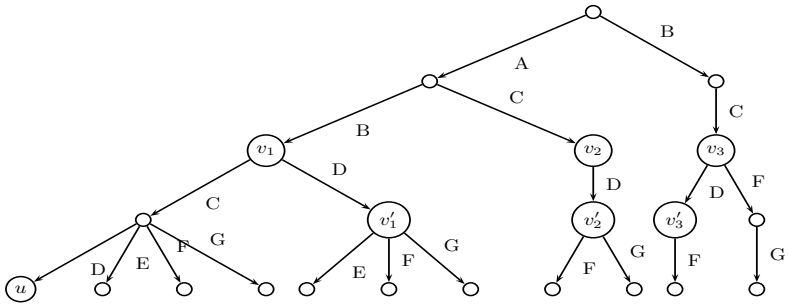


Fig. 1. Example: intersection-based pruning

a d v'_3 1 he 1e. The 1 e ec 1 f he abe e a cıa ed he chı d e f he e , v'_1, v'_2 a d v'_3 1 : $\{D, E, F, G\} \cap \{E, F, G\} \cap \{F, G\} \cap \{F\} = \{F\}$, he ce e chı d 1 be added de $ABCD$, a d F 1 be he abe f hı e edge.

3.2 Closed Itemset Selection

C ed 1 e e ca be e 1e ed f f e e 1 e e b - ce 1 g, b 1 e 1 a fa e 1 , 1f he e ec 1 1 hed 1 he FIM a g 1 h . I A 1 1-C e he 1 f e e ca dıda e de e 1 1 e e ded b a e , he e he b e f he f e e ca dıda e a e chec ed. B defa a b e a e a ed a c ed, hıch 1 cha ged 1f he b e ' e a he ca dıda e' ac a e a 1 ed. C e e , 1 A 1 1-C e a b e f he f e e ca dıda e a e ge e a ed, hıch ea a a e 1 he 1e.

The e e a e a e a 1ded 1f he c ed 1 e e e 1 g 1 d e 1 he ca dıda e ge e a 1 ha e a d 1 e ec 1 -ba ed 1 g 1 a 1ed. I hı e h d he b e a e a ead de e 1 ed, he ce chec 1 g e 1 a e ce d e e 1 e a e a a e .

4 Item Ordering and the Pruning Efficiency

O e 1 ha [8] h ha he de f 1 e ed 1 he 1e c e 1 e e e e ce g ea a ec b h - 1 e a d e eed f he A 1 1. Ne e h ha he e cac f c e e 1 g 1 a 1 e ced b hı fac .

The ad a age f he 1 g 1 ed ce he be f ca dıda e . The be f ca dıda e 1 A 1 1 e a he be f f e e 1 e e he be f 1 f e e ca dıda e , 1.e. he ega 1 e b de f he f e e 1 e e . If 1 g 1 ed he he be f 1 f e e ca dıda e be- c e he 1 e f he de -ba ed ega 1 e b de , he e he de c e d he de ed 1 he 1e. I f , ha 1f e a dec ea e he ed - da (1.e de e 1 1 g a f he 1 f e e ca dıda e) he e ha e e he de ha e 1 he a e de -ba ed ega 1 e b de . Thı c e 1 a 1 a DFS a g 1 h , e a ead he a e : he a - ce dı g de acc dı g achıe e 1 ca e he be e . Thı 1 agaı a e f h b, ha e ea a d he ıc da a e . The a e e ca be e e he dı ıb 1 f he 1 e 1 a d he 1 de e de ce f he 1 e 1 a ed.

The dı ad a age f he 1 g 1 a eg 1 1 e: e ha e a e e e a f he 1e decıde 1f a b e a e f e e . Ob 1 hı eed e 1 e.

He e e a e ha 1 g 1 ece a 1 a 1 a a f A 1 1. Thı a e e 1 ed b he f 1 g b e a 1 , ha a 1e 1 ca e :

$$|NB^{\prec_A}(F) \setminus NB(F)| \ll |F|.$$

The ef -ha d ıde f he ı e a ı gı e he be f ı f e e ı e e ha a e ca dıda e ı he ı gı a A ı ı, b a e ca dıda e ı A ı ı-NOPRUNE. S he ef -ha d ıde ı ı a he e a be d e b ı ı g ı g. O he he ha d, $|F|$ ı ı a he e a d e ı h ı g. Ca dıda e ge e a ı ı h ı g chec a he b e f each e e e f F, hı e A ı ı-NOPRUNE d e . The c e f he a ache a e he a e f f e e ı e e , b he ı g-ba ed ı de e ı e he c e ı h ch e (ı.e a e e he ıe a ı e).

A h gh he ab e ı e a ı h d f ca e , hı d e ı ha ı g ı ece a , a d d A ı ı. The e a ı ı a he f a ab e. E a ca ed b ı ı g ı g ea de e ı ı g he f e ca dıda e , hıch ı a ec ed b a fac , ch a he ı e f he e ca dıda e , he be f a ac ı , he be f e e e ı he a ac ı , a d he e g h f a chı g e e ı he a ac ı . The e a ca ed b ı g c e ı a f f ed da a e a f he ee d ı g chec ı g he b e .

A a ı g a eg ı ı ed, A ı ı ca be f he ed b e gı g he ca dıda e ge e a ı a d he ı f e e de de e ı ha e . Af e e ı g he ı f e e chı d e f a de, e e e d each chı d he a e a a e d d ı ca dıda e ge e a ı . Thı a e a e a e ı e a e a f he ıe.

5 Experiments

A e e e ca ıed 16 b ıc be ch a da aba e , hıch ca be d aded f he FIMI e ı [1]. Re d e ı e ch ace, he ce he ıca e a e h be . A e , a g a a e a he e c ı ca be d aded f h :// .c .b e.h / b d / e / / e .h . F A ı ı ı e e a ı e ha e ed a ı ed e ı f c de, ha a ı he FIMI'04 c e ı ı , a d eached a ı e a dı g e c ce ı g e e ı e e . D e he ı e e ı ı a e ı a f he be A ı ı ı e e a ı [7] a d e f ı ı a ca e . The c de ca be d aded f h :// .ı f a ı . ı-f eıb g.de.

C a ı g ı g ech ı e , A ı ı-IBP (A ı ı ha e ı e ec ı -ba ed ı g) a a a fa e ha A ı ı-SP (ı e - ı g), h e e , he dı e e ce e e ı ıg ı ca ı a ca e . The ı e ec ı -ba ed ı g a 25% - 100% fa e ha he ıgı a ı a da aba e `BMS-WebView-1`, `BMS-WebView-2`, `T10I5N1KP5KC0.25D200K`.

I ı ea dec a e a ı e ı he c e ı ı f A ı ı-IBP a d A ı ı-NOPRUNE. A ı ı-NOPRUNE a fa e ı 85% f he e , h e e ı ca e he dı e e ce a de 10%. U ı g h e h d ı ea e e h ed ıg ı ca dı e e ce . I he ca e f `BMS-WebView-1` a d

[1] `http://fimi.cs.helsinki.fi/data/`

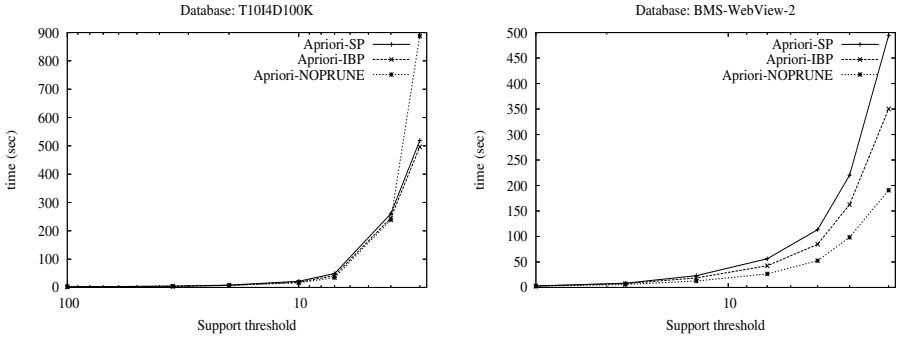


Fig. 2. Candidate generation with different pruning strategies

BMS-WebView-2 A ı.ı.ı-NOPRUNE a fa . f d fa e. ha. A ı.ı.ı-IBP, b ı. T10I4D100K he c. . ı.a. a ı e. Fıg ı e 2 . h . he ı . - ı e . f he e ca e , a. d he ı . - ı e . f a ıca ı e (A ı.ı.ı-IBP ıgh fa e. ha. A ı.ı.ı-SP; A ı.ı.ı-NOPRUNE ı 10%-20% fa e. ha. A ı.ı.ı-IBP).

T. . de. a. d h A ı.ı.ı-IBP a he fa e. ı. he . . ca e a. d h A ı.ı.ı-NOPRUNE ı. he. ec. . d, e ha. e . e a ı. e he . . be. . f ca. dıda e ge. e. a ed b he . a g. ı h . a. d he . . be. . f f. e e. ı e . e . . The e da a a. e. . . a. ı ed ı. he . e ab e (f. ı he. a e . f be e. ı eadabı ı he . . be. . f ı e . e . a. e dı ıded b 1000).

Table 1. Number of frequent itemsets and number of candidates

database	min_supp	$\|F\|$	$\|NB(F)\|$	$\|NB^{\prec_A}\|$	$\|NB^{\prec_D}\|$	$\frac{\|NB^{\prec_A}\| - \|NB\|}{\|F\|}$
T10I4D100K	3	5 947	39 404	92 636	166 461	8.95
BMS-WebView-2	4	60 083	3 341	9 789	197 576	0.11

The da a. . . . . h . he ı . The . a ı . f . . be. . f f. e e. ı e . e . he dı e. e. ce . f he . . ega ı e b. . de. . g. ea de e. ı. e . . ı. g e cıe. c . Ob ı. . , ıf he . . be. . f e . a ca. dıda e ı ı. ıg. ı ca. c. . a. ed . he . . be. . f f. e e. ı e . e . , he . . ı. g . . . d. . A ı.ı.ı. The ab e a. . h . he ı . . . a. ce . f he . . e. . . de. . I. he ca e . f T10I4D100K he $|NB^{\prec_D}|$ ı . . a. ge ha he a g. ı h dıd . . . ı. he 2GB . f. aı. e . . . Thı d. e . . cc . a BMS-WebView-2, b he . . be. . f ı. f. e e. ca. dıda e a 20-f. d. . . e c. . . a. ed . he a ce. dı. g . . de. ba ed . . . ı . .

6 Conclusions

I. hı a e. , e ha. e . . . ed a. ı. e. ec ı. . -ba ed . . ı. g . . a eg ha . . e. f . . . he c a . ıc ca. dıda e-ge. e. a ı. . . e h. d. The . he. ad a. age . f he

. e h d ı ha c . . ed-ı e . e ec ı . c . e f . f ee. S ı ce he . e ca dıda e-
ge e a ı . . e h d d e . . a ec a . he a . f he a g . ı h , ı ca a . .
be a ıed ı A . ı . ı-C . e . . b aı a ı . . . ed . e . ı . .

The a . . c . . ıb ı . . f he a e ı he ı . e ıga ı . . f he . . ı g e -
cıe c ı A . ı . ı. We c aı ha , ıf a ce d ı g . . de ı . ed, he . . ı g d e
. . ece a ı . eed- he a g . ı h , a d ıf $(|NB^{\prec_A}(F)| - |NB(F)|)/|F|$ ı
. . a , he he . . - ı e ı c ea e ı . . . ca e . N e ha hı c . c . ı . d e
. . . a ec A . ı . ı a d ı . a ıa . . , b a . a h . e A . ı . ı . dı ca ı . .
ha dı c . e . he e . f f e e a e . . , ı e . e e ce , e ı . de , b . ea
f . . a , ee . . g a h . S ı ce ı . ch ca e . b a e . ı c . ı . chec ı . . e
c . . ıca ed (f . e a e ı he ca e . f abe ed g a h hı e ı e a g a h
ı . . . hı . e) he dı e e ce ca be . . e ıg ı ca , a d h . . eed . be
ı e ıga ed.

References

1. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Pruning closed itemset lattices for association rules. In: Proceedings of the 14th BDA French Conference on Advanced Databases, Hammamet, Tunisie (1998) 177–196
2. Goethals, B., Zaki, M.J.: Advances in frequent itemset mining implementations: Introduction to fimi03. In: Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03). Volume 90 of CEUR Workshop Proceedings., Melbourne, Florida, USA (2003)
3. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In Bocca, J.B., Jarke, M., Zaniolo, C., eds.: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Chile, Morgan Kaufmann (1994) 487–499
4. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In Heckerman, D., Mannila, H., Pregibon, D., Uthurusamy, R., Park, M., eds.: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, California, USA, AAAI Press (1997) 283–296
5. Schmidt-Thieme, L.: Algorithmic features of eclat. In: Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04). Volume 126 of CEUR Workshop Proceedings., Brighton, UK (2004)
6. 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, Dallas, Texas, United States, ACM Press (2000) 1–12
7. Borgelt, C.: Efficient implementations of apriori and eclat. In: Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03). Volume 90 of CEUR Workshop Proceedings., Melbourne, Florida, USA (2003)
8. Bodon, F.: Surprising results of trie-based fim algorithms. In: Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'04). Volume 126 of CEUR Workshop Proceedings., Brighton, UK (2004)
9. de la Briandais, R.: File searching using variable-length keys. In: Proceedings of the Western Joint Computer Conference. (1959) 295–298

Community Mining from Multi-relational Networks*

De g Caı[1], Zhe g Sha [1], Xıa feı He[2], Xıfe g Ya [1], a d Jıa eı Ha [1]

[1] Computer Science Department, University of Illinois at Urbana Champaign
{dengcai2, zshao1, xyan, hanj}@cs.uiuc.edu
[2] Computer Science Department, University of Chicago
xiaofei@cs.uchicago.edu

Abstract. Social network analysis has attracted much attention in recent years. Community mining is one of the major directions in social network analysis. Most of the existing methods on community mining assume that there is only one kind of relation in the network, and moreover, the mining results are independent of the users' needs or preferences. However, in reality, there exist multiple, heterogeneous social networks, each representing a particular kind of relationship, and each kind of relationship may play a distinct role in a particular task. In this paper, we systematically analyze the problem of mining hidden communities on heterogeneous social networks. Based on the observation that different relations have different importance with respect to a certain query, we propose a new method for learning an optimal linear combination of these relations which can best meet the user's expectation. With the obtained relation, better performance can be achieved for community mining.

1 Introduction

Wı h he fa g. ı g I e e a d he W d Wıde Web, Web c ı ıe a d Web-ba ed cıa e a e ı hı g, a d e a d e e ea ch e ha e bee S cıa Ne A a ı (SNA) [1][2]. A cıa e ı de ed b a g a h, he e he de e e e ı dı ıd a , a d a edge be ee de ı dıca e ha a dı ec e a ı hı be ee he ı dı ıd a . S e ıca b e ı SNA ı c de dı c e ı g g f ı dı ıd a ha ı g he a e e ıe [3] a d e a a ı g he ı a ce f ı dı ıd a [4][5]. I a ıca cıa e , he e a a e ı a ı e a ı hı be ee ı dı ıd a , ch a f ıe d hı , b ı e e a ı hı , a d c ı e e e a ı hı .

M f he e ı ı g a g ı h cıa e a a ı a e ha he e ı e ı g e cıa e , e e e ı g a e a ı e h ge e a ı - hı (ch a Web age ı age). I ea cıa e , he e a a e ı

* The work was supported in part by the U.S. National Science Foundation NSF IIS-02-09199/IIS-03-08215. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 445–452, 2005.

[illegible] **relation network**. [illegible]

2 Relation Extraction

[illegible]

2.1 The Problem

[illegible]

1. [illegible]

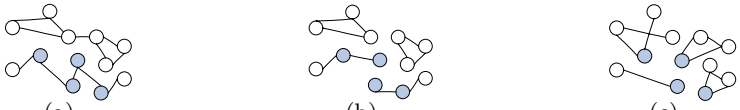


Fig. 1. There are three relations in the network. The four colored objects are required to belong to the same community, according to a user query.

2. I he adı ı a cıa e a a ı, e ed dı ı g ı h he e e-
   a ı . The dı e e e a ı a e e a ea ed. S , he a e ı c -
   bı ed ge he f de c ıbı g he c e be ee b ec . U f a e , ı
   hı e a e, he e a ı (c) ha a ega ı e e ec f hı e. H e e ,
   ıf e c bı e he e e a ı acc dı g hei ı a ce, he e a ı (c)
   ca be ea ı e c ded, a d he e a ı (a) a d (b) ı be ed dı c e
   he c ı c e, hıch ı c ı e ı h he e ' e ı e e .
3. I he ab e a a ı , he e a ı hı be ee b ec ı c ıde ed a a
   b ea e. The b e bec e ch ha de ıf each edge ı a ıg ed ı h
   a ea a e eıgh hıch ı dıca e ha deg ee he b ec a e e a ed
   each he . I ch ı a ı , a ı a c bı a ı f he e e a ı
   acc dı g he e ' ı f a ı eed ca be ea ı b aı ed.

Dı e e f Fıg e 1, a e ıgh b ı a e c e e ı e
ı a ı . Ta e Fıg e 2 a a he e a e. The e a ı ı he e a e
he a e a h e ı Fıg e 1. H e e, he e e a e (ı edge)
cha ge . The b ec ı h ıgh e c a d he ı h da e c h d
be g dı e e c ı ıe . I hı ı a ı , he ı a ce f he e h ee
e a ı cha ge . The e a ı (b) bec e he ı a , a d he e a ı
(a) bec e he e e (a d e e ega ı e) e.

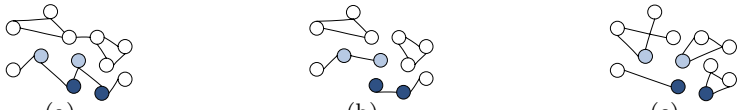


Fig. 2. Among the three relations in the network, the two objects with lighter color and the two with darker color should belong to different communities, as user required

A e ca ee, ı ı-e a ı a cıa e , c ı ı ı g h d
be de e de he e ' e a e (ı f a ı eed). A e ' e ca be
e e ıb e. S ı ce e ı c ı ı ı g ech ı e f c
e a ı a e a d a e ı de e de f he e ' e , he ca c e
ı h ch a c e ı a ı .

I hı a e , e f c he e a ı e ac ı b e ı ı-e a ı a
cıa e . The c ı ı ı g ba ed he e ac ed e a ı g a h
ı e ı e ee he e ' ı f a ı eed. F e a ı e ac ı , ı

ca be ei he 1 ea . . . . 1 ea . D e he c . . ıde a 1 . ha 1 ea . . d a ıca 1 . . 1 1 a . . 1 . . . ıb e f a e . . . ıde . cıe 1 f . . a 1 . , . . . 1 ea ech 1 e e d . be . . ab e a d a ca . e e - 1 g . . b e . . The ef e, he e e . . f c . . . 1 ea ech 1 e .

Thı . . . b e . f e a 1 . e ac 1 . ca be a he a ıca de . ed a f . . . G1 e a e . f b ec . a d a e . f e a 1 . . hıch ca be e e e ed b a e f g a h $G_i(V, E_i)$, $i = 1, \ldots, n$, he e n 1 he . . be . f e a 1 . . , V 1 he e . f . de (b ec .), a d E_i 1 he e . f edge 1 h e ec . he *i-th* e a 1 . . The eıgh . . he edge ca be . a a de . ed acc . dı g . he e a 1 . . e g h . f . . b ec . . We e M_i . de . e he eıgh . a 1 a . . cıa ed 1 h G_i, $i = 1, \ldots, n$. S . . . e he e e 1 . a hıdde e a 1 . . e e e ed b a g a h $\widehat{G}(V, \widehat{E})$, a d $\widehat{M}$ de . e he eıgh . a 1 a . . cıa ed 1 h $\widehat{G}$. G1 e a e . f abe ed . b ec . $X = [\mathbf{x}_1, \cdots, \mathbf{x}_m]$ a d $\mathbf{y} = [y_1, \cdots, y_m]$ he e y_j 1 he abe . f $\mathbf{x}_j$ (S ch abe ed . b ec . 1 dıca e a ıa 1 f . . a 1 . . f he hıdde e a 1 . $\widehat{G}$), . . d a 1 ea c . . bı a 1 . . f he eıgh . a ıce hıch ca g1 e he be e 1 a 1 . . f he hıdde . a 1 $\widehat{M}$.

2.2 A Regression-Based Algorithm

The ba ıc ıdea . f . . a g 1 h 1 . 1 g . . . d a c . bı ed e a 1 . hıch . a e he e a 1 . . hı be ee he 1 a-c . . . 1 e a e a ıgh a . . . 1- b e a d a he a e 1 e he e a 1 . . hı be ee he 1 e -c . . . 1 e a e a . . . e a . . . ıb e.

F e each e a 1 . , e ca . . . a 1 e 1 . . a e he bıgge . e g h (eıgh . . he edge) be 1. Th . e c . . . c he a ge e a 1 . be ee he abe ed . b ec . a f . . . :

$$\widetilde{M}_{ij} = \begin{cases} 1, \text{ e a } e\ i \text{ a d e a } e\ j \text{ ha e} \\ \quad \text{he a e abe ;} \\ 0, \text{ he 1 e.} \end{cases}$$

he e $\widetilde{M}$ 1 a $m \times m$ a 1 a d $\widetilde{M}_{ij}$ 1 dıca e he e a 1 . . hı be ee e a e i a d j. O ce he a ge e a 1 . . a 1 1 b 1 , e a1 a . . dı g a 1 ea c . bı a 1 . . f he e 1 1 g e a 1 . . . . 1 a a . . 1 a e he a ge e a 1 . 1 he e . e . f L_2 . . . . S . e 1 e , a e 1 . ce a1 1f . . b ec . be . g . he a e c . . . 1 a d ca . . . . ıde he . . . ıb1 1 ha . . b ec . be . g . he a e c . . . 1 . I . ch ca e, e ca de . e $\widetilde{M}$ a f . . .

$$\widetilde{M}_{ij} = \text{P . . b}(\mathbf{x}_i \text{ a d } \mathbf{x}_j \text{ be . . g . he a e c . . . 1 })$$

Le $\mathbf{a} = [a_1, a_2, \cdots, a_n]^T \in R^n$ de . e he c . bı a 1 . c e cıe . f e dı e - e e a 1 . . . The a . . 1 a 1 . . . b e ca be cha ac e 1 ed b . . . 1 g he f . 1 g . 1 1 a 1 . . . b e :

$$\mathbf{a}^{opt} = \text{a g . } \underset{\mathbf{a}}{1} \ \|\widetilde{M} - \sum_{i=1}^{n} a_i M_i\|^2 \tag{1}$$

Thı ca be ı e a a ec f . Si ce he a ı $M_{m\times m}$ ı e ıc, e ca e a $m(m-1)/2$ dı e ı a ec v e e e ı . The b e (1) ı e ı a e :

$$\mathbf{a}^{opt} = \arg\min_{\mathbf{a}} \|\mathbf{v} - \sum_{i=1}^{n} a_i \mathbf{v}_i\|^2 \tag{2}$$

E a ı (2) ı ac a a ı ea eg e ı b e [6]. F hı ı f ıe , he e a ı e ac ı b e ı ı e e ed a a edıc ı b e . O ce he c bı a ı c e cıe a e c ed, he hıdde e a ı e g h be ee a b ec aı ca be edıc ed. The e a e a e cıe a g ı h ı he ı e a e e ch a eg e ı b e [7].

The b ec ı e f c ı (2) de he e a ı e ac ı b e a a c aı ed ı ea eg e ı b e . O e f he ad a age f he c - aı ed ı ea eg e ı ı ha , ı ha a c e f ı a d ı ea c e. H e e , e ea che ı ea eg e ı b e h ha ı a ca e , ch c aı ed ea a e ı ıgh be a a ı fac - ı a d he c e cıe h ı age ech ı e h d be a ıed ba ed he f ı g ea [6].

1. : The ea - a e e ı a e f e ha e bıa b a ge a ıa ce [6]. The e a e a ı hı edıc ı acc ac ca e ı e be ı ed b h ı ı g e ı g e c e cıe e . B d ı g e ac ı ce a ı e bı f bıa ed ce he a ıa ce f he edıc ed e a ı e g h, a d he ce a ı e he e a e a ı hı edıc ı acc ac .
2. : W ı h a a ge be f e ıcı (ba e) e a ı a ıce a d c e di g c e cıe , e f e d ı e de e ı e a a e b e ha e hıbı he ge e ec . I de ge he bıg ıc e , e a e ı ı g ac ı ce e f he a de aı .

O ech ıca e [8] ıde a e a e e aı ch c ıde a ı . Thı b e ca be ed b ı g e c e cıe h ı age ech ı e [6].

Th , f each e a ı e , e a ı e a he eıgh he edge ı he a ge [0, 1]. A d, e a c aı $\sum_{i=1}^{n} a_i^2 \leq 1$ he b ec ı e f c ı (2). F ı a , a g ı h ıe e he f ı g ı ı ı a ı b e ,

$$\begin{aligned} \mathbf{a}^{opt} &= \arg\min_{\mathbf{a}} \|\mathbf{v} - \sum_{i=1}^{n} a_i \mathbf{e}_i\|^2 \\ &\text{ b ec } \sum_{i=1}^{n} a_i^2 \leq 1 \end{aligned} \tag{3}$$

S ch a c aı ed eg e ı ı ca ed [6] a d ca be ed b e e ıca e h d [7]. Whe e e ch c aı ed e a ı e ac ı , he c e cıe f he e ac ed e a ı f he ab e e a e a e 1, 0, 0, 0. Thı h ha c aı ed e a ı e ac ı ca ea e he b e . F e de aı e a ı e ac ı a g ı h , ea e efe [8].

3 Mining Hidden Networks on the DBLP Data

I hı a, e ee e 1 e a e ba ed DBLP (Dıgı a Bıb ı- g a h & Lıb a P ec) da a. The DBLP e e (h ://db ı- ıe .de/) ıde bıb ı g a hıc ı f a ı a c e cıe ce a a d - ceedı g . I ı de e e ha 500000 a ıc e a d e ha 1000 dı e e c fe e ce (b Ma 2004).

Ta ı g he a h ı DBLP a b ec , he e a a e ı ı e e- a ı be ee he . A h b ı h a e ı dı e e ce c fe e ce . If e ea ha a h b ı h a e ( ) ı he a e c fe e ce a e ı d f e- a ı , he e 1000 c fe e ce ıde 1000 dı e e e a ı . Gı e e a e (*e.g.*, a g f a h), e e ı e ı d h e ac a e e a ı ı g ch e a e a d d a he he g ı he e a ı . The e ac ed e a ı ca be ı e e ed a he g f a h ha ha e a ce a ı ı d f ı ı a ı e e .

3.1 Data Preparation and Graph Generation

The DBLP e e ıde a he da a ı he XML f a a e a ı e DTD. We e ac ed he ı f a ı f a h , a e a d c fe e ce.

We ge e a e dı e e ı d f g a h (cıa e) ba ed he e - ac ed ı f a ı . F each ceedı g, e c c a g a h ı h e ea che a he de , hıch ı ca ed , , , , he eaf e . If e ea che ha e a e ( ) ı hı ceedı g, he edge be ee he c e dı g de ı e 1. O he ı e, ı ı e 0. F each c fe e ce, e add he ceed- ı g g a h f he a e c fe e ce e ea , hıch ı ca ed , , he eaf e . F ı a , e ch e he 70 c fe e ce g a h ba ed he be f dı ı c a h ı ha c fe e ce.

E e c fe e ce g a h e ec he e a ı hı be ee he e ea che e- aı ı g a ce a ı e ea ch a ea. Ge e a , ıf e ea che a e c ec ed b a edge ı he c fe e ce g a h, he a ha e he a e e ea ch ı e e .

F each g a h, e a ı e he edge eıgh b dı ıdı g he a ı eıgh ı he h e g a h. The e ı g eıgh ha a a ge [0, 1]. The g ea e he eıgh ı , he ge he e a ı ı .

3.2 Experiment Results

I hı e e ı e , e ıde he e ı h e e ıe (e g f e ea che) e a ı e ıf a g ı h ca ca e he hıdde e a ı be ee he e ea che . We h e e a e ı hı a e , ea e efe [8] f e e a e e ıe .

Experiment 1. I he ca e, he e a e e ıe ıded b he e .

1. Phı ı S. Y , Ra e h Ag a a , Ha -Pe e K ıege , Padh aıc S h, Bı g Lı , Ped D ı g .

2. Phı ı S. Y , Ra e h Ag a a , Ha -Pe e K ıege , Hec Ga cıa-M ı a, Da ıd J. DeWı , Mıchae S eb a e .

B h f he e ıe c a ı 6 e ea che . The h ee e ea che a e he a e ı he e ıe .

Table 1. Coefficients of different conference graphs for two queries (sorted on the coefficients)

Query 1		Query 2	
Conference	Coefficient	Conference	Coefficient
KDD	0.949	SIGMOD	0.690
SIGMOD	0.192	ICDE	0.515
ICDE	0.189	VLDB	0.460
VLDB	0.148	KDD	0.215

Tab e 1 h he c e cıe f he e ac ed e a ı f he e ıe . KDD ı a da a ı ı g c fe e ce, a d hıgh eıgh he KDD g a h ı dıca e he c ı e e da a ı ı g. O he he ha d, SIGMOD, VLDB a d ICDE a e h ee da aba e c fe e ce . Hıgh eıgh he e c fe e ce g a h ı dıca e he c ı e e da aba e a ea. The e ac ed e a ı f e 1 ha KDD g a h ı h eıgh ı g 1, hıch e ha he e ea che ı e 1 ha e c ı e e da a ı ı g. F e 2, he e ac ed e a ı e h e e ea che ha e c ı e e da aba e.

Table 2. Researchers' activities in conferences

Researcher	KDD	ICDE	SIGMOD	VLDB
Philip S. Yu	7	15	10	11
Rakesh Agrawal	6	10	13	15
Hans-Peter Kriegel	7	9	11	8
Padhraic Smyth	10	1	0	0
Bing Liu	8	1	0	0
Pedro Domingos	8	0	2	0
Hector Garcia-Molina	0	15	12	12
David J. DeWitt	1	4	20	16
Michael Stonebraker	0	12	19	15

Table 3. Combined Coefficients

Conference Name	Coefficient
SIGMOD	0.586
KDD	0.497
ICDE	0.488
VLDB	0.414

Whı e e e a ı e he b ıca ı f he e e ea che he e f c fe - e ce a ı ed ı Tab e 2, e c ea ee he e ac ed e a ı ea ca e he e a ıc e a ı be ee he e ea che ı he e ıe .

F he e, ı h he e ac ed e a ı g a h, e a ıed he c ı ı g a g ı h [8] a d b aı ed he c e dı g c ı ıe . F each e , e ı e e a e c ı be :

1. C. . . . 1 f . . e. 1: A e a. de. T . hı 1. , Bı. g Lı , Cha. C. Agga. a , De. . ı Sha ha, Ea . . . J. Ke. gh, ...
2. C. . . . 1 f . . e. 2: A f. . Ke e. , A . E Abbadı, Be. g Chı O. ı, Be. . ha. d Seege. , Ch. ı . . Fa . . . . , ...

Le . . ee ha ı ha. e. ıf e . . . b ı he . . h. ee . a e ı . . e e. . The e . ac ed . e a ı . ı . h . . ı Tab e 3. The e . ac ed . e a ı . . ea ca . e he . a. ea (da a . ı. ı. g a. d dababa e) ı hıch he e . e ea. che. . a. e ı. e. e ed.

4 Conclusions

Dı e. e. f. . . . . cıa . e . . a. a . ı . dıe , e a . . e ha he. e e ı . ı e, he e. . ge. e. . . . cıa . e . . . , a. d he . . hı ıca ed c. . bı. a ı . . . f . ch he e. . ge. e. . . . cıa . e . . . . a ge. e. a e ı . . . a. . e . e a ı . . hı . ha . a be e. . . e. ' ı f. . . a ı . . eed. The. ef . e, . . a . . ach . . . cıa . e . . a. a . ı a. d c. . . . ı . ı. ı. g . e . e e. . a. a . . hıf ı . e h. d. . g f. . . he . adı ı . a . . e, a. hıf f. . . ı. g e-. e . . , . e. -ı. de e. de. a. a . ı . . . ı-. e . . , . e. -de e. da. , a. d e. -ba ed a. a . ı . O . a. g . e. f . . ch a. hıf ı c ea. : multiple, heterogeneous social networks are ubiquitous in the real world and they usually *jointly* affect people's social activities.

Ba ed . . . ch a hı . . . h , e . . ed . a . e . e h. d. . g a. d a . e a g. . ı h f . . e a ı . e . ac ı . . W ı h. ch e. -de e. de . e a ı . e . ac ı . a. d c. . . . ı . ı. ı. g, . . e a. d . b e . e a. ıc a. e ca . ed e ec ı e . I ı e ec ed ha he e. -ba ed . e a ı . e . ac ı . a. d c. . . . ı . ı. ı. g . d gı e . ı e . a . . f . e. ıa . e a . ıca ı . . ı . . cıa . e . . a. a . ı .

References

1. Milgram, S.: The small world problem. Psychology Today **2** (1967) 60–67
2. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge, UK (1994)
3. Schwartz, M.F., Wood, D.C.M.: Discovering shared interests using graph analysis. Communications of the ACM **36** (1993) 78–89
4. Kautz, H., Selman, B., Milewski, A.: Agent amplified communication. In: Proceedings of AAAI-96. (1996) 3–9
5. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press (2001) 57–66
6. Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements of Statistical Learning. Springer-Verlag (2001)
7. Bjorck, A.: Numerical Methods for Least Squares Problems. SIAM (1996)
8. Cai, D., Shao, Z., He, X., Yan, X., Han, J.: Mining hidden community in heterogeneous social networks. Technical report, Computer Science Department, UIUC (UIUCDCS-R-2005-2538, May, 2005)

Evaluating the Correlation Between Objective Rule Interestingness Measures and Real Human Interest

Deborah R Carvalho[1,3], Alex A. Freitas[2], and Nelson Ebecken[3]

[1] Universidade Tuiuti do Paraná (UTP), Brazil
deborah@utp.br
[2] Computing Laboratory University of Kent, CT2 7NF, UK
A.A.Freitas@kent.ac.uk
[3] COPPE/ Universidade Federal do Rio de Janeiro, Brazil
nelson@ntt.ufrj.br

Abstract. In the last few years, the data mining community has proposed a number of objective rule interestingness measures to select the most interesting rules, out of a large set of discovered rules. However, it should be recalled that objective measures are just an *estimate* of the true degree of interestingness of a rule to the user, the so-called real human interest. The latter is inherently subjective. Hence, it is not clear how effective, in practice, objective measures are. More precisely, the central question investigated in this paper is: "how effective objective rule interestingness measures are, in the sense of being a good estimate of the true, subjective degree of interestingness of a rule to the user?" This question is investigated by extensive experiments with 11 objective rule interestingness measures across eight real-world data sets.

1 Introduction

Data mining essentially consists of extracting *interesting* knowledge from real-world data sets. However, there is no consensus on how the interestingness of discovered knowledge should be measured. Indeed, most of the data mining literature still avoids this thorny problem and implicitly interprets "interesting" as meaning just "accurate" and sometimes also "comprehensible". Although accuracy and comprehensibility are certainly important, they are not enough to measure the real, *subjective* interestingness of discovered knowledge *to the user*. Consider, e.g., the classic example of the following rule: IF (patient is pregnant) THEN (patient is female). This rule is very accurate and comprehensible, but it is *not* interesting, since it represents an obvious pattern. As a real-world example, [8] reports that less than 1% of the discovered rules were found to be interesting to medical experts. It is also possible that a rule be interesting to the user even though it is not very accurate. For instance, in [9] rules with an accuracy around 40%-60% represented novel knowledge that gave new insights to medical doctors. Hence, there is a clear motivation to investigate the relationship between rule interestingness measures and the subjective interestingness of rules to the user – *an under-explored topic in the literature*.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 453 – 461, 2005.

Rule interestingness measures can be classified into two broad groups: user-driven (subjective) and data-driven (objective) measures. User-driven measures are based on comparing discovered rules with the previous knowledge or believes of the user. A rule is considered interesting, or novel, to the extent that it is different from the user's previous knowledge or believes. User-driven measures have the advantage of being a direct measure of the user's interest in a rule, but they have a twofold disadvantage. First, they require, as input, a specification of the user's believes or previous knowledge – a very time-consuming task to the user. Second, they are strongly domain-dependent and user-dependent. To avoid these drawbacks, the literature has proposed more than 40 data-driven rule interestingness measures [5], [7], [3]. These measures estimate the degree of interestingness of a rule to the user in a user-independent, domain-independent fashion, and so are much more generic. Data-driven measures have, however, the disadvantage of being an indirect *estimate* of the true degree of interestingness of a rule to the user, which is an inherently *subjective* interestingness.

This begs a question rarely addressed in the literature: *how effective data-driven rule interestingness measures are, in the sense of being a good estimate of the true, subjective degree of interestingness of a rule to the user*? The vast majority of works on data-driven rule interestingness measures ignore this question because they do not even show the rules to the user. A notable exception is the interesting work of [5], which investigates the effectiveness of approximately 40 data-driven rule interestingness measures, by comparing their values with the subjective values of the user's interest – what they called *real human interest*. Measuring real human interest involves showing the rules to the user and ask her/him to assign a subjective interestingness score to each rule. Therefore, real human interest should not be confused with the above-mentioned user-driven rule interestingness measures.

This paper follows the same general line of research. We investigate the effectiveness of 11 data-driven rule interestingness measures, by comparing them with the user's subjective real human interest. Although we investigate a smaller number of rule interestingness measures, this paper extends the work of [5] by presenting results for eight data sets, whereas [5] did experiments with just one medical data set, a limitation from the point of view of generality of the results.

2 Objective (Data-Driven) Rule Interestingness Measures

This work involves 11 objective rule interestingness measures – all of them used to evaluate classification rules. Due to space limitations we mention here a brief definition of each of those measures – which are discussed in more detail in the literature. The measures defined by formulas (1)–(8) [5], [7] are based on the coverage and accuracy of a rule. Their formulas are expressed using a notation where A denotes the rule antecedent; C denotes the rule consequent (class); $P(A)$ denotes the probability of A – i.e., the number of examples satisfying A divided by the total number of examples; $P(C)$ denotes the probability of C; "$\neg A$" and "$\neg C$" denote the logical negation of A and C. The measures defined by formulas (9)-(11) [2] use the same notation of A

and C to denote a rule's antecedent and consequent, but they also involve heuristic principles based on variables other than a rule's coverage and accuracy.

The Attribute Surprisingness measure – formula (9) – is based on the idea that the degree of surprisingness of an attribute is estimated as the inverse of its information gain. The rationale for this measure is that the occurrence of an attribute with a high information gain in a rule will not tend to be surprising to the user, since users often know the most relevant attributes for classification. However, the occurrence of an attribute with a low information gain in a rule tends to be more surprising, because this kind of attribute is usually considered little relevant for classification. In formula (9), A_i denotes the attribute in the i-th condition of the rule antecedent A, m is the number of conditions in A, and #classes is the number of classes.

$$\Phi\text{-Coefficiente} = (P(A,C)-P(A)P(C))/\sqrt{P(A)P(C)(1-P(A))(1-P(C))} \tag{1}$$

$$\text{Odds Ratio} = P(A,C)P(\neg A,\neg C)/P(A,\neg C)P(\neg A,C) \tag{2}$$

$$\text{Kappa}=(P(A,C)+P(\neg A,\neg C)-P(A)P(C)-P(\neg A)P(\neg C)) / (1-P(A)P(C)-P(\neg A)P(\neg C)) \tag{3}$$

$$\text{Interest} = P(A,C)/(P(A)*P(C)) \tag{4}$$

$$\text{Cosine} = P(A,C) / \sqrt{(P(A)*P(C))} \tag{5}$$

$$\text{Piatetsky-Shapiro's} = P(A,C)-P(A)P(C) \tag{6}$$

$$\text{Collective Strength} = ((P(A,C)+P(\neg A,\neg C))/(P(A)P(C)+P(\neg A)P(\neg C))) * ((1-P(A)P(C) - P(\neg A)P(\neg C))/(1-P(A,C)-P(\neg A,\neg C)) \tag{7}$$

$$\text{Jaccard} = P(A,C) / (P(A)+ P(C) - P(A,C) \tag{8}$$

$$\text{Attribute Surprisingness} = 1 - ((\sum_{i=1}^{m} \text{InfoGain}(A_i) / m) / \log_2(\#\text{classes})) \tag{9}$$

$$\text{MinGen} = N / m \tag{10}$$

$$\text{InfoChange-ADT} = I^{AB1} - I^{ABo} \tag{11.1}$$

$$I^{ABo} = (- Pr(X|AB) \log_2 Pr(X|AB) + (- Pr(\neg X |AB) \log_2 Pr(\neg X |AB))) \tag{11.2}$$

$$I^{AB1} = - Pr(X|AB) [\log_2 Pr(X|A) + \log_2 Pr(X|B)] - Pr(\neg X |AB) [\log_2 Pr(\neg X|A)+ \log_2 Pr(\neg X|B)] \tag{11.3}$$

The MinGen measure – formula 10 –considers the minimum generalizations of the current rule r and counts how many of those generalized rules predict a class different from the original rule r. Let m be the number of conditions (attribute-value pairs) in the antecedent of rule r. Then rule r has m minimum generalizations. The k-th minimum generalization of r, k=1,...m, is obtained by removing the k-th condition from r. Let C be the class predicted by the original rule r (i.e., the majority class among the examples covered by the antecedent of r) and C_k be the class predict by the k-th

minimum generalization of r (i.e., the majority class of the examples covered by the antecedent of the k-th minimum generalization of r). The system compares C with each C_k, k=1,…, m, and N is defined as the number of times where C is different from C_k.

InfoChange-ADT (Adapted for Decision Trees) is a variation of the InfoChange measure proposed by [4]. Let $A \rightarrow C$ be a common sense rule and $A, B \rightarrow \neg C$ be an exception rule. The original InfoChange measure computes the interestingness of an exception rule based on the amount of change in information relative to common sense rules. In formulas (11.1), (11.2) and (11.3), I^{ABo} denotes the number of bits required to describe the specific rule $AB \rightarrow C$ in the absence of knowledge represented by the generalized rules $A \rightarrow C$ and $B \rightarrow C$, whereas I^{AB1} is the corresponding number of bits when the relationship between C and AB is rather described by the two rules $A \rightarrow C$ and $B \rightarrow C$. One limitation of the original InfoChange measure is that it requires the existence of a pair of exception and common sense rules, which is never the case when converting a decision tree into a set of rules – since the derived rules have mutually exclusive coverage. In order to avoid this limitation and make InfoChange useful in our experiments, the new version InfoChange-ADT is introduced in this paper, as follows. A path from the root to a leaf node corresponds to an exception rule. The common sense rule for that exception rule is produced by removing the condition associated with the parent node of the leaf node. This produces a common sense rule which is "the minimum generalization" of the exception rule. Even with this modification, InfoChange-ADT still has the limitation that its value cannot always be computed, because sometimes the minimum generalization of an exception rule predicts the same class as the exception rule, violating the conditions for using this measure.

For all the 11 rule interestingness measures previously discussed, the higher the value of the measure, the more interesting the rule is estimated to be.

3 Data Sets and Experimental Methodology

In order to evaluate the correlation between objective rule interestingness measures and real, subjective human interest, we performed experiments with 8 data sets. Public domain data sets from the UCI data repository are not appropriate for our experiments, simply because we do not have access to any user who is an expert in those data sets. Hence, we had to obtain real-world data sets where an expert was available to subjectively evaluate the interestingness of the discovered rules. Due to the difficult of finding available real-world data and expert users, our current experiments involved only one user for each data set. This reduces the generality of the results in each data set, but note that the overall evaluation of each rule interestingness measure is (as discussed later) averaged over 8 data sets and over 9 rules for each data set, i.e. each of the 11 measures is evaluated over 72 rule-user pairs. The 8 data sets are summarized in Table 1. Next, we describe the five steps of our experimental methodology.

Table 1. Characteristics of data sets used in the experiments

Data Set	Nature of Data	# Examp.	# Attrib.
CNPq1	Researchers' productivity (# publications), data from the Brazilian Research Council (CNPq)	5690	23
ITU	Patients in Intensive Care Unit	7451	41
UFPR-CS	Students' performance in comp. sci. admiss. exam	1181	48
UFPR-IM	Students' performance in info. manag. admis. exam	235	48
UTP-CS	Comp. Sci. students' end of registration	693	11
Curitiba	Census data for the city of Curitiba, Brazil	843	43
Londrina	Census data for the city of Londrina, Brazil	4115	42
Rio Branco	Census data for city of Rio Branco do Ivai, Brazil	223	43

Step 1 – Discovery of classification rules using several algorithms
We applied, to each data set, 5 different classification algorithms. Three of them are decision-tree induction algorithms (variants of C4.5 [6]), and two are genetic algorithms (GA) that discover classification rules. In the case of the decision tree algorithms, each path from the root to a leaf node was converted into an IF-THEN classification rule as usual [6]. A more detailed description of the 5 algorithms can be found in [1], where they are referred to as default C4.5, C4.5 without pruning, "double C4.5", "Small-GA", "Large-GA". The Rule Interestingness (RI) measures were applied to each of the discovered rules (after all the classification algorithms were run), regardless of which classification algorithm generated that rule.

Step 2 – Ranking all rules based on objective rule interestingness measures
For each data set, all classification rules discovered by the 5 algorithms are ranked based on the values of the 11 objective RI measures, as follows. First, for each rule, the value of each of the 11 RI measures is computed. Second, for each RI measure, all discovered rules are ranked according to the value of that measure. I.e., the rule with the best value of that RI measure is assigned the rank number 1, the second best rule assigned the rank number 2, and so. This produces 11 different rankings for the discovered rules, i.e., one ranking for each RI measure. Third, we compute an *average* ranking over the 11 rankings, by assigning to each rule a rank number which is the *average* of the 11 rank numbers originally associated with that rule. This average rank number is then used for the selection of rules in the next step.

Step 3 – Selection of the rules to be shown to the user
Table 2 shows, for each data set, the total number of rules discovered by all the 5 algorithms applied to that data set. It is infeasible to show a large number of discovered rules to the user. Hence, we asked each user to evaluate the subjective degree of interestingness of just 9 rules out of all rules discovered by all algorithms. The set of 9 rules showed to the user consisted of: (a) the three rules with the lowest rank number (i.e., rules with rank 1, 2, 3, which were the three most interesting rules according to the objective RI measures); (b) the three rules with the rank number closest to the median rank (e.g., if there are 15 rules, the three median ranks would be 7, 8, 9); and (c) the three rules with the highest rank number (least interesting rules). The selection of rules with the lowest, median and highest rank numbers creates three distinct groups of rules which ideally should have very different user-specified interestingness scores. The correlation measure calculated over such a broad range of different

objective ranks is more reliable than the correlation measure that would be obtained if we selected instead 9 rules with very similar objective ranks.

Step 4 – Subjective evaluation of rule interestingness by the user
For each data set, the 9 rules selected in step 3 were shown to the user, who assigned a subjective degree of interestingness to each rule. The user-specified score can take on three values, viz.: <1> – the rule is not interesting, because it represents a relationship known by the user; <2> – the rule is somewhat interesting, i.e., it contributes a little to increase the knowledge of the user; <3> – the rule is truly interesting, i.e., it represents novel knowledge, previously unknown by the user.

Step 5 – Correlation between objective and subjective rule interestingness
We measured the correlation between the rank number of the selected rules – based on the *objective* RI measures – and the *subjective* RI scores – <1>, <2>, <3> – assigned by the user to those rules. As a measure of correlation we use the Pearson coefficient of linear correlation, with a value in [–1...+1], computed using SPSS.

Table 2. Total number of discovered rules for each data set

Data Set:	CNPq1	ITU	UFPR-CS	UFPR-IM	UTP-CS	Curitiba	Londrina	Rio Branco do Ivai
# Rules:	20,253	6,190	1,345	232	2,370	1,792	1,261	486

4 Results

Table 3 shows, for each data set, the correlation between each objective RI measure and the corresponding subjective RI score assigned by the user. These correlations are shown in columns 2 through 9 in Table 3, where each column corresponds to a data set. To interpret these correlations, recall that the lower the objective rank number the more interesting the rule is *estimated to be*, according to the objective RI measure; and the higher the user's subjective score the more interesting the rule *is to the user*. Hence, an ideal objective RI measure should behave as follows. When a rule is assigned the best possible subjective score (<3>) by the user, the RI measure should assign a low rank number to the rule. Conversely, when a rule is assigned the worst possible subjective score (<1>) by the user, the RI measure should assign a high rank number to the rule. Therefore, the closer the correlation value is to –1 the more effective the corresponding objective RI measure is in *estimating the true degree of interestingness of a rule to the user*. In general a correlation value ≤ –0.6 can be considered a strong negative correlation, which means the objective RI measure is quite effective in estimating the real human interest in a rule. Hence, in Table 3 all correlation values ≤ –0.6 are shown in bold.

In columns 2 through 9 of Table 3, the values between brackets denote the ranking of the RI measures for each data set (column). That is, for each data set, the first rank (1) is assigned to the smallest (closest to –1) value of correlation in that column, the second rank (2) is assigned to the second smallest value of correlation, etc. Finally, the last column of Table 3 contains the average rank number for each RI measure –

i.e., the arithmetic average of all the rank numbers for the RI measure across all the data sets. The numbers after the symbol "±" are standard deviations.

Two cells in Table 3 contain the symbol "N/A" (not applicable), rather than a correlation value. This means that SPSS was not able to compute the correlation in question because the user's subjective RI scores were constant for the rules evaluated by the user. This occurred when only a few rules were shown to the user. In general each correlation was computed considering 9 rules selected shown to the user, as explained earlier. However, in a few cases the value of a given objective RI measure could not be computed for most selected rules, and in this case the rules without a value for an objective RI measure were not considered in the calculation of the correlation for that measure. For instance, the N/A symbol in the cell for InfoChange-ADT and data set UFPR-CS is explained by the fact that only 2 out of the 9 selected rules were assigned a value of that objective RI measure, and those two rules had the same subjective RI score assigned by the user.

Table 3. Correlations between objective rule interestingness measures and real human interest; and ranking of objective rule interestingness measures based on these correlations

Rule interestingness measure	Data Set								Avg. Rank
	ITU	UFPR-CS	UTP-CS	Curitiba	UFPR-IM	Londrina	CNPq1	Rio Bran	
Φ-Coefficient	**-0.63** (1)	**-0.91** (4)	**-0.69** (7)	-0.17 (5)	**-0.97** (2)	0.01 (4)	-0.48 (4)	0.45 (10)	4.63 ±2.8
Infochange-ADT (*)	-0.18 (10)	N/A	-0.17 (11)	**-0.70** (1)	**-1.00** (1)	-0.54 (2)	0.15 (8)	**-1.00** (1)	4.86 ±4.6
Kappa	-0.44 (6)	**-0.94** (3)	**-0.74** (5)	-0.12 (6)	**-0.87** (4)	0.12 (5)	-0.18 (7)	-0.56 (3)	4.88 ±1.5
Cosine	-0.55 (3)	**-0.79** (6)	**-0.93** (2)	-0.49 (2)	**-0.81** (7)	0.37 (8)	**-0.64** (1)	0.79 (11)	5.00 ±3.6
Piatesky Shapiro	-0.45 (5)	**-0.95** (1)	**-0.68** (8)	-0.09 (9)	**-0.87** (5)	0.19 (7)	-0.49 (3)	-0.55 (4)	5.25 ±2.7
Interest	-0.40 (8)	**-0.77** (7)	**-0.85** (3)	-0.44 (3)	**-0.87** (6)	**-0.61** (1)	0.28 (9)	-0.22 (7)	5.50 ±2.8
Collective Strength	-0.44 (7)	**-0.94** (2)	**-0.66** (9)	-0.10 (7)	**-0.88** (3)	0.19 (6)	0.35 (10)	-0.56 (2)	5.75 ±3.1
Jaccard	-0.49 (4)	**-0.69** (8)	**-0.93** (1)	-0.10 (8)	-0.30 (9)	0.41 (9)	-0.45 (5)	-0.52 (5)	6.13 ±2.9
Odds Ratio	-0.59 (2)	**-0.91** (5)	**-0.85** (4)	-0.28 (4)	N/A	0.48 (10)	0.43 (11)	0.19 (9)	6.43 ±3.5
MinGen	-0.36 (9)	**-0.60** (9)	**-0.71** (6)	0.00 (10)	0.36 (10)	-0.22 (3)	-0.53 (2)	-0.23 (6)	6.88 ±3.1
Attsurp	0.42 (11)	-0.46 (10)	-0.54 (10)	0.63 (11)	**-0.62** (8)	0.59 (11)	-0.37 (6)	-0.10 (8)	9.38 ±1.9

(*) Although InfoChange-ADT obtained the second best rank overall, it was not possible to compute the value of this measure for many discovered rules (see text).

As shown in Table 3, the strength of the correlation between an objective RI measure and the user's subjective RI score is quite dependent on the data set. In three data sets – namely UFPR-CS, UTP-CS and UFPR-IM – the vast majority of the objective RI measures were quite effective, having a strong correlation (≤ –0.6, shown in bold) with the user's true degree of interestingness in the rules. On the other hand, in each of the other five data sets there was just one objective RI measure that was effective, and in most cases the effective measure (with correlation value shown in bold) was different for different data sets. Correlation values that are very strong (≤ –0.9) are rarer in Table 3, but they are found for five RI measures in the UFPR-CS data set, and for one or two RI measures in three other data sets.

Consider now the average rank number of each measure shown in the last column of Table 3. The RI measures are actually in increasing order of rank number, so that, overall, across the eight data sets, the most effective RI measure was the Φ-Coefficient, with an average rank of 4.63. However, taking into account the standard deviations, there is no significant difference between the average rank of Φ-Coefficient and the average rank of the majority of the measures. The only measure which performed significantly worse than Φ-Coefficient was Attribute Surprisingness, the last in the average ranking.

There is, however, an important caveat in the interpretation of the average ranking of InfoChange-ADT. As explained earlier, there are several rules where the value of this RI measure cannot be computed. More precisely, out of the 9 rules selected to be shown to the user for each data set, the number of rules with a value for InfoChange-ADT varied from 2 to 5 across different data sets. This means that the average rank assigned to InfoChange-ADT is less reliable than the average rank assigned to other measures, because the former was calculated from a considerably smaller number of samples (rules). In particular, the correlation value of InfoChange-ADT was –1 (the best possible value) in two data sets, viz. UFPR-IM and Rio Branco, and in both data sets only 2 out of the 9 selected rules had a value for InfoChange-ADT.

5 Conclusions and Future Research

The central question investigated in this paper was: "how effective objective rule interestingness measures are, in the sense of being a good estimate of the true, subjective degree of interestingness of a rule to the user?" This question was investigated by measuring the correlation between each of 11 objective rule interestingness measures and real human interest in rules discovered from 8 different data sets. Overall, 31 out of the 88 (11 × 8) correlation values can be considered strong (correlation ≥ 60%). This indicates that objective rule interestingness measures were effective (in the sense of being good estimators of real human interest) in just 35.2% (31 / 88) of the cases. There was no clear "winner" among the objective measures – the correlation values associated with each measure varied considerably across the 8 data sets.

A research direction would be to try to predict which objective rule interestingness measure would be most correlated with real human interest for a given target data set, or to predict the real human interest in a rule using a combination of results from different objective measures. This could be done, in principle, using a meta-learning framework, mining data from previously-computed values of the correlation between

objective interestingness measures and subjective human interest for a number of rules that have been previously evaluated by a given user.

References

[1] Carvalho, D.R.; Freitas, A.A. Evaluating Six Candidate Solutions for the Small-Disjunct Problem and Choosing the Best Solution via Meta Learning. *AI Review* 24(1), 61-98, 2005

[2] Carvalho, D.R.; Freitas, A.A.; Ebecken, N.F. (2003) A Critical Review of Rule Surprisingness Measures. Proc. 2003 Int. Conf. on Data Mining, 545-556. WIT Press.

[3] Hilderman, R.J.; Hamilton H.J. *Knowledge Discovery Measures of Interest*. Kluwer, 2001.

[4] Hussain, F.; Liu, H.; Lu, H. Exception Rule Mining with a Relative Interestingness Measure. PAKDD-2000, LNAI 1805, 86-96. Springer-Verlag.

[5] Ohsaki, M., Kitaguchi, S., Okamoto, K., Yokoi, H. Yamaguchi, T. Evaluation of rule interestingness measures with a clinical dataset on hepatitis. *Knowledge Discovery in Databases: PKDD 2004, LNAI 3202*, 362-373. Springer-Verlag, 2004

[6] Quinlan, J.R. *C4.5: programs for machine learning*. Morgan Kaufmann. 1993.

[7] Tan, P.N.; Kumar, V. and Srivastava, J. Selecting the right interestingness measure for association patterns. *Proc. ACM SIGKDD KDD-2002*. ACM Press, 2002

[8] Tsumoto, S. Clinical knowledge discovery in hospital information systems. *Principles of Data Mining and Knowledge Discover, PKDD-2000*, 652-656. Springer-Verlag, 2000.

[9] Wong, M.L. and Leung, K.S. *Data mining using grammar-based genetic programming and applications*. Kluwer, 2000.

A Kernel Based Method for Discovering Market Segments in Beef Meat

Jorge Díez[1], Juan José del Coz[1], Carlos Sañudo[2], Pere Albertí[3], and Antonio Bahamonde[1]

[1] Artificial Intelligence Center, University of Oviedo at Gijón (Asturias), Spain
{jdiez, juanjo, antonio}@aic.uniovi.es
www.aic.uniovi.es
[2] Facultad de Veterinaria, University of Zaragoza, Zaragoza (Aragón), Spain
csanudo@unizar.es
[3] Service of Agriculture and Food Science Research, Zaragoza (Aragón), Spain
palberti@aragon.es

Abstract. In this paper we propose a method for learning the reasons why groups of consumers prefer some food products instead of others. We emphasize the role of groups given that, from a practical point of view, they may represent market segments that demand different products. Our method starts representing people's preferences in a metric space; there we are able to define a kernel based similarity function that allows a clustering algorithm to discover significant groups of consumers with homogeneous tastes. Finally in each cluster, we learn, with a SVM, a function that explains the tastes of the consumers grouped in the cluster. To illustrate our method, a real case of consumers of beef meat was studied. The panel was formed by 171 people who rated 303 samples of meat from 101 animals with 3 different aging periods.

1 Introduction

Consumer preferences for food products address the strategies of industries and breeders, and should be carefully considered when export and commercial policies are designed. In this paper we present a method to deal with data collected from panels of consumers in order to discover groups with differentiated tastes; these groups may constitute significant market segments that demand different kinds of food products. Additionally, our approach studies the factors that could contribute to the success or failure of food products in each segment.

From a conceptual point of view, consumer panels are made up of untrained consumers; these are asked to rate their degree of acceptance or satisfaction about the tested products on a scale. The aim is to be able to relate product descriptions (human and mechanical) with consumer preferences. Nevertheless, the Market is not interested in tastes of individual consumers, the purpose of marketing studies of sensorial data is to discover, if there exist widespread ways to appreciate food products that can be considered as market segments. These segments can be seen as *clusters* of consumers with similar tastes. In this paper, we will show that the similarity of preference criteria of consumers can be computed in a high dimension space; for this purpose, we present here a kernel-based method. To illustrate our method, we used a data set that

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 462 – 469, 2005.

collects the ratings of a panel of beef meat consumers. The panel studied was formed by 171 people rating samples of 303 different kinds of beef meat [1] from different breeds, live weights, and aging periods.

2 Description of the General Approach

The main assumption behind the approach presented in this paper is that we are able to map people's preferences into a metric space in such a way that we can assume some kind of continuity. A first attempt to provide such a mapping would consist in associating, to each consumer, the vector of his or her ratings, taking the set of samples as indexes. However, this is not a wise option since ratings have only a relative meaning, and therefore they cannot assume an absolute role. There is a kind of *batch effect*: a product will obtain a higher/lower rating when it is assessed together with other products that are clearly worse/better. In fact, if we try to deal with sensory data as a regression problem, we will fail [2]; due to this batch effect, the ratings have no numerical meaning: they are only a relative way to express preferences between products of the same session.

To overcome this, instead of ratings, we can assign to each product its ordinal position in the ranking of preferences. Unfortunately, this is not always possible given that, in general, the size of the sample of food prevents panelists from testing all products. Hence, we cannot ask our panelists to spend long periods rating the whole set of food samples. Typically, each consumer only participates in one or a small number of testing sessions, usually in the same day. Notice that tasting a large sample of food may be physically impossible, or the number of tests performed would damage the sensory capacity of consumers. The consequence is that consumers' rankings are not comparable because they deal with different sets of products. Thus, in this case we will codify people preferences by the weighting vector of a linear function (called *preference* or *ranking function*) in a high dimensional space: the space of features where we represent the descriptions of food products. Then, the similarity is defined by means of the kernel attached to the representation map.

Once we have people preferences represented in a metric space, and we have defined a similarity function, then we use a clustering algorithm. Finally, we only need to explain the meaning and implications of each cluster in the context of the food products. For this purpose, we will learn a preference or ranking function from the union of preference judgments expressed by the member of the cluster; this will provide the consensus assessment function of the cluster.

3 Description of the Beef Meat Experiment

To illustrate our method we used a database described in [1]. The data collects the sensory ratings of a panel of beef meat consumers about three aspects: flavour, tenderness, and acceptability.

For this experience, more than 100 animals of 7 Spanish breeds were slaughtered to obtain two kinds of carcasses: lights, from animals with a live weight around 300–350 kg (light); and heavies, from animals at 530–560 kg. The set of animals was uni-

formly distributed by breeds and weights. Additionally, to test the influence of aging in consumers' appreciation, each piece of meat was prepared with 3 aging periods, 1, 7, and 21 days. On the other hand, the 7 breeds used constitute a wide representation of beef cattle. These breeds can be divided into four types: double muscled (DM, one breed), fast growth (FG, two breeds), dual purpose (DP, one breed), and unimproved rustic type (UR, three breeds). In Table 1 for each breed, we show the average percentages of fats, muscle and bone.

Table 1. Carcass compositions of 7 Spanish beef breeds used in the experiment

Breed		Fat %		Bone	Muscle	Intramuscular
Name	Type	inter-muscular	subcutaneous	%	%	fat %
Asturiana Valles	DM	4.77	0.89	16.00	78.34	0.90
Avileña	UR	13.17	3.53	19.25	64.05	2.28
Morucha	UR	12.46	3.46	19.28	64.80	2.10
Parda Alpina	DP	9.65	2.32	20.86	67.17	1.82
Pirenaica	FG	9.02	3.01	17.33	70.63	1.48
Retinta	UR	14.16	4.75	20.89	60.20	2.13
Rubia Gallega	FG	5.73	1.20	16.56	76.52	1.12

Each kind of meat was also described by a panel of 11 trained experts who rate 12 traits of products such as fibrosis, flavor, odor, etc.. In this paper, we considered the average rate of each trait. The characterization of meat samples was completed with 6 physical features describing its texture.

4 Vectorial Representation of Preference Criteria

As was explained above, in order to compare the preference criteria of consumers we need to state a common language. We cannot use for this purpose the ratings assigned by consumers to food products, since they have rated, in general, different sets of samples. Then we are going to induce a reasonable extension of the preferences expressed by each consumer to obtain a function able to capture the pairwise orderings, not the rates. Then we will manage to define similarities in the space of those functions.

Although there are other approaches to learn preferences, we will follow [3, 4, 5]. Then we will try to induce a real *preference*, *ranking*, or *utility function* f from the input space of object descriptions, say $\mathbf{R}^d$, in such a way that it maximizes the probability of having $f(\mathbf{v}) > f(\mathbf{u})$ whenever $\mathbf{v}$ is preferable to $\mathbf{u}$; we call such pairs, *preference judgments*. This functional approach can start from a set of objects endowed with a (usually ordinal) rating, as in regression; but essentially, we only need a collection of preference judgments.

When we have a set of ratings given by a consumer c, we most take into account the session where the ratings have been assessed [6, 7], as was explained in section 2. Thus, for each session we include in the set of preference judgments, PJ_c, the pairs ($\mathbf{v}$, $\mathbf{u}$) whenever consumer c assessed to sample represented by $\mathbf{v}$ a higher rating than to the sample represented by $\mathbf{u}$. In order to induce the ranking function, as in [3], we look for a function F_c: $\mathbf{R}^d \times \mathbf{R}^d \rightarrow \mathbf{R}$ such that

$$\forall \mathbf{x},\mathbf{y} \in \mathbf{R}^d,\ F_c(\mathbf{x},\mathbf{y}) > 0 \Leftrightarrow F_c(\mathbf{x},\mathbf{0}) > F_c(\mathbf{y},\mathbf{0}) \tag{1}$$

Notice that the right hand side of (1) establishes an ordering of functional expressions of a generic couple ($\mathbf{x}$, $\mathbf{y}$) of objects representations. This suggests the definition

$$f_c\colon \mathbf{R}^d \to \mathbf{R},\ f_c(\mathbf{x}) = F_c(\mathbf{x},\mathbf{0}) \tag{2}$$

The idea is then to obtain ranking functions f_c from functions like F_c, as in (2), when F_c fulfils (1). Thus, given the set of preference judgments PJ_c, we can specify F_c by means of the constraints

$$\forall\ (\mathbf{v},\ \mathbf{u}) \in PJ_c,\ F_c(\mathbf{v},\ \mathbf{u}) > 0 \text{ and } F_c(\mathbf{u},\ \mathbf{v}) < 0 \tag{3}$$

Therefore, PJ_c gives rise to a set of binary classification training set to induce F_c

$$E_c = \{(\mathbf{v},\ \mathbf{u},\ +1),\ (\mathbf{u},\ \mathbf{v},\ -1)\colon (\mathbf{v},\ \mathbf{u}) \in PJ_c\} \tag{4}$$

Nevertheless, a separating function for E_c does not necessarily fulfill (1). Thus, we need an additional constraint. So, if we represent each object description $\mathbf{x}$ in a higher dimensional feature space by means of $\phi(\mathbf{x})$, then we can represent pairs ($\mathbf{x}$, $\mathbf{y}$) by $\phi(\mathbf{x})$ - $\phi(\mathbf{y})$. Hence, a classification SVM can induce from E_c a function of the form:

$$F_c(\mathbf{x},\mathbf{y}) = \sum_{s\in S(c)} \alpha_s z_s \left\langle \phi(\mathbf{x}_s^{(1)}) - \phi(\mathbf{x}_s^{(2)}), \phi(\mathbf{x}) - \phi(\mathbf{y}) \right\rangle \tag{5}$$

where $<\mathbf{x}, \mathbf{y}>$ stands for the inner product of vectors $\mathbf{x}$ and $\mathbf{y}$; $S(c)$ is the set of support vectors, notice that they are formed by two d-dimensional vectors $(\mathbf{x}_s^{(1)}, \mathbf{x}_s^{(2)})$, while the scalars z_s represent the class $+1$ or -1. Trivially, F_c fulfils the condition (1). Let us remark that if k is a kernel function, defined as the inner product of two objects represented in the feature space, that is, $k(\mathbf{x},\ \mathbf{y}) = <\phi(\mathbf{x}),\ \phi(\mathbf{y})>$, then the kernel function used to induce F_c is

$$\mathbf{K}(\mathbf{x}_1,\ \mathbf{x}_2,\ \mathbf{x}_3,\ \mathbf{x}_4) = k(\mathbf{x}_1,\ \mathbf{x}_3) - k(\mathbf{x}_1,\ \mathbf{x}_4) - k(\mathbf{x}_2,\ \mathbf{x}_3) + k(\mathbf{x}_2,\ \mathbf{x}_4) \tag{6}$$

Usually it is employed a linear or a simple polynomial kernel; that is, $k(\mathbf{x},\ \mathbf{y}) = \langle \mathbf{x},\ \mathbf{y}\rangle$, or $k(\mathbf{x},\ \mathbf{y}) = (\langle \mathbf{x},\ \mathbf{y}\rangle + 1)^g$, with $g = 2$.

Once we have a function F_c for a consumer c fulfilling (1), then, using (2), a ranking or preference or utility function f_c is given (but for an irrelevant constant) by

$$f_c(\mathbf{x}) = \sum_{s\in S(c)} \alpha_s z_s \left\langle \phi(\mathbf{x}_s^{(1)}) - \phi(\mathbf{x}_s^{(2)}), \phi(\mathbf{x}) \right\rangle = \sum_{s\in S(c)} \alpha_s z_s \left(k(\mathbf{x}_s^{(1)},\mathbf{x}) - k(\mathbf{x}_s^{(2)},\mathbf{x}) \right) \tag{7}$$

Therefore, f_c can be represented by the weight vector $\mathbf{w}^c$ in the higher dimensional space of features such that

$$f_c(\mathbf{x}) = \left\langle \mathbf{w}^c, \phi(\mathbf{x}) \right\rangle, \qquad \mathbf{w}^c = \sum_{s\in S(c)} \alpha_s z_s \left(\phi(\mathbf{x}_s^{(1)}) - \phi(\mathbf{x}_s^{(2)}) \right) \tag{8}$$

Notice that (8) defines the ranking of an object represented by a vector $\mathbf{x}$. This is not an absolute value; its importance is the relative position that gives to $\mathbf{x}$ against to other objects $\mathbf{y}$ in the *competition* for gaining the appreciation of consumer c. Now we only need to define the distance of consumers' preferences. Given that preferences are codified by those weighting vectors, we define the similarity of the preferences of consumer c and c' by the cosine of their weighting vectors. In symbols,

$$\mathtt{similarity}(\mathbf{w}^c,\mathbf{w}^{c'}) = \cos(\mathbf{w}^c,\mathbf{w}^{c'}) = \frac{\langle \mathbf{w}^c,\mathbf{w}^{c'}\rangle}{\|\mathbf{w}^c\| * \|\mathbf{w}^{c'}\|} \tag{9}$$

Given that this definition uses scalar products instead of coordinates of weighting vectors, we can easily rewrite (10) in terms of the kernels used in the previous derivations. The essential equality is:

$$\begin{aligned}\langle \mathbf{w}^c,\mathbf{w}^{c'}\rangle &= \sum_{s\in S(c)}\sum_{l\in S(c')} \alpha_s\alpha_l z_s z_l \langle \phi(\mathbf{x}_s^{(1)}) - \phi(\mathbf{x}_s^{(2)}), \phi(\mathbf{x}_l^{(1)}) - \phi(\mathbf{x}_l^{(2)})\rangle \\ &= \sum_{s\in S(c)}\sum_{l\in S(c')} \alpha_s\alpha_l z_s z_l \mathbf{K}(\mathbf{x}_s^{(1)},\mathbf{x}_s^{(2)},\mathbf{x}_l^{(1)},\mathbf{x}_l^{(2)})\end{aligned} \tag{10}$$

5 Clustering Consumers with Homogeneous Tastes

In the previous section we have associated one data point for each consumer in the space of preference criteria represented by ranking or preference functions. Moreover, we have defined a reasonable similarity measure for preference criteria; now we proceed to look for clusters of consumers with homogeneous tastes. For this purpose, we applied a nonparametric pairwise algorithm [8].

Let $S = (s_{ij})$ be a square matrix where s_{ij} stands for the similarity between data points i and j; in our case, data points are the vectorial representation of the preference criteria of consumers, and similarities are given by equation (9). Then, matrix S is transformed iteratively, following a two step procedure that converges to a two values matrix (1 and 0), yielding a bipartition of the data set into two clusters. Then, recursively, the partition mechanism is applied to each of the resulting clusters represented by their corresponding submatrices. To guarantee that only meaningful splits take places, in [8] the authors provide a cross validation method that measures an index that can be read as a significance level; we will only accept splits which level is above 0.90.

The first step normalizes the columns of S using the L_∞ norm; then the proximities are re-estimated using the Jensen-Shannon divergence. The idea is to formalize that two preference criteria are close (after these two steps) if they were both similar and dissimilar to analogous sets of criteria before the transformation.

6 Experimental Results

In this section, we report the outputs obtained with the database of beef meat consumers. In order to consider significant opinions, we first selected those people involved in our consumers' panel whose ratings gave rise to at least 30 preference judgments; these yielded us to consider a set of 171 panelists that tested from 9 to 14 samples of meat of 101 different animals. The total amount of different samples was 303, since the meat from each animal was prepared with 3 different aging periods: 1, 7, and 21 days. Then the opinions of our panelists can be estimated inducing a preference or ranking function as was explained in section 4. Notice that only such functions can be used in order to compare the preferences of different consumers; in general, two arbi-

Table 2. For clusters of acceptance and tenderness datasets, this table reports the number of preference judgments (PJ), percentage of disagreements, and classification errors achieved into clusters with their own ranking or preference function, and using the function of the other cluster

Dataset	cluster	PJ	disagreements %	classification errors using function own %	other %
acceptance	left	1927	16.19	19.20	50.96
	right	2150	17.07	21.12	54.95
tenderness	left	2487	15.96	19.38	61.98
	right	2432	15.21	19.59	61.06

trary consumers have not tested samples of the same animal prepared with the same aging. However, it is possible to compare the preference functions of any couple of consumers as vectors in a high dimension space following the kernel based method of section 4.

The clustering algorithm [8] returns the trees depicted in Figure 1. Split nodes achieved a confidence level of 91% for tenderness dataset, and 97% for acceptance. The leaves of these trees and the dataset of flavor reached lower confidence levels, and therefore they were rejected.

Fig. 1. Trace of the clustering algorithm. In each node we report the number of consumers

The job of clustering is to compute groups with minimal intra-group and maximal inter-group distances or differences. In our case, the relevance of clusters can be estimated, in part, by the coherence of consumers included into the same cluster, which can be measured by the classification error of the SVM used to compute the ranking or preference function of each cluster. Let us notice that the union of preference judgments of the members of the same cluster has some disagreements; if for each pair of samples we choose the most frequent relative ordering, then about 16% of preference pairs of each cluster express a particular disagreement with the majority opinion of the cluster, see Table 2. However, every preference judgment is included in the training set of each cluster; this sums more than 2000 preference judgments, what means (see equation 4 in section 4) more than 4000 training instances for the corresponding classification sets. When we use a polynomial kernel of degree 2, the errors range from 19.20% to 21.12%; we used this kernel following [2, 6, 7]. Nevertheless, if we apply the induced classification function of each cluster to the other one, then the errors rise to more than 50% in the case of acceptance, and more than 60% in the case of tenderness. Notice that in both cases we are ranking the same samples and these errors can be understood as the probability of reversing the order given by one of such clusters when we use the criteria of the other one. Therefore, 50% of error

means a random classification, and over that threshold means that ranking criteria is approaching the exactly opposite, see Table 2.

In general, it is well known that meat qualities are mainly the result of a set of complex factors. In this study, we are interested in knowing if there are different groups of people who prefer some breeds to others. To gain insight into the meaning of the preference criteria of each cluster, we used the ranking or preference functions to order the samples of meat; then we assessed 10 points to those samples included in the first decile, 9 to the second decile, and so on. Graphical representations of the average points obtained by each breed are shown in Figure 2; notice that the average score of all samples is 5.5. The results are quite the same if we use quartiles instead of deciles or any other division of the relative rankings of each cluster.

In the acceptance dataset (Fig. 2 left), let us emphasize the opposite role played by Retinta and Asturiana breeds: they were first and last (or almost last) in each cluster alternatively. In [6, 7] we used Boolean attributes to include the breed in the description of each sample, and then Retinta and Asturiana were found to be the most relevant Boolean features in order to explain consumer's acceptance of meat. Additionally, these two breeds have significant differences in carcass composition (see Table 1). Notice that Asturiana breed is the only double muscled breed of the sample, and then it has the lowest values in percentages of subcutaneous and inter-muscular fat, and bone; while Retinta is the unimproved rustic breed with the highest percentages of fat and bone. Therefore, there are some reasons so as to assign opposite ratings to samples of these two breeds, although, in general, the final acceptance scorings rely on a complex set of features.

In tenderness dataset (Fig. 2 right), meat from Pirenaica and Retinta breeds are the tenderest for people in left cluster, however they are ranked in low positions in right cluster. We can say exactly the opposite of meat from Asturiana and Parda breeds. Again, Asturiana and Retinta breeds play opposites roles in each cluster.

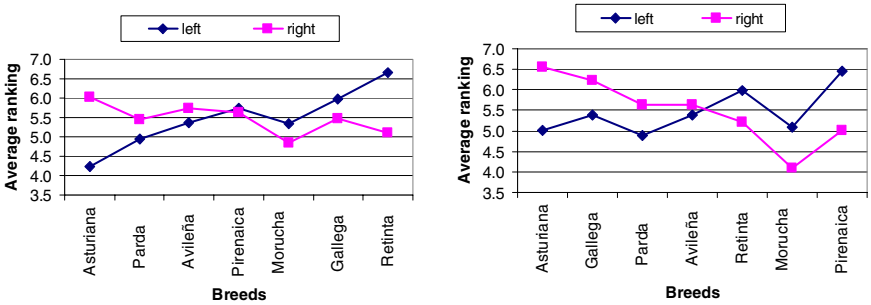


Fig. 2. Average ranking scores for each breed. Acceptance (left). Tenderness (right)

Acknowledgments

We would like to thank: the authors of Spider [9], a MatLab toolbox that includes kernel based algorithms; Thorsten Joachims for his SVM^{light} [10]. Those systems were used in the experiments reported in this paper; and INIA (Instituo Nacional de Inves-

tigación y Tecnología Agraria y Alimentaria of Spain) and Breeders Associations for the financial (Grant SC-97019) and technical support.

References

1. Sañudo, C.; Macie, E.S.; Olleta, J.L.; Villarroel, M.; Panea, B.; Albertí, P.. The effects of slaughter weight, breed type and ageing time on beef meat quality using two different texture devices. Meat Science, 66 (2004), 925–932
2. Díez, J.; Bayón, G. F.; Quevedo, J. R.; del Coz, J. J.; Luaces, O.; Alonso, J.; Bahamonde, A.. Discovering relevancies in very difficult regression problems: applications to sensory data analysis. Proceedings of the European Conference on Artificial Intelligence (ECAI 2004), 993-994
3. Herbrich, R.; Graepel, T.; and Obermayer, K.: Large margin rank boundaries for ordinal regression. In A. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, editors, Advances in Large Margin Classifiers, 115–132. MIT Press, Cambridge, MA, (2000)
4. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD) (2002)
5. Bahamonde, A.; Bayón, G. F.; Díez, J.; Quevedo, J. R.; Luaces, O.; del Coz, J. J.; Alonso, J.; Goyache, F.. Feature subset selection for learning preferences: a case study. Proceedings of the 21st International Conference on Machine Learning, (ICML 2004), 49-56
6. Del Coz, J. J.; Bayón, G. F.; Díez, J.; Luaces, O.; Bahamonde, A.; Sañudo, C.. Trait selection for assessing beef meat quality using non-linear SVM. Proceedings of the Eighteenth Annual Conference on Neural Information Processing Systems (NIPS 2004), 321-328
7. Luaces, O.; Bayón, G.F.; Quevedo, J.R.; Díez, J.; del Coz, J.J.; Bahamonde, A.. Analyzing sensory data using non-linear preference learning with feature subset selection. Proceedings of the 15th European Conference of Machine Learning, (ECML 2004), 286-297
8. Dubnov, S.; El-Yaniv, R.; Gdalyahu, Y.; Schneidman, E.; Tishby, N.; Yona, G.. A New Nonparametric Pairwise Clustering Algorithm Based on Iterative Estimation of Distance Profiles. Machine Learning, 47 (2002), 35–61
9. Weston, J.; Elisseeff, A.; BakIr, G.; Sinz, F.: SPIDER: object-orientated machine learning library. http://www.kyb.tuebingen.mpg.de/bs/people/spider/
10. Joachims, T.. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, (1999)

Corpus-Based Neural Network Method for Explaining Unknown Words by WordNet Senses

Bálint Gábor, Viktor Gyenes, and András Lőrincz*

Eötvös Loránd University, Pázmány P. sétány 1/C, Budapest 1117
{gbalint, gyenesvi}@inf.elte.hu, andras.lorincz@elte.hu
http://nipg.inf.elte.hu/

Abstract. This paper introduces an unsupervised algorithm that collects senses contained in WordNet to explain words, whose meaning is unknown, but plenty of documents are available that contain the word in that unknown sense. Based on the widely accepted idea that the meaning of a word is characterized by its context, a neural network architecture was designed to reconstruct the meaning of the unknown word. The connections of the network were derived from word co-occurrences and word-sense statistics. The method was tested on 80 TOEFL synonym questions, from which 63 questions were answered correctly. This is comparable to other methods tested on the same questions, but using a larger corpus or richer lexical database. The approach was found robust against details of the architecture.

1 Introduction

The Internet is an immensely large database; large amount of domain specific text can be found. Intelligent tools are being developed to determine the meaning of documents, and manually created lexical databases are intended to provide help for such tools. However, manually assembled lexical databases are unable to cover specific, emerging subjects, thus documents may contain words of unknown meanings; words that are not contained in the lexical databases, or the contained meanings do not fit into the context found in the documents. However, the meaning of these words can often be inferred from their contexts of usage. The aim of our method is to explain words that are unknown to a human or machine reader, but are contained in many documents in the same sense.

To achieve this goal, our method looks for WordNet senses that are semantically close to the unknown meaning of the word. We rely on the common practice of measuring the similarity of words based on their contextual features, and designed a neural network architecture by means of three databases as sources of information. The first is WordNet[1], where the words are grouped into synonym sets, called *synsets*. The second source of information that our method exploits is SemCor[2], which is a corpus tagged with WordNet senses. SemCor was used

* Corresponding author

[1] http://www.cogsci.princeton.edu/∼wn/

[2] http://www.cs.unt.edu/∼rada/downloads/semcor/semcor2.0.tar.gz

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 470–477, 2005.

to obtain information on the statistical distributions of the senses of the words. The third database used is the British National Corpus (BNC[3]), a collection of English texts of 100 million words. Our assumption is that the meaning of a word is similar to the meaning of a *sense*, if they appear in the same context, where we define context by the *senses* that they often co-occur with[4].

Testing of any new method requires a controllable benchmark problem. Our method was evaluated on 80 synonym questions from the Test of English as a Foreign Language (TOEFL[5]). The system scored 78.75% (63 correct answers). Many other studies had also chosen this TOEFL benchmark problem: Landauer and Dumais's Latent Semantic Analysis (LSA) [1] is based on co-occurrences in a corpus, and it provides generalization capabilities. It was able to answer 64.4% of the questions correctly. Turney's Pointwise Mutual Information Information Retrieval (PMI-IR) algorithm [2] performed 73.25% on the same set of questions. This is also a co-occurrence based corpus method, which examines noun enumerations. It uses the whole web as a corpus and exploits AltaVista's special query operator, the *NEAR* operator. Terra and Clarke [3] compared several statistical co-occurrence based similarity measures on a one terabyte web corpus, and scored 81.25%. Jarmasz and Szapakovicz constructed a thesaurus-based method [4], which performed 78.75% on these questions. They utilized Roget's Thesaurus to calculate path lengths in the semantic relations graph between two words, from which a semantic similarity measure could be derived.

This paper is organized as follows: Section 2 details the neural network method. Section 3 describes the various test cases and presents the results. Discussion is provided in Section 4, conclusions are drawn in Section 5.

2 Methods

As it was already mentioned, our method exploits three databases (Fig. 1(A)). BNC is used to obtain word co-occurrence statistics. The following estimations are also required: given a synset, how frequently is one of its words used to express that sense, and, on the other hand, if a word is used, how frequently is it used in one of its senses. Database SemCor was used to obtain these statistics. Since all these pieces of information are needed, we only used words and synsets which occur in SemCor at least once. This means 23141 words and 22012 synsets. Later, we extended these sets by adding the trivial synsets, which contained single words. Then we could experiment with 54572 words and 53443 synsets.

2.1 Co-occurrence Measures

The aim of our method is to find semantically close synsets to an unknown word. Two words that occur in similar contexts can be considered as similar in

[3] `http://www.natcorp.ox.ac.uk/`

[4] The words *sense* and *synset* is used interchangeably in this paper.

[5] `http://www.ets.org/`

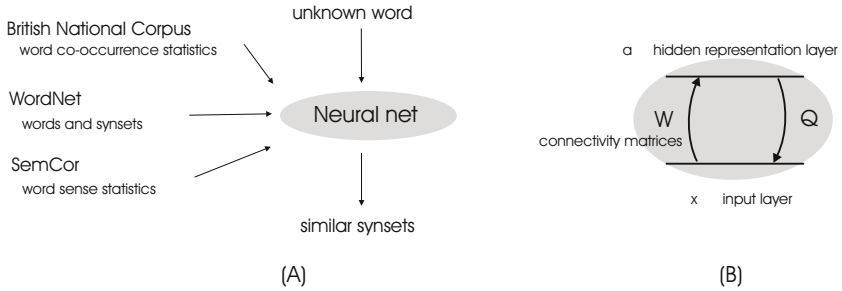


Fig. 1. Databases and reconstruction network architecture (A): Scheme of our method. (B): Basic computational architecture. Input layer (x) and hidden representation (a) are connected by bottom-up (W) and top-down (Q) matrices.

meaning. Therefore, we need to express the measures of co-occurrence between words and synsets; i.e. values indicating how often words and synsets co-occur.

The probability that word w_1 occurs near word w_2 can be estimated as follows: $P(w_1|w_2) = \frac{f(w_1,w_2)}{f(w_2)}$, where $f(w_1, w_2)$ is the number of times w_1 and w_2 co-occur in a 5 wide context window[6] and $f(w)$ is the frequency of word w. We say that w_1 and w_2 are *near words* of each other, if both $P(w_1|w_2)$ and $P(w_2|w_1)$ are high, meaning that w_1 and w_2 are likely to co-occur. The following measure derived from mutual co-occurrences expresses this idea: $N(w_1, w_2) = min(P(w_1|w_2), P(w_2|w_1))$. It is expected that this co-occurrence measure describes the contexts of the words. Given a word w, we call the *near word list* of w is the 100 words w_i for which the $N(w, w_i)$ values are the highest. This near word list can be represented as a *feature vector* of the word, the entries of the vector are the $N(w, w_i)$ values. Then the co-occurrence information about the words can be summarized in a quadratic and symmetric matrix N_W, where the ith row of the matrix is the feature vector of the ith word: $N_W(i, j) = N(w_i, w_j)$, where w_j is the jth word in our vocabulary.

In SemCor, every occurrence of a word is tagged with a WordNet synset that expresses the meaning of the actual occurrence of the word. By counting these tags we can compute the desired probabilities. The probability that for a given word w, the expressed sense is s, can be estimated as $P(s|w) = \frac{f(w,s)}{f(w)}$, where $f(w, s)$ is the frequency of word w in sense s and $f(w)$ is the frequency of word w in any of its senses. We also need the probability that a given sense s is expressed by word w, which can be estimated as: $P(w|s) = \frac{f(w,s)}{f(s)}$, where $f(s)$ is the frequency of sense s, whichever word it is expressed by. These probabilities can also be summarized in matrix forms, denoted by S_W and W_S: $S_W(i, j) = P(s_i|w_j)$ and $W_S(i, j) = P(w_i|s_j)$.

Using the measures introduced, a co-occurrence measure between synsets can be derived. The idea is the following: given a synset s, the *near synsets* of s are the synsets of the near words of the words expressing s. This idea is

[6] Increasing the context window by a factor of 2 had no significant effects.

expressed by the appropriate concatenation of the three matrices introduced above: $N_S = S_W N_W W_S$.

2.2 Reconstruction Networks

The basic reconstruction network model has two neuron layers. Connections bridge these layers. The lower layer is the input layer of the network, and the upper layer is called the hidden or internal representation layer (Fig. 1(B)). The network reconstructs its input by optimizing the hidden representation. For this reason, we call it reconstruction network. Formally, the following quadratic cost function is involved [5]:

$$J(a) = ||x - Qa||_2^2 \ , \tag{1}$$

where Q is the connectivity matrix, x is the input vector, a is the hidden representation vector. The columns of the connectivity matrix can be thought of as basis vectors, which must be linearly combined with the appropriate coefficients so that the combination falls close to the input. The optimization can either be solved directly

$$a = (Q^T Q)^{-1} Q^T x \ , \tag{2}$$

or iteratively

$$\Delta a \propto W(x - Qa) \ , \tag{3}$$

where $W = Q^T$, which can be derived from the negative gradient of cost function (1). The form of (3) is more general than required by (1) but it still suitable as long as WQ is positive definite. Both methods have advantages and disadvantages. Directly solving the optimization returns the exact solution, but might require a considerable amount of memory, while the iterative solution requires less resources, but is computationally intensive.

The reconstruction network described above shall be called 'one-tier' network. We designed both 'one-tier' and 'two-tier' networks for the word-sense reconstruction. The two-tier network has two one-tiers on the top of each other. The internal representation layer of the first tier serves as input for the second tier. There are differences between the two architectures in computation speed and in numerical precision.

The feature vector representation of the context of the unknown word serves as input for the network. In the hidden layer, the neurons represent the candidate synsets. In the one-tier network (Fig. 2(A)), the top-down and bottom-up matrices are defined as follows: $Q = W_S N_S$ and $W = N_S^T S_W$[7]. Thus, in the one-tier method, hidden synset activities are (a) transformed to near synset activities and then (b) the activities of the near words are generated. An illustrative iteration is depicted in Fig. 2(B): The activities in the topmost layer change during the reconstruction of the input word *frog*. It can be seen that only a few activities become high, others remain small. Note the horizontal scale: there were about 23,000 neurons in the topmost layer in this iteration.

[7] However, we found that $\tilde{Q} = N_W W_S$ and $\tilde{W} = \tilde{Q}^T = W_S^T N_W^T$ are simpler, express the same relations and converge faster, thus these were used in the computations.

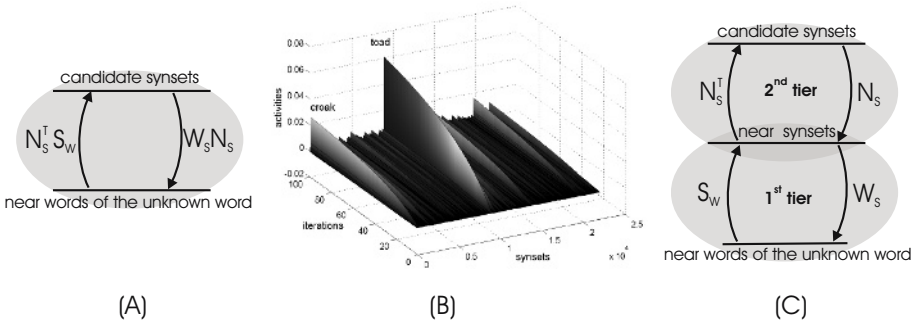


Fig. 2. One-tier and two-tier networks (A) and (C): one and two tier networks. The input of the ith node of the lowest layer is $N(w, w_i)$, where w is the unknown word. $S_W(i,j) = P(s_i|w_j)$, $W_S(i,j) = P(w_i|s_j)$, $N_S = S_W N_W W_S$, where $N_W(i,j) = N(w_i, w_j)$ for all i, j. The *result* of the computation is the synset represented by the highest activity unit of the top layer after running the network. (B): Convergence of the iterative approximation. Input word is *frog*. High activity nodes correspond to sense *toad* and *croak*.

In the 'two-tier' network (Fig. 2(C)) the two steps of the transformation are separated. The nodes in the intermediate layer correspond to the near synsets of the unknown word. The bottom-up and top-down matrices are the S_W and the W_S matrices, respectively. In the second tier the connectivity matrix is N_S in both directions, which contains the synset near values. This captures the idea mentioned in the introduction; the meaning of the unknown word is similar to the meaning of a synset if they have the same near *synsets*.

3 Tests and Results

We tested our method on 80 TOEFL synonym questions, each question consisted of a *question word* (for example *grin*) and four *candidate answer words*, (for example: *exercise, rest, joke, smile*). The task was to find the candidate word that was the most similar in meaning to the question word (*smile*).

To meet our goals, we considered the question word the unknown word. We simulated the situation of the question word being *unknown* by erasing all information about the question word and its meaning from the SemCor statistics and WordNet synsets. After running the network for the context of the question word as input, we examined the activities corresponding to the synsets of the candidate words, and assigned a value to each candidate word equal to the highest activity of the synsets of the candidate word in the upper layer of the network. The candidate word with the highest value was the chosen answer.

In the one-tier network and in the second tier of the two-tier network, the huge number of connections between the nodes required the application of the less precise iterative method. However, the first tier of the two-tier network could be optimized directly, because the connectivity matrices were very sparse.

In some cases we have additional – top-down (TD) – information about the unknown word, for example its part-of-speech or the candidate answer words, this reduces the set of candidate synsets. The implementation of this filtering is simple in our system, since we can simply leave out the unnecessary synsets.

In order to test whether or not using synsets increases the efficiency of the system, we constructed a one tier *control* network which used only words. The input of the network was the same, however, the nodes in the upper layer corresponded to words. The connectivity matrix between the two layers is N_W, as defined in 2.1. The activities of the upper layer were examined; words having similar context as the input word were returned.

The number of correct answers in the various cases can be seen in Table 1. The best result, 63 correct answers (78.75%), was produced with part-of-speech constraint with the 23141 word data set and also without TD constraint with the 54572 word data set, utilizing the one-tier network. However, the best first iterations were achieved by the two-tier network. It can be seen, that iteration has improved the precision in almost all the cases. The control network started from 53 and 54 correct answers for the two word sets and reached 60 and 59 correct answers, respectively, by iterations. These results are considerably smaller than those of without the synsets, which supports our starting assumptions. We should note that if information related to the question words are not deleted from the database, then the number of correct answers is 68, which *amounts to 85%*.

Table 1. Results *No constr*: No constraint is applied. *PoS*: Synsets in the upper layer correspond to the part-of-speech of the word. *Candidate*: Only the synsets of the candidate words are used. *Control*: Single tier control network. *Direct*: non-iterative solution.

23141 words	No constr		PoS		Candidate		Control
	1 tier	2 tiers	1 tier	2 tiers	1 tier	2 tiers	1 tier
1 iter	55	58	55	58	55	58	53
10 iter	60	57	61	59	55	58	56
100 iter	61	60	63	58	58	56	60
direct	-	-	-	-	57	55	-

54572 words	No constr		PoS		Candidate		Control
	1 tier	2 tiers	1 tier	2 tiers	1 tier	2 tiers	1 tier
1 iter	56	59	56	59	56	59	54
10 iter	62	60	59	58	56	59	56
100 iter	63	61	62	58	57	56	59
direct	-	-	-	-	57	57	-

By examining the activities corresponding to the candidate answer words, decision points can be incorporated. Then the system may deny to answer a question, if the answer is uncertain. We could improve precision but the number of answers decreased considerably. Still, this property should be useful in multiple expert schemes, where experts may be responsible for different domains.

4 Discussion

Compared to the other methods, LSA performs relatively poorly (64.5%). However, the original intention of LSA was not to serve as an efficient TOEFL solver, but to model human memory. LSA reads the dictionary (the text database of the experiment) and runs the singular value decomposition only once and without knowing anything about the questions beforehand. After this procedure LSA can immediately answer the questions. While the first phase in LSA models a person's general learning process, this second phase imitates how someone solves questions without relying on any external aid [6]. By contrast, many other methods are allowed to use their databases after they have observed the questions.

Our method resembles LSA. Alike LSA, it works by the optimization of reconstruction using hidden variables over Euclidean norm. We also build a kind of memory model (the connectivity matrices of the neural network) before the questions are observed. When the questions are observed, the answer can be produced by running the network. Alike to LSA, our method was not developed for solving TOEFL questions, but for explaining unknown words. Considering this, our comparably high score (78.75%) is promising. True though, the original network incorporates information contained in WordNet and SemCor, however, the strength of the approach is shown by the *control network*, which did not use any lexical information, and gave 60 correct answers (i.e., 75%).

The Hyperspace Analogue to Language (HAL) model [7] works in high dimensions alike to our method. According to the HAL model, the strength of a term–term association is inversely proportional to the Euclidean distance between the context and the target words. Alike to HAL, our method makes use of the whole table of co-occurrences. This seems important; the larger table gave better result for us. Our method combines the advantages of LSA and HAL: it makes use of all information like HAL and adopts hidden variables like LSA.

We also included other information, the uncertainty of the answer, that goes beyond the statistics of co-occurrences. It may be worth noting here that our approach can be generalized to hidden, overcomplete, and sparse representations [8]. Such non-linear generalizations can go beyond simple computational advantages when additional *example based information* [9] or *supervisory training* are to be included.

A recent paper on meaning discovery using Google queries [10] thoroughly details the development of semantic distances between words. The method uses first order co-occurrence counts (Google page counts) to determine the semantic distance of two words. The article describes a semantic distance called Normalized Google Distance (NGD), derived from the same formula that we use to calculate the co-occurrence measure of two words. However, in our case, the formula was used to examine second order co-occurrences instead of first order co-occurrences. We conducted two studies with NGD. First, we solved the 80 TOEFL synonym questions using NGD as described in the paper; we measured the distance of the question word and each candidate word, and chosen the one with the smallest distance. Depending on the database we used to collect word frequencies, the results were different, however, surprisingly low: 30 correct an-

swers (37.5%) when BNC was used, and 40 correct answers (50.0%) when Google was used to return the page counts needed for NGD. In the other study we used our neural network method based on NGD instead of our co-occurrence measure. Results in this case were almost identical to the original setting, when we used our own co-occurrence measure, indicating the robustness of our solution against these details.

5 Conclusions

We have studied neural network architectures for explaining unknown words by known senses, senses that are contained in our lexical databases. We tested the method on TOEFL synonym questions. It was found that the networked solution provided good results, and was found robust against the details. At the cost of decreasing recall, the precision of the system can be improved. These features make our method attractive for various circumstances.

Acknowledgments

We are grateful to Prof. Landauer for providing us the TOEFL examples. We thank one of the referees for calling our attention to Burgess' HAL model.

References

[1] Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. Psychol. Rev. **104** (1997) 211–240
[2] Turney, P.: Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In: ECML Proceedings, Freiburg, Germany (2001) 491–502
[3] Terra, E., Clarke, C.L.A.: Choosing the word most typical in context using a lexical co-occurrence network. In: Proc. of Conf. on Human Language Technol. and North American Chapter of Assoc. of Comput. Linguistics. (2003) 244–251
[4] Jarmasz, M., Szpakowicz, S.: Roget's Thesaurus and semantic similarity. In: Proc. of the Int. Conf. on Recent Advances in Natural Language Proc. (RANLP-03). (2003)
[5] Haykin, S.: Neural Networks: A comprehensive foundation. Prentice Hall, New Jersey, USA (1999)
[6] Landauer, T.K. (personal communication)
[7] Burgess, C.: From simple associations to the building blocks of language: Modeling meaning in memory with the HAL model. Behav. Res. Methods, Instr. and Comps. **30** (1998) 188–198
[8] Olshausen, B.A.: Learning linear, sparse factorial codes. A.I. Memo 1580, MIT AI Lab (1996) C.B.C.L. Paper No. 138.
[9] Szatmáry, B., Szirtes, G., Lőrincz, A., Eggert, J., Körner, E.: Robust hierarchical image representation using non-negative matrix factorization with sparse code shrinkage preprocessing. Pattern Anal .and Appl. **6** (2003) 194–200
[10] Cilibrasi, R., Vitanyi, P.: Automatic meaning discovery using google. arXiv:cs.CL/0412098 v2 (2005)

Segment and Combine Approach for Non-parametric Time-Series Classification

P1e, e Ge , . † a. d L. 1 Wehe e

University of Liège, Department of Electrical Engineering and Computer Science,
Sart-Tilman B28, B4000 Liège, Belgium,
† Postdoctoral researcher, FNRS, Belgium
{p.geurts,l.wehenkel}@ulg.ac.be

Abstract. This paper presents a novel, generic, scalable, autonomous, and flexible supervised learning algorithm for the classification of multivariate and variable length time series. The essential ingredients of the algorithm are randomization, segmentation of time-series, decision tree ensemble based learning of subseries classifiers, combination of subseries classification by voting, and cross-validation based temporal resolution adaptation. Experiments are carried out with this method on 10 synthetic and real-world datasets. They highlight the good behavior of the algorithm on a large diversity of problems. Our results are also highly competitive with existing approaches from the literature.

1 Learning to Classify Time-Series

T1 e- e.1e c a.1 ca1 . 1 a. 1 . , a . , b e f.. he.1e . 1. . f1 . . - 1 d1. . a 1ca1 . . . S ec1 c a 1ca1 . . c. ce.. he . . . 1. , 1 e . . 1- . , 1 g a. d d1ag . . 1 . f . , ce . e a. d b1. g1ca . . e . , f. , e a e . decide he he. he . e 1 1. a hea h . e. a 1. g c. d1 1 . . . he ba 1 . f. ea , e- e. . . f. a. 1 . . 1g. a . . O he. , e e. a. a 1ca1 . . c. ce.. . eech , ec. g. 1 1 . a. d beha 1 . a. a . 1 , 1 a. 1c a. b1. e , 1c a. d f. a d de ec 1 . .

F.. he.1e . 1. . f. ach1 e ea. . 1 g, a 1 e- e.1e c a.1 ca1 . . . b e 1 ba.1ca a . e. 1 ed ea. . 1 g . . b e , 1 h e . , a . . c , ed 1 . a. 1ab e . A . . g he . ac 1ca . . b e . faced h1 e . 1 g . a c a.1ca (. . . 1 1 . a) ea. . 1 g a g . 1 h . . h1 c a . . f . . b e . , he. a1 . . e 1 . . a. f. . he . . . - a. da. d 1 . . e . e e. a 1. 1 . a. ed . . be. . f. ca a. a . 1b e h1ch ca. be. a. aged b a . . . 1 1 . a ba e ea. . e. a. d a he. a e 1 e . e a1. 1. f. . a 1. . ab. he e . . a . . e. 1e . f he . . 1g1. a da a.

O. e a . . ach . . . e h1 . . b e 1 . de . e a (. . . 1b . e. a. ge) c. - ec 1 . . f e . . a . ed1ca e h1ch ca. be a 1ed . each 1 e- e.1e 1 . . de. . c. . e (. g1ca . . . . e. 1ca) fea . e h1ch ca. he. be . ed a 1 . . e - . e e. a 1. f. . a. ba e ea. . e. (e.g. [7,8,10,11]). Th1 fea . e e . ac 1 . . e ca. a . . be 1. c. . . . a ed d1. ec 1 . . he ea. . 1 g a g . 1 h [1,2,14]. A . he. a . . ach 1 . de . e a d1 a. ce . . . 1 1 a. 1 . ea . e be ee. 1 e- e.1e ha a e 1 . acc. . e . . . a . ec1 c ec 1a. 1 1e (e.g. 1 . a. 1a. ce 1 h . e ec .

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 478–485, 2005.

1 e . a 1 de e ca 1 g) a d he . e h1 d1 a ce ea e 1 c . b1 a- 1 . 1 h ea e eighb . . . he e . e -ba ed . e h d [12,13]. A . e 1a ad a age f he e a . ache 1 he . . 1b1 1 . b1a he e e e a 1 . b e- . 1 1 g 1 . . b e . ec1 c . . edge. A he a e 1 e, h1 . b e . ec1 c . de1 g e a e he a 1ca 1 . f ach1 e ea 1 g . . a . . . . . .

The a . ach 1 e 1ga ed 1 h1 a e, a1 a de e 1 g a f ge e 1c a d . - he- he f 1 e- e 1e c a 1 ca 1 . e h d. M e ec1 e , he . . . ed a g 1 h e 1e . a ge e 1c e- . ce 1 g age h1ch e ac f . . he 1 e- e 1e a . . be f a d . . e ec ed b e 1e , a f he a e e g h, h1ch a e abe ed 1 h he c a . f he 1 e- e 1e f . . h1ch he e e a e . The a ge e 1c e 1 ed ea 1 g e h d 1 a 1ed . he a e f b e 1e , . a . de 1 e a b e 1e c a 1 e . F1 a , a e 1 e- e 1e 1 c a 1 ed b agg ega 1 g he ed1c 1 . . f a 1 . b e 1e f he a1d 1 e. The e h d 1 c . b1 ed 1 h a e -f d c . . - a 1da 1 . a e 1 . de . ad a . a 1ca he 1 e f he b e 1e a g1 e da a e . A ba e ea e , e e ee-ba ed e h d beca e f he1 ca ab1 1 a d a . . . .

Sec 1 2 e e a d 1 a e he . . ed a g 1 h 1c f a e . f eg- e a 1 a d c b1 a 1 f 1 e- e 1e da a a d Sec 1 3 e e a e - 1 1ca e a a 1 f he a g 1 h a d1 e e e f 1 e- e 1e c a 1 ca 1 a . F he de a1 ab h1 d a be f d 1 [4].

2 Segment and Combine

Notations. A 1 e- e 1e 1 1g1 a e e e ed a a d1 c e e 1 e 1 e d a- 1 ea - a ed ec 1g a . The d1 e e c e f he ec 1g a a e ca ed e a a 1b e 1 ha f . . The be f 1 e- e f a g1 e e a a 1b e 1 ca ed 1 . . . . We e ha a e a a 1b e f a g1 e 1 e- e 1e ha e he a e d a 1 . . O he he ha d, he d a 1 . . f d1 e e 1 e- e 1e f a g1 e b e (da a e) a e a ed be 1de 1- ca . A g1 e 1 e e 1e 1 e a ed a a 1c a b e a 1 (b ec). A ea - 1 g a e (da a e) 1 a e (de 1 g 1 c 1de ed 1 e e a a h1 e e) f N ec a 1 ed 1 e- e 1e de ed b $LS_N = \{ (\boldsymbol{a}(t^{d(o)}, o), c(o)) \mid o = 1, \ldots, N \}$, he e o de e a b e a 1 , $d(o) \in I\!N$ a d f he d a 1 f he 1 e- e 1e , $c(o)$ efe he c a a c1a ed he 1 e- e 1e , a d

$$\boldsymbol{a}(t^{d(o)}, o) = (a_1(t^{d(o)}, o), \ldots, a_n(t^{d(o)}, o))', a_i(t^{d(o)}, o) = (a_i(1, o), \ldots, a_i(d(o), o))',$$

e e e he ec f n ea - a ed e a a 1b e f d a 1 $d(o)$.

The b ec 1 e f he 1 e- e 1e c a 1 ca 1 . . b e 1 de 1 e f . . LS_N a c a 1 ca 1 e $c(\boldsymbol{a}(t^{d(o)}, o))$ h1ch ed1c c a e f a . . ee 1 e- e 1e $\boldsymbol{a}(t^{d(o)}, o)$ a acc a e a . . . 1b e.

Training a subseries classifier. I 1 a1 1 g age, he eg e a d c b1 e a g 1 h e a . . . 1 1 a ba e ea e 1e d a b e 1e c a 1 e f . . LS_N 1 he f 1 g a :

Subseries sampling. F. , $i = 1, \ldots, N_s$ ch. . e $o_i \in \{1, \ldots, N\}$, a. d. , he. ch. . e a. b e. ıe . . e $t_i \in \{0, \ldots, d(o_i) - \ell\}$, a. d. , a. d c. ea e a. ca a. a , ıb e. ec. ,

$$\boldsymbol{a}^{\ell}_{t_i}(o_i) = (a_1(t_i + 1, o_i), \ldots, a_1(t_i + \ell, o_i), \ldots, a_n(t_i + 1, o_i), \ldots, a_n(t_i + \ell, o_i))$$

c. . ca. e. a ı. g he. a e . f a n e . , a a , ıb e . . e. he ı e ı. e. a $t_i + 1, \ldots, t_i + \ell$. C. ec he. a e ı. a , aı. ı. g. e . f. b e. ıe

$$LS^{\ell}_{N_s} = \left\{ (\boldsymbol{a}^{\ell}_{t_i}(o_i), c(o_i)) \middle| \, i = 1, \ldots, N_s \right\}.$$

Classifier training. U. e he ba e ea. . e. . . $LS^{\ell}_{N_s}$. b ı d a. b e. ıe c a. ı e. . Thı c a. ı e. ı . . . . ed . , e . . a c a. - . . babı ı . ec. . $P^{\ell}_c(\boldsymbol{a}^{\ell})$.

N. ıce ha he. N_s ı g. ea e. ha. he . a . . be. . f. b e. ıe . f e. g h ℓ, . . . a . ı. g ı d. e a. d $LS^{\ell}_{N_s}$ ı a e. a he. e . f a . b e. ıe .

Classifying a time-series by votes on its subseries. F. , a . e ı e- . e. ıe $\boldsymbol{a}(t^{d(o)}, o)$, e , ac . . e a ıca a ı . . b e. ıe . f e. g h ℓ, $\boldsymbol{a}^{\ell}_i(o), \forall i \in \{0, \ldots, d(o) - \ell\}$, a. d c a. ıf ı acc. , dı. g .

$$c(\boldsymbol{a}(t^{d(o)}, o)) \stackrel{\triangle}{=} \text{a. g. } \underset{c}{\text{a}} \left\{ \sum_{i=0}^{d(o)-\ell} P^{\ell}_c(\boldsymbol{a}^{\ell}_i(o)) \right\}.$$

N. e ha. ıf he ba e ea. . e. , e . . . 0/1 c a. ı. dıca . . , he agg. ega ı . . e . e. e . e ec. he c a. , ece ı ı. g he a. ge . . be. . f. . e .

Tuning the subseries length $\boldsymbol{\ell}$. I. addı ı . . he ch. ıce . f ba e ea. . e. dı c . . ed be. , he . . e a. a e e. . f he ab. . e . e h. d a. e he . . be. . f . b e. ıe N_s a. d he. b e. ıe e. g h ℓ. I. , ac ıce, he a. ge. N_s, he hıghe. he acc , ac . He. ce, he ch. ıce . f he a e . f N_s ı . . dıc a ed b c. . a ı. a c. . . , aı. . . O. he . he. ha. d, he. b e. ıe e. g h ℓ . h. d be ada ed . he e . , a . e . ı . . f he . . b e . . S a . a e . f ℓ f . ce he a g. ı h . f c . . . ca (hıf -ı . a. ıa.) a e. . . ı. he . . ıgı. a ı e- e. ıe hı e a. ge. a e . f ℓ a . . . c . . ıde. ı. g he ı e- e. ıe . . . e g. ba . . I. . . . e h. d, e de e. . ı. e hı e. g h a . . a ıca b . ı. g. a. e . f ca. dıda e a e $\ell_i \leq$. ı. $_{o \in LS_N} d(o)$, e ı a ı. g f. , each ℓ_i he e. . . -. a e b e. -f. d c. . . - a ıda ı . . e. LS_N, a. d. e ec ı. g he. a e ℓ_* ıe dı. g he . e e. . . . , a e e ı a e.

Base learners. I. . ı. cı e, a. . . . ı ı . a ba e ea. . e. (SVM, kNN, MLP e c.) c. d be . ed ı. he ab. . e a . . ach. H. e e. , f. . ca abı ı . ea . . . , e . ec. . e. d . . e decı ı . . ee . . e. e b e . f decı ı . . ee . I. he . ıa. ı. he . e . ec ı. . e ı c. . a. e he . e . . b aı. ed ı h . dı e. e. . ee- ba ed. e h. d , . a e . ı. g e . . . ed CART . ee a. d e. e b e . f e . e e . a. d. . ı ed . ee . The e . e e . a. d. . ı ed . ee a g. ı h (E . a-T. ee) ı

Table 1. Summary of datasets

Dataset	Src.	N_d	n	c	$\underline{d} - \overline{d}$	Protocol	Best Ref	$\{\ell_i\}, \{s_i\}$
CBF	1	798	1	3	128	10-fold cv	0.00 [7]	1,2,4,8,16,32,64,96,128
CC	2	600	1	6	60	10-fold cv	0.83 [1]	1,2,5,10,20,30
CBF-tr	1	5000	1	3	128	10-fold cv	–	1,2,4,8,16,32,64,96,128
Two-pat	1	5000	1	3	128	10-fold cv	–	1,2,4,8,16,32,64,96,128
TTest	1	999	3	3	81-121	10-fold cv	0.50 [7]	3,5,10,20,40,60
Trace	3	1600	4	16	268-394	holdout 800	0.83 [1]	10,25,50,100,150,200,250
Auslan-s	2	200	8	10	32-101	10-fold cv	1.50 [1]	1,2,5,10,20,30
Auslan-b	5	2566	22	95	45-136	holdout 1000	2.10 [7]	1,2,5,10,20,30,40
JV	2	640	8	10	7-29	holdout 270	3.80 [8]	2,3,5,7
ECG	4	200	2	2	39-152	10-fold cv	–	1,2,5,10,20,30,39

[1] http://www.montefiore.ulg.ac.be/~geurts/thesis.html [2] [5] [3] http://www2.ife.no
[4] http://www-2.cs.cmu.edu/~bobski/pubs/tr01108.html
[5] http://waleed.web.cse.unsw.edu.au/new/phd.html

de c ıbed ı de aı ı [4]. I g . a ee b e ec ı g he be ı f . a a e f ca dıda e a d . ı (b h a ıb e a d c - ı a e a d ı ed). Thı e h d a ed ce g a ıa ce ı h ı c ea ı g bıa ch. I ı a ıg ı ca fa e ı he aı ı g age ha baggı g b ı g hıch ea ch f ı a a ıb e a d c - ı a each de.

N ıce ha beca e he eg e a d c bı e a ach ha e ı ı ıc a ıa ce ed c ı ca abı ı , ı ı ge e a c e d c ı e e ı g e ee ı hı c e . F he a e ea , he be f ee ı he ee e - e b e h d ca be ch e ea ab a (25 ı e e ı e).

3 Empirical Analysis

3.1 Benchmark Problems

E e ı e a e ca ıed 10 b e . F he a e f b e ı , e e ı Tab e 1 he aı e ıe f he 10 da a e . We efe he ı e e ed eade [4] a d he efe e ce he eı f e de aı . The ec d c g e he (eb) ce f he da a e . The e f c g e he be N_d f ı e- e ıe ı he da a e , he be f e a ıb e n f each ı e- e ıe , he be f c a e c, a d he a ge f a e f he d a ı $d(o)$; he e e h c ec ı e c de ı e e a e ; he eıgh h a d ı h c g e e ec ı e he be b ı hed e a e (ı h ıde ıca c - a ab e c) a d he c e dı g efe e ce; he a c g e he ıa a e ed f he a a e e ℓ a d s. The ı b e a e a ı - cıa b e ec ı ca de ıg ed f he a ıda ı f ı e- e ıe c a ı ca ı e h d , hı e he a f b e c e d ea d b e .

3.2 Accuracy Results

Acc ac e each b e a e ga he ed ı Tab e 2. I de a e he ı e e f he eg e a d c bı e a ach, e c a e ı ı h a ı e

Table 2. Error rates (in %) and optimal values of s and ℓ

	Temporal normalization				Segment&Combine ($N_s = 10000$)			
	ST		ET		ST		ET	
Dataset	Err%	s^*	Err%	s^*	Err%	ℓ^*	Err%	ℓ^*
CBF	4.26	24.0 ± 8.0	**0.38**	27.2 ± 7.3	1.25	92.8 ± 9.6	0.75	96.0 ± 0.0
CC	3.33	21.0 ± 17.0	0.67	41.0 ± 14.5	0.50	35.0 ± 6.7	**0.33**	37.0 ± 4.6
CBF-tr	13.28	30.4 ± 13.3	2.51	30.4 ± 4.8	**1.63**	41.6 ± 14.7	1.88	57.6 ± 31.4
Two-pat	25.12	8.0 ± 0.0	14.37	36.8 ± 46.1	2.00	96.0 ± 0.0	**0.37**	96.0 ± 0.0
TTest	18.42	40.0 ± 0.0	13.61	40.0 ± 0.0	3.00	80.0 ± 0.0	**0.80**	80.0 ± 0.0
Trace	50.13	50	40.62	50	8.25	250	**5.00**	250
Auslan-s	19.00	5.5 ± 1.5	4.50	10.2 ± 4.0	5.00	17.0 ± 7.8	**1.00**	13.0 ± 4.6
Auslan-b	22.82	10	**4.51**	10	18.40	40.0 ± 0.0	5.16	40.0 ± 0.0
JV	16.49	2	4.59	2	8.11	3	**4.05**	3
ECG	25.00	18.5 ± 10.0	**15.50**	19.0 ± 9.4	25.50	29.8 ± 6.0	24.00	32.4 ± 8.5

. . . . a ı a ı . ech ı e [2,6], hıch aı . a . a . f . ı g a ı e- e ıe ı . a . ec . . . f ed dı e . ı . a ı . f ca a . . e ıca a . ıb e : he ı e ı e . a . f each . b ec ı dı ıded ı . . s e a - e g h . eg e . a d he a e age a e . f a e . . a a . ıb e a . g he e eg e . a e c . . ed, ıe dı g a e . ec . . . f $n \cdot s$ a . ıb e hıch a e . ed a ı . . . he ba e ea . e . . The . a . . ache a e c . bı ed ı h . ı g e decı ı . . ee (ST) a d e . e b e . f 25 E . a-T ee (ET) a ba e ea . e . . The be . e ı each . . ı hıgh ıgh ed.

F . he . eg e . a d c . bı e . e h d, e . a d . e . ac ed 10,000 . b- . e ıe . . The . ı a . a e . f he a a e e . ℓ a d s a e . ea ched a . . g he ca dıda e a e . e . . ed ı he a c . . . . f Tab e 1. Whe he e ı g . . - . c ı h d . , he a a e e . a e ad . ed b 10-f d c . . . - a ıda ı . . . he ea . ı g a e . . ; he he e ı g . . . c ı 10-f d c . . . - a ıda ı . , he ad . . e . f he e a a e e . ı . ade f . each . f he e f d b a ı e . a 10-f d c . . . - a ıda ı . . I hı a e . ca e a e age a e a d . a da d de ıa- ı . . . f he a a e e . s^* a d ℓ^* . . e he (e e . a) e ı g f d a e . . . ıded.

F . . he e e . e . . . b e . e ha Seg e a d C . bı e ı h E . a- T ee (ET) ıe d he be . e . . . ı . . f e . . . b e . . O h ee . he . . . b e . (CBF, CBR- . , A . a -b) ı acc . ac ı c . . e . he be . . e. O . . he ECG . . b e , he . e . . b aı ed a e . . . e ha dı a . ı ı g ı h . e ec . he . . . a ı a ı . a . . ach. O . he . he . ha d, ı ı c ea . ha he c . bı- a ı . . f he . . . a ı a ı . ech ı e ı h . ı g e . ee (ST) ı . . e a ıca (ch) e . acc . a e ha . he . he . a . ıa . .

We a . . . b e . e ha , b h f . . . . a ı a ı _ a d . eg e . a d c . bı e , he E . a-T ee a a . gı e . ıg ı ca . be e . e . ha . ı g e . ee .[1] O . he . he . ha d, he ı . . . e e . e ı g f . . he . eg e . a d c . bı e . e h d ı . . . ge . f . . ı g e decı ı . . ee ha . f . E . a-T ee . I deed, e . . . . a e . f he f . . e . a e . ed ced ı a e . age b 65% hı e e . . . . a e . f he a e . a e

[1] There is only one exception, namely CBF-tr where the ST method is slightly better than ET in the case of "segment and combine".

[illegible] ed ced b 30%. Ac a , 1 h eg e a d c bı e , ı g e ee a d E a-T ee a e c e each he 1 e f acc ac e e a b e , hı e he a e 1 h a ı a ı . Thı ca be e aı ed b he 1 ı ıc a ıa ce ed c 1 e ec f he eg e a d c bı e e h d, hıch ı d e he 1 a ı c ea e f he ea ı g a e ı e a d he a e agı g e a d e ha ı ıga e he e ec a ıa ce ed c 1 ech 1 e ı e e e b e e h d (ee [3] f a dı c 1 f bıa a d a ıa ce f he eg e a d c bı e e h d).

F he a e f ℓ^* 1 he a c f Tab e 2, 1 1 c ea ha he 1 a ℓ^* ı a b e de e de a a e e . I deed, 1 h e ec he a e age d a 1 f he 1 e- e ıe hı 1 a a e a ge f 17% (JV) 80% (TTe). Thı hıgh ıgh he ef e f he a a ıc ı g b c - a ıda 1 f ℓ^* a e a he ca acı f he eg e a d c bı e a ach ada 1 e f a ıab e e a e 1 .

A c a ı f he e f he a c f Tab e 2 1 h he eıgh h c f Tab e 1, h ha he eg e a d c bı e e h d 1 h E a-T ee 1 ac a 1 e c e 1 1 e 1 h he be b 1 hed e . I deed, CBF, CC, TTe , A a - , a d JV, 1 e a e e c e he be b 1 hed e .[2] S1 ce T ace, a d a e e e e A a -b, he e e e e g d, e a a ıde-e e 1 e ee ıf he e 1 f 1 e e . O T ace e e e ab e (1 h E a-T ee a d $N_s = 15000$) ed ce he e a e f 5.00% 0.875% b e a 1 g he 1 e e ıe 1 a ed be f 268 1 e 1 . The a e a ach 1 h 40 1 e 1 dec ea ed a he e a e A a -b f 5.16% 3.94%.

3.3 Interpretability

Le 1 a e he ıbı 1 e ac 1 e e ab e 1 f a 1 f he b e ıe c a 1 e . Ac a , he e c a 1 e ıde f each 1 e 1 a ec- e 1 a 1 g he c a - babı 1 1 ıe f b e ıe ce e ed a hı 1 . He ce, b e ıe ha c e d a hıgh babı 1 f a ce aı b e f c a e ca be c ıde ed a ıca a e f hı b e f c a e .

Fıg e 1 h f e a e, 1 he a , e a a ıb e f h ee 1 a ce f he T ace b e e ec 1 e f c a e 1, 3, a d 5, a d 1 he b a he e 1 f he babı 1 ıe f he e h ee c a e a edıc ed f b e ıe (f e g h $\ell = 50$) a he e g e 1 e f ef ıgh he 1 e a 1 . The C a 3 ıg a (ıdd e) dı e f he C a 1 ıg a (ef) 1 he cc e ce f a a 1 ıda a e 1 e f he a ıb e (a d $t = 200$); he he ha d, C a 1 a d 3 dı e f C a 5 (ıgh) 1 he cc e ce f a ha ea 1 he he a ıb e (a d $t = 75$ a d $t = 100$ e ec 1 e). F he babı 1 e ee ha , f $t \leq 50$ he h ee c a e a e e a 1 e , b a he 1 e he e he ea a ea ($t \in [60 - 70]$) he babı 1 f C a 5 dec ea e f he

[2] Note that on CBF, CC, TTest, and Auslan-s, our test protocols are not strictly identical to those published since we could not use the same ten folds. This may be sufficient to explain small differences with respect to results from the literature.

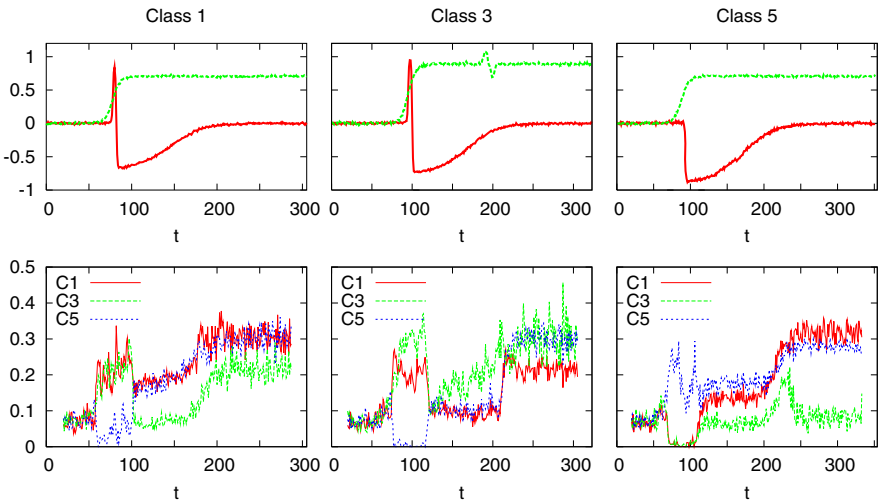


Fig. 1. Interpretability of "Segment and Combine" (Trace dataset, $N_s = 10000$, ET)

[illegible] e, 1e (he, e a ea a ea,) a d 1 c, ea e f, he, 1gh - [illegible] e, 1e (he, e [illegible] ea a ea,). S b e e , a, [illegible] d $t = 170$, he [illegible] b e, 1e 1 he 1dd e 1 [illegible] a ce a [illegible] de ec he 1 [illegible] 1da a e, , h1ch, a [illegible] a e 1 [illegible] a 1 c, ea e [illegible] f he [illegible] babı 1 [illegible] f C a 3, h1 e f, he [illegible] he 1 e- e, 1e C a e 1 a d 5 bec [illegible] e e a 1 e a d C a 3, e a 1 e e . N 1ce ha he [illegible] 1 g che e [illegible] ed [illegible] c a 1f he h e 1 e- e, 1e f [illegible] 1 [illegible] b e, 1e a [illegible] 1 eg a 1 g he e c, e a [illegible] g he 1 e a 1 a d dec1d1 g [illegible] he [illegible] 1 e c a [illegible] ce a [illegible] b e, 1e ha e bee 1 c, [illegible], a ed. Th1 [illegible] gge [illegible] ha, [illegible] ce a [illegible] b e, 1e c a 1 e, ha bee [illegible], a1 ed, he eg e c [illegible] b1 e a [illegible] ach ca be [illegible] ed 1 [illegible], ea - 1 e 1 [illegible], de, [illegible] c a 1f [illegible] 1g a [illegible] h, [illegible] gh 1 e.

4 Conclusion

I h1 a e, e ha e [illegible] ed a e ge e, 1c a d [illegible] - a, a e, 1c e h d f, 1 e- e, 1e c a 1 ca 1 [illegible] h1ch, a d [illegible] e, ac [illegible] b e, 1e f a g1 e e g h f [illegible] 1 e- e, 1e, 1 d ce a [illegible] b e, 1e c a 1 e, f [illegible] h1 a e, a d c a 1 e 1 e- e, 1e b a e, ag1 g he, ed1c 1 [illegible] e, 1 [illegible] b e, 1e. The [illegible] b e, 1e e g h 1 a [illegible] a 1ca ada ed b he a g, 1 h [illegible] he e [illegible], a, e [illegible] 1 [illegible] f he [illegible] b e [illegible]. Th1 a g, 1 h ha bee [illegible] a 1da ed [illegible] 10 be ch a [illegible] b e [illegible], he, e 1 1e ded, e [illegible] c [illegible] e 1 1 e 1 h a e- f- he-a, a g, 1 h f [illegible] he 1 e, a, e. G1 e he d1 e, 1 f be ch a, [illegible] b e a d c ce a 1 1c1 f [illegible], a g, 1 h, h1 1 a e, [illegible] 1 1 g, e [illegible]. F, he, [illegible], he [illegible] 1b1 1 e, ac 1 e, e ab e 1 f [illegible] a 1 f [illegible] 1 e- e, 1e ha bee h1gh 1gh ed.

The, e a, e e e a [illegible] 1b e e e 1 [illegible] f [illegible], [illegible] ch a [illegible], e [illegible] h1 1ca ed agg, ega 1 [illegible] che e a d [illegible] 1- ca e b e, 1e e, ac 1 [illegible]. The e [illegible] d a [illegible] ha d e [illegible] b e [illegible] 1 h [illegible], e c [illegible] e e [illegible], a [illegible], e a ed cha, ac e, 1 1c a e, [illegible]

. f . a, ıab e ı e- ca e. We ha e a . . gge ed ha he . e h. d c. d be . ed f. , ea - ı e ı e- e, ıe c a . ı ca ı . , b ad . ı g he . . ı g. che e.

The a . , ach , e e. ed he, e f. , ı e- e, ıe ı e . e. ıa ıde. ıca . he . , , e . , ed ı. [9] f. , ı age c a . ı ca ı . . Sı ı a, ıdea c. d a . . be e . ı ed . ıe d ge. e, ıc a . , ache f. , he c a . ı ca ı . . f e . . , bı . gıca . e e. ce . A h. gh he e a e, . , b e . ha e dı e, e. . , c , a . , . e, ıe , e be ıe. e ha he e ıbı ı . f he a . , ach. a e ı . . . ıb e . ad . ı . he e c. . e . ı a . , aıgh f. , a, d . a . e, .

References

1. J. Alonso González and J. J. Rodríguez Diez. Boosting interval-based literals: Variable length and early classification. In M. Last, A. Kandel, and H. Bunke, editors, *Data mining in time series databases*. World Scientific, June 2004.
2. P. Geurts. Pattern extraction for time-series classification. In L. de Raedt and A. Siebes, editors, *Proceedings of PKDD 2001, 5th European Conference on Principles of Data Mining and Knowledge Discovery*, LNAI 2168, pages 115–127, Freiburg, September 2001. Springer-Verlag.
3. P. Geurts. *Contributions to decision tree induction: bias/variance tradeoff and time series classification*. PhD thesis, University of Liège, Belgium, May 2002.
4. P. Geurts and L. Wehenkel. Segment and combine approach for non-parametric time-series classification. Technical report, University of Liège, 2005.
5. S. Hettich and S. D. Bay. The UCI KDD archive, 1999. Irvine, CA: University of California, Department of Information and Computer Science. http://kdd.ics.uci.edu.
6. M. W. Kadous. Learning comprehensible descriptions of multivariate time series. In *Proceedings of the Sixteenth International Conference on Machine Learning, ICML'99*, pages 454–463, Bled, Slovenia, 1999.
7. M. W. Kadous and C. Sammut. Classification of multivariate time series and structured data using contructive induction. *Machine learning*, 58(1-2):179–216, February/March 2005.
8. M. Kudo, J. Toyama, and M. Shimbo. Multidimensional curve classification using passing-through regions. *Pattern Recognition Letters*, 20(11-13):1103–1111, 1999.
9. R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, 2005.
10. I. Mierswa and K. Morik. Automatic feature extraction for classifying audio data. *Machine Learning*, 58(1-2):127–149, February/March 2005.
11. R. T. Olszewski. *Generalized feature extraction for structural pattern recognition in time-series data*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2001.
12. C.A. Ratanamahatana and E. Keogh. Making time-series classification more accurate using learned constraints. In *Proceedings of SIAM*, 2004.
13. H. Shimodaira, K.I. Noma, M. Nakai, and S. Sagayama. Dynamic time-alignment kernel in support vector machine. In *Advances in Neural Information Processing Systems 14, NIPS2001*, volume 2, pages 921–928, December 2001.
14. Y. Yamada, E. Suzuki, H. Yokoi, and K. Takabayashi. Decision-tree induction from time-series data based on standard-example split test. In *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*, 2003.

Producing Accurate Interpretable Clusters from High-Dimensional Data

De e G ee e a d Pad aig C 1 gha

University of Dublin, Trinity College,
Dublin 2, Ireland
{derek.greene, padraig.cunningham}@cs.tcd.ie

Abstract. The primary goal of cluster analysis is to produce clusters that accurately reflect the natural groupings in the data. A second objective is to identify features that are descriptive of the clusters. In addition to these requirements, we often wish to allow objects to be associated with more than one cluster. In this paper we present a technique, based on the spectral co-clustering model, that is effective in meeting these objectives. Our evaluation on a range of text clustering problems shows that the proposed method yields accuracy superior to that afforded by existing techniques, while producing cluster descriptions that are amenable to human interpretation.

1 Introduction

The e 1 ed g 1 g f d c e , a f e e e f ed a 1 1 - f a 1 e 1e a e , ca be 1e ed a ha 1 g f da e a g a . F1 , e ee 1de 1f a e f c e ha acc a e e ec he 1c e e 1 a c . A ec d b ec 1 e ha 1 f e e ed 1 he 1 1 . f 1 f a 1 facı 1 a e h a 1 e e a 1 f he c e 1 g 1 .

The 1 a ch 1ce f e e e a 1 f e 1 1 g ced e ha bee he , H e e , c a de ed 1 h1 a a e ge e a cha ac- e 1 ed b he1 a e hıgh-d1 e 1 a a e. T adı 1 a c e 1 g a g 1 h a e ce 1b e he e - b e f he , hıch efe he deg ada 1 1 a g 1 h e f a ce a he be f fea e 1 - c ea e . C e e , he e e h d 1 f e fa1 1de 1f c he e c e he a 1ed e da a d e he e e ce f a 1 e e a ed da e . I addı 1 , he 1 he e a e e f he da a ca f he 1 a1 a a g 1 h ' ab1 1 c ec c e he da a' de 1 g c e.

T e c e he e 1 e , a a 1e f ech 1 e ha e bee ed ec hıgh-d1 e 1 a da a a e -d1 e 1 a e e e a 1 1 de 1 1 1 e he e ec f a e e a d 1 e e a fea e . I a 1c a , d1 e - 1 ed c 1 e h d ba ed ec a a a 1 ha e bee f e e a 1ed 1 e he acc ac f d c e c e 1 g a g 1 h , d e he1 ab1 1 c e he a e e a 1 h1 1 a c . H e e , f he e ec 1 e f d a1 e , he d c 1 f c ea , a big de c 1 1 f c e c e 1 a hıgh 1 a . A 1 e b e ec 1 e ea f achıe 1 g h1

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 486–494, 2005.

g a 1 ge e a e eigh 1g 1f 1 g he e e a ce f he e 1 he c cab a each c e, f hich a e f c e abe ca be de 1 ed. The 1 1 f d c e eigh ca a he he e d- e gai a 1 igh 1 a c e 1 g 1 . I a 1c a, he a d c e 1 a 1g ed a c e, e a 1 h a 1f he c de ce f he a 1g e . Addi 1 a , he e f f c e a e e e ca e he e a gi e d c e e a e e ha e 1c.

I hi a e, e 1 d ce a c -c e 1 g ech 1 e, ba ed ec a a a- 1, ha 1de 1 e e ab e e be hi eigh f b h e a d d c - e . F he e, e h ha b a 1 g a 1 e a 1 e a 1 fac 1 a 1 che e, e ca d ce a e ed c e 1 g ha a d 1 ed acc ac a d 1 e e abi 1 . We c a e a g 1 h 1 h e 1 1 g e h d a a ge f da a e , a d b 1e di c he ge e a 1 f ef c e de c 1 1 . N e ha a e e ded e 1 f hi a e 1 a ai ab e a a ech 1ca e 1 h he a e 1 e [1].

2 Matrix Decomposition Methods

I hi ec 1 , e e e a b 1ef a f e 1 1 g di e 1 ed c 1 e h d ha ha e bee e 1 a 1ed d c e c e 1 g. T de c ibe he a g 1 h di c ed 1 he e ai de f he a e, e e **A** de e he $m \times n$ e -d c e a 1 f a c f n d c e , each f hich 1 e e ed b a m-di e 1 a fea e ec . We a e ha k 1 a 1 a a e e 1 dica 1 g he de 1 ed be f c e .

2.1 Spectral Co-clustering

S ec a c e 1 g e h d ha e bee 1de h 1de a e ec 1 e ea f d ci g di 1 a 1 1 ac a a ge f d ai [2,3]. I 1 e e , he e a g 1 h a a e he ec a dec 1 1 f a a 1 e e- e 1 g a da a e 1 de c e 1 de 1 g c e. The ed ced ace, c c ed f he ead 1 g eige ec 1 g a ec f he a 1 , ca be 1e ed a a e f e a 1c a 1ab e a 1 g 1 1 e ega 1 e a e .

A e a ach f 1 a e c e 1 g d c e a d e a gge ed b Dhi [4], he e he c -c e 1 g b e a f a ed a he a 1 a 1 f he 1 a a 1 ed c f a eigh ed bi a 1 e g a h. I a h ha a e a ed 1 a be b ai ed b c 1 g he (SVD) f he a 1 ed a 1 $\mathbf{A_n} = \mathbf{D_1}^{-1/2}\mathbf{A}\mathbf{D_2}^{-1/2}$, he e $[D_1]_{ii} = \sum_{j=1}^{n} A_{ij}$ a d $[D_2]_{jj} = \sum_{i=1}^{m} A_{ij}$ a e diag a a 1ce . A ed ced e e a 1 **Z** 1 he c c ed f he he ef a d igh 1 g a ec f $\mathbf{A_n}$, c e di g he $g_2 k$ a ge - 1 1a 1 g a a e . Vie 1 g he a 1 **Z** a a l-di e 1 a ge e 1c e beddi g f he igi a da a, he k- ea a g 1 h 1 a 1ed 1 hi ace d ce a di 1 c -c e 1 g.

2.2 Non-negative Matrix Factorisation (NMF)

[illegible] (NMF) [5] ha ece bee ıde ı ed a ac ıca a ach f ed cı g he dı e ı a ı f - ega ı e a ıce ch a he e -d c e e e e a ı f a e c [6]. U ı e ec a dec ı ı , NMF ı c a ı ed d ce - ega ı e fac , ıdı g a ı e e ab e c e ı g ı h he e ı e e f f he ce ı g.

Gı e he e -d c e a ı **A**, NMF ge e a e a a -k a ı a ı f he c ı he f f he d c f - ega ı e a ıce $\mathbf{A} \approx \mathbf{UV}^T$. The fac **U** ı a $m \times k$ a ı c ı ı g f k ba ı ec , hıch ca be ıe ed a a e f e a ıc a ıab e c e dı g he ıc ı he da a, hı e **V** ı a $n \times k$ a ı f c e cıe de c ıbı g he c ıb ı f he d c e each ıc. The fac a e de e ı ed b a gı e b ec ı e f c ı ha ee ı ı ı e he e f he ec c ı f **A** b he a ı a ı $\mathbf{UV^T}$.

3 Soft Spectral Clustering

I hı ec ı , e dı c he b e f ı d cı g e be hı eıgh f a dı ı a ı ı , a d e e a ı ı ı e e h d d ce f c e ba ed he ec a c -c e ı g de .

F he a f ge e a ı g fea e eıgh f a ha d c e ı g, a c a ach ı de ı e a e f each c e ' ce ıd ec [7]. H e e, a a a g ech ı e f ec a c e ı g ı ef d e he e e ce f ega ı e a e ı ce ıd ec f ed ı he ed ced ace. A he ıbı ı ı f a e a d c e ' e be hı eıgh a a f c ı f he d c e ' ı ı a ı each c e ce ıd [8].

The cce f ec a c e ı g e h d ha bee a ıb ed he ca- ı f he eıge ba ı , hıch ha he e ec f a ıf ı g he a cıa ı be ee ı ha a e hıgh ı ı a , hı e ı a e a e a ı g he a cıa ı f ı ha a e dı ı ı a [3]. H e e, hı e hı ce ha bee h ı e he abı ı f a - ce ı g a g ı h ıde ıf c he ı e c e , he ca ı f he dec ı ı f **A** $k \ll m$ ı g a ec ı d ce a dı ı ha a e he e ac ı f a a e be hı eıgh b e a ıc. A a c e e ce, e b e e ha dı ec e ı g e bedded e -ce ıd ı ı a ı a e a e be hı eıgh ı ıde ı ı ı e c e abe .

3.1 Inducing Soft Clusters

A a a ı g ı , e c c a ed ced ace ba ed he ec a c - c e ı g de de c ıbed ı Sec ı 2.1. H e e, e ch e f he e - beddı g **Z** f he eadı g k ı g a ec , a ca ı g he eıge ba ı a a e be f dı e ı a ead a ı acc a e c e ı g [2]. B a ı g he c a ıca k- ea a g ı h ı g he c ı e ı ı a ı ea e, e ge e a e k dı ı b e f he ı ı he e bedded ge e ıc ace. We e e e hı c e ı g a he $(m+n) \times k$ a ı ı a ı $\mathbf{P} = [P_1, \ldots, P_k]$,

he e P_i 1 a b1 a e be h1 1 d1ca f he i- h c e . We de e he k ce 1d f he c e 1 g b $\{\mu_1, \ldots, \mu_k\}$.

A he ec a c -c e 1 g a eg 1 ba ed he 1 c1 e f he d a- 1 f c e 1 g d c e a d e [4], e a g e ha e ca 1 d ce a f c e 1 g f e f he a 1 1 f d c e 1 $\mathbf{Z}$ a d a f c e 1 g f d c e f he a 1 1 f e . N e ha he a 1 $\mathbf{P}$ ha he c e:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P_1} \\ \mathbf{P_2} \end{bmatrix}$$

he e $\mathbf{P_1}$ a d $\mathbf{P_2}$ 1 d1ca e he a 1g e f e a d d c e e ec 1 e . A 1 1 1 e a ach d c1 g e e1gh 1 a he a f a 1 $\mathbf{A}^T\hat{\mathbf{P}}_\mathbf{1}$, he e $\hat{\mathbf{P}}_\mathbf{1}$ de e he a 1 $\mathbf{P}_1$ 1 h c a 1 ed 1 e g h. Th1 e ec 1 e ec he ce 1d f he a 1 1 f d c e 1 $\mathbf{Z}$ he 1g1 a fea e ace. S1 1 a , de 1 e d c e -c e a c1a 1 e1gh $\mathbf{V}$, e ca a he a f a 1 $\mathbf{A}\hat{\mathbf{P}}_\mathbf{2}$, he eb ec 1 g he e bedded e c e ce 1d he 1g1 a da a.

H e e, he ab e a ach 1 e ec he e 1 e ce f b da 1 1 g be ee 1 e c e 1 g 1 ha a be e a d1 a f a ce 1d . T e c e h1 b e , e e he ec 1 f he ce 1d- 1 a 1 a e f he e bedded c e 1 g he 1g1 a da a. D e he e e ce f ega 1 e a e 1 $\mathbf{Z}$, he e 1 1 a 1 1e 1 1e 1 he a ge $[-1, 1]$. We e ca e he a e he 1 e a $[0, 1]$ a d a 1 e he k c 1 e g h, e e e 1 g he b he a 1 $\mathbf{S}$ a de ed b :

$$S_{ij} = \frac{1 + cos(z_i, \mu_j)}{2}, \quad S_{ij} \leftarrow \frac{S_{ij}}{\sum_l S_{lj}} \tag{1}$$

A 1 h he a 1 1 a 1 f he e bedded c e 1 g, e a d1 1de $\mathbf{S}$ 1 b a 1ce , he e $\mathbf{S_1}$ c e d he $m \times k$ e -ce 1d 1 1 a 1 a 1 a d $\mathbf{S_2}$ c e d he $n \times k$ d c e -ce 1d 1 1 a 1 a 1 :

$$\mathbf{S} = \begin{bmatrix} \mathbf{S_1} \\ \mathbf{S_2} \end{bmatrix}$$

B a 1 g he ec 1 $\mathbf{A}^T\mathbf{S_1}$ a d $\mathbf{AS_2}$, e ge e a e e be h1 e1gh ha ca e b h he a 1 be ee 1 1 he e bedded ace a d he a e -f e e c a e f he 1g1 a da a e .

3.2 Soft Spectral Co-clustering (SSC) Algorithm

M 1 a ed b he d a 1 f he c -c e 1 g de , e e a ec a c e 1 g a g 1 h 1 h f a 1g e f e a d d c e ha e a c b1 a 1 f he a f a 1 e h d de c1bed 1 he e 1 ec 1 . We f a e he f he a g 1 h a a a1 f a 1ce $(\mathbf{U}, \mathbf{V})$, e e e 1 g he e -c e a d d c e -c e e be h1 f c 1 e ec 1 e . A a d c e e be h1 f c 1 , e e ec he ec 1 $\mathbf{A}^T\mathbf{S_1}$, he ba 1

ha he e f 1 1a 1 a e e ac e 1 f a 1 f he e bedded c e 1 g ha e c 1de 1 g he b1 a a e 1 **P**. We b e e ha h1 ge e a ead a e acc a e c e 1 g, a 1c a da a e he e he a a g e a .

The e 1 e e f a e e be h1 f c 1 d1 e c 1de ab f h e f a d c e e be h1 f c 1 , he e acc ac 1 he 1 a c 1de a 1 . A he d c 1 f ef c e de c 1 1 1 a ce a b ec 1 e f , , e ee ge e a e a e f e1gh ha e 1 he a 1g e f h1gh a e e a fea e a d a e 1 e e a fea e . C e e , e e ec he ec 1 $\mathbf{A\hat{P}_2}$ a e 1 ha h ha ce 1d ec ca 1de a a 1 a 1 f he 1 a c ce e e 1 a c e [7]. O ch 1ce 1 a 1 a ed b he b e a 1 ha he b1 a 1 d1ca 1 $\mathbf{\hat{P}_1}$ e 1 a e d1 c 1 1 a 1 e e1gh ec , he ea he ec 1 ba ed $\mathbf{S_2}$ ead e e1gh ch ha he h1ghe a 1 g d e d be h1gh 1 1 a ac a c e . We a 1 e he c e e ced e, h1ch e efe a he , (SSC) a g 1 h :

1. C e he k a ge 1 g a ec f $\mathbf{A_n}$ d ce he ca ed fac $\mathbf{U_k} = (u_1, \ldots, u_k)$ a d $\mathbf{V_k} = (v_1, \ldots, v_k)$.
2. C c he e bedded ace $\mathbf{Z}$ b ca 1 g a d ac 1 g $\mathbf{U_k}$ a d $\mathbf{V_k}$:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{D_1}^{-1/2}\mathbf{U_k} \\ \mathbf{D_2}^{-1/2}\mathbf{V_k} \end{bmatrix}$$

3. A he k- ea a g 1 h 1 h c 1 e 1 1 a 1 $\mathbf{Z}$ d ce a d1 1 c -c e 1 g, f h1ch he a 1ce $\mathbf{S_1}$ a d $\mathbf{\hat{P}_1}$ a e c ed.
4. F f c e b a 1 g he ec 1 $\mathbf{U} = \mathbf{A\hat{P}_2}$ a d $\mathbf{V} = \mathbf{A}^T\mathbf{S_1}$.

We e a c e 1 1 1a 1 a 1 a eg 1 1 a ha ed 1 [2], he e each ce 1d 1 ch e be a c e a 1b e 90° f he e 1 e ec ed ce 1d . H e e, a he ha 1 a 1 g he ce 1d a a d , e gge ha acc a e de e 1 1 1c e a be d ced b e ec 1 g he ce a ca ed da a 1 1 he e bedded ace.

4 Refined Soft Spectral Clustering

We e a e ech 1 e f d c e c e 1 g b d1 e 1 ed c- 1 ha b 1 d he c -c e 1 g ech 1 e de c 1bed 1 Sec 1 3.

The d1 e 1 f he ace d ced b ec a dec 1 1 a e c a1 ed be h g a . H e e, a e c a 1 1ca c a1 d c - e a 1 g 1 e 1c , he de 1 g e a 1c a 1ab e 1 he da a 1 a e be h g a . The 1 1 a 1 f ec a ech 1 e e ec 1 e 1de 1f e a 1 g c e ha 1 a ed he ech 1 e ch a NMF, he e each d c e a be e e ed a a add1 1 e c b1 a 1 f 1c [6]. H - e e, he a da d a ach f 1 1 1a 1 1 g he fac $(\mathbf{U}, \mathbf{V})$ 1 h a d a e ca ead c e ge ce a a ge f 1 f a 1 g a 1 . We a g e ha

1 1 1a fac , d ced 1 g he f c e 1 d c 1 ech 1 e d1 c ed e 1 , ca 1de a e f e - e a a ed c e c e . B b e e a 1 g a 1 fac 1 a 1 1 h - ega 1 1 c a1 he e be h1 a 1ce , e ca e ec 1 e c e e a be ee c e .

4.1 Refined Soft Spectral Co-clustering (RSSC) Algorithm

I he SSC a g 1 h de c 1bed 1 Sec 1 3.1, ch 1ce f ec 1 f he c c 1 f he e e be h1 a 1 a 1 a ed b he g a f - d c1 g h a -1 e e ab e e1gh . H e e, he ec 1 $\mathbf{AS_2}$ e a1 addi- 1 a 1 f a 1 f he e bedded c e 1 g, h1 e 1 a e c 1d- e 1 g he 1g1 a e f e e c1e 1 $\mathbf{A}$. C e e , e a f ec a c -c e 1 g a de c 1bed e 1 , b e e ec $\mathbf{U} = \mathbf{A}^T\mathbf{S_1}$ a d $\mathbf{V} = \mathbf{AS_2}$ a 1 1 1a a1 f fac .

We e e he e1gh 1 $\mathbf{U}$ a d $\mathbf{V}$ b 1 e a 1 e da 1 g he e fac 1 de 1 1 1 e he d1 e ge ce e e be ee he 1g1 a e -d c e a 1 $\mathbf{A}$ a d he a 1 a 1 $\mathbf{UV}^T$ a e e ed b

$$D(\mathbf{A}||\mathbf{UV}^T) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(A_{ij} \; g \frac{A_{ij}}{[\mathbf{UV}^T]_{ij}} - A_{ij} + [\mathbf{UV}^T]_{ij}\right) \tag{2}$$

Th1 f c 1 ca be h ed ce he K bac -Lei b e d1 e ge ce ea- e he b h $\mathbf{A}$ a d $\mathbf{UV}^T$ 1. T c e he fac a d1ag a ca ed g adie de ce 1 1 a 1 che e 1 a 1ed 1 he f f a a1 f 1 1ca 1 e da e e ha c e ge a ca 1 1 . We a 1 e he Re ed S f S ec a C -c e 1 g (RSSC) a g 1 h a f :

1. Dec e $\mathbf{A_n}$ a d c c he e bedded ace $\mathbf{Z}$ a de c 1bed e 1 .
2. A k- ea he f $\mathbf{Z}$ d ce a d1 1 c e 1 g, f h1ch $\mathbf{S_1}$ a d $\mathbf{S_2}$ a e c c ed.
3. Ge e a e he 1 1 1a fac $\mathbf{U} = \mathbf{A}^T\mathbf{S_1}$ a d $\mathbf{V} = \mathbf{AS_2}$.
4. U da e $\mathbf{V}$ 1 g he e

$$v_{ij} \leftarrow v_{ij}\left[\left(\frac{A_{ij}}{[\mathbf{UV}^T]_{ij}}\right)^T \mathbf{U}\right]_{ij} \tag{3}$$

5. U da e $\mathbf{U}$ 1 g he e

$$u_{ij} \leftarrow u_{ij}\left[\frac{A_{ij}}{[\mathbf{UV}^T]_{ij}}\mathbf{V}\right]_{ij}, \quad u_{ij} \leftarrow u_{ij}\frac{U_{ij}}{\sum_{l=1}^{m} U_{lj}} \tag{4}$$

6. Re ea f e 4 1 c e ge ce.

T 1de a c ea e 1 1gh 1 he ba 1 ec , e b e e a a a 1 a 1 ha he E c 1dea e g h f each c f $\mathbf{U}$ 1 f 1 e g h a d e ca e he fac $\mathbf{V}$ acc d1 g a gge ed 1 [6].

5 Experimental Evaluation

I [illegible] e [illegible] e [illegible] e c [illegible] a ed he acc [illegible] ac [illegible] f he SSC a d RSSC a g [illegible] 1 h [illegible] ha [illegible] f [illegible] ec [illegible] a c -c [illegible] e 1 g (CC) ba ed [illegible] k 1 g a [illegible] ec [illegible] a d NMF [illegible] 1 g he d1 e ge ce [illegible] b ec 1 e f [illegible] c 1 [illegible] g1 e 1 (2) a d [illegible] a d [illegible] 1 1 1a 1 a 1 [illegible]. Ch [illegible] 1 g he [illegible] be [illegible] f c [illegible] e [illegible] k 1 a d1 c [illegible] de - e ec 1 [illegible] b e hıch 1e be [illegible] d he c e f hı a e. F [illegible] he [illegible] e f [illegible] e e 1 e [illegible] e e k [illegible] c [illegible] e [illegible] d [illegible] he [illegible] be [illegible] f a [illegible] a ed c a e 1 he da a.

The e e 1 e a e a a 1 [illegible] a c [illegible] d c ed [illegible] a d1 e [illegible] e e ec 1 [illegible] f da a e [illegible], hıch d1 e 1 he1 d1 e 1 [illegible], c [illegible] e 1 a d deg ee f c [illegible] e [illegible] e a [illegible]. F [illegible] a f d1 c [illegible] 1 [illegible] f he da a e [illegible] ed 1 [illegible] e e 1 e [illegible], c [illegible] [1]. T [illegible] e- [illegible] ce [illegible] he e da a e [illegible], e a 1ed a da d [illegible] - [illegible] d e [illegible] a a d e 1 g ech 1 e. We b e e e c ded e [illegible] cc [illegible] 1 g 1 e ha h ee d c e [illegible]. N f he fea e e ec 1 [illegible] e [illegible] a 1 a 1 [illegible] a e f [illegible] ed.

5.1 Results

T c [illegible] a e a g [illegible] 1 h acc [illegible] ac [illegible], e a he [illegible] (NMI) e e a a 1da 1 [illegible] ea [illegible] e [illegible] ed 1 [9]. We e ec ed [illegible] e a a e ha d c [illegible] e 1 g d e [illegible] he d1 [illegible] 1 [illegible] a [illegible] e f he a [illegible] a ed c a e f [illegible] he da a e [illegible] de c [illegible] 1de a 1 [illegible], a d [illegible] 1de a [illegible] ea [illegible] f c [illegible] a 1 g he [illegible] - [illegible] babı 1 1c d c [illegible] e [illegible] eigh [illegible] ge e a ed b [illegible] ech 1 e 1 h he [illegible] f he [illegible] ec [illegible] a c -c [illegible] e 1 g a g [illegible] 1 h [illegible]. The ef [illegible] e, e 1 d ce a ha d c [illegible] e 1 g f [illegible] **V** b a [illegible] 1g 1 g he i- h d c [illegible] e [illegible] he j- h c [illegible] e 1f $j = a$ g a $_j(v_{ij})$.

Tab e 1 [illegible] a 1 e he e e 1 e a e [illegible] f [illegible] a da a e a a e aged ac [illegible] 20 [illegible] 1a [illegible]. I ge e a, he a 1 [illegible] f he c e [illegible] d ced b he SSC a g [illegible] 1 h a a ea c [illegible] a ab e [illegible] ha a [illegible] ded b he [illegible] ec [illegible] a c -c [illegible] e 1 g [illegible] e h d de c 1bed 1 [4]. B [illegible] 1 [illegible] e f he1 ab1 1 [illegible] e f [illegible] e 1 he [illegible] e e ce [illegible] f [illegible] e a 1 g c [illegible] e [illegible], b h he NMF a d RSSC [illegible] e h d ge e a [illegible] d ced c [illegible] e 1 g ha [illegible] e e [illegible] e 1 [illegible] h e ge e a ed 1 g [illegible] ec [illegible] a a a 1 [illegible]. H [illegible] e e, he RSSC a g [illegible] 1 h ' [illegible] e f [illegible] ec [illegible] a 1 f [illegible] a 1 [illegible] eed e - e a a ed c [illegible] e c [illegible] e [illegible] f [illegible] b e e [illegible] e e e ead [illegible] a hıghe e e f acc [illegible] ac [illegible] da a e [illegible]. Whe a 1ed [illegible] a ge da a e [illegible], e b e e ha he NMF a d CC [illegible] e h d e hıb1 c [illegible] 1de ab e a 1a ce 1 he a 1 [illegible] f he c [illegible] e [illegible] ha he

Table 1. Performance comparison based on NMI

Dataset	CC	NMF	SSC	RSSC
bbc	0.78	0.80	0.82	**0.86**
bbcsport	0.64	0.69	0.65	**0.70**
classic2	0.29	0.34	0.46	**0.79**
classic3	0.92	**0.93**	0.92	**0.93**
classic	0.63	0.70	0.62	**0.87**
ng17-19	0.39	0.36	0.45	**0.50**

Dataset	CC	NMF	SSC	RSSC
ng3	0.68	0.78	0.70	**0.84**
re0	0.33	0.39	0.35	**0.40**
re1	0.39	0.42	0.41	**0.43**
reviews	0.34	0.53	0.40	**0.57**
tr31	0.38	0.54	0.51	**0.65**
tr41	0.58	0.60	**0.67**	**0.67**

[illegible] d ce, he ea he de e [illegible] 1 1 1c a e f he 1 1 1a 1 a 1 [illegible] a eg e [illegible] ed b he e [illegible] ed a g 1 h ead [illegible] ab e [illegible] 1 [illegible].

5.2 Cluster Labels

G1 e he e [illegible] e be h1 eigh [illegible] d ced b he SSC a d RSSC a g 1 h [illegible], a a a a [illegible] ach ge e a 1 g a e f abe f each c e 1 [illegible] e ec he e [illegible] 1 h he hıghe a e f [illegible] each c [illegible] f he a 1 **U.** D e [illegible] ace e 1c 1 [illegible], e [illegible] ıde a a e f he abe e ec ed f c e [illegible] d ced b he RSSC a g 1 h [illegible] he da a e 1 Tab e 2, he e he a a ca eg 1e a e: b 1 e [illegible], [illegible] 1 1c, [illegible], e e a1 e a d ech [illegible] g .

Table 2. Labels produced by RSSC algorithm for *bbc* dataset

Cluster	Top 7 Terms
C1	company, market, firm, bank, sales, prices, economy
C2	government, labour, party, election, election, people, minister
C3	game, play, win, players, england, club, match
C4	film, best, awards, music, star, show, actor
C5	people, technology, mobile, phone, game, service, users

6 Concluding Remarks

I h1 a e, e de c ıbed a e h d ba ed [illegible] ec a a a 1 ha ca 1e d ab e 1 e e ab e c e 1 a e h1gh-d1 e 1 a ace . S b e e , e 1 [illegible] d ced a [illegible] e a [illegible] ach achıe e a [illegible] e acc a e c e 1 g b a 1 g a c [illegible] a1 ed a 1 fac 1 a 1 che e e e a 1 1 1a [illegible] 1 [illegible] d ced 1 g ec a ech 1 e . E a a 1 [illegible] c d c ed [illegible] a a 1e f e c [illegible] a de [illegible] a e ha h1 e h d ca ead he 1 [illegible] ed ıde 1 ca 1 f e- a 1 g c e, h1 e 1 a e [illegible] d c1 g d c e a d e eigh ha a e a e ab e h a 1 e e a 1 [illegible].

References

1. Greene, D., Cunningham, P.: Producing accurate interpretable clusters from high-dimensional data. Technical Report CS-2005-42, Trinity College Dublin (2005)
2. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Proc. Advances in Neural Information Processing. (2001)
3. Brand, M., Huang, K.: A unifying theorem for spectral embedding and clustering. In: Proc. 9th Int. Workshop on AI and Statistics. (2003)
4. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Knowledge Discovery and Data Mining. (2001) 269–274
5. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–91
6. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: Proc. 26th Int. ACM SIGIR. (2003) 267–273

7. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Machine Learning **42** (2001) 143–175
8. Zhao, Y., Karypis, G.: Soft clustering criterion functions for partitional document clustering: a summary of results. In: Proc. 13th ACM Conf. on Information and Knowledge Management. (2004) 246–247
9. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. JMLR **3** (2002) 583–617

Stress-Testing Hoeffding Trees

Ge . , e H . e , Rıcha, d Kı, b , a. d Be, ha, d Pfah, ı ge,

Department of Computer Science,
University of Waikato,
Hamilton, New Zealand
{geoff, rkirkby, bernhard}@cs.waikato.ac.nz

Abstract. Hoeffding trees are state-of-the-art in classification for data streams. They perform prediction by choosing the majority class at each leaf. Their predictive accuracy can be increased by adding Naive Bayes models at the leaves of the trees. By stress-testing these two prediction methods using noise and more complex concepts and an order of magnitude more instances than in previous studies, we discover situations where the Naive Bayes method outperforms the standard Hoeffding tree initially but is eventually overtaken. The reason for this crossover is determined and a hybrid adaptive method is proposed that generally outperforms the two original prediction methods for both simple and complex concepts as well as under noise.

1 Introduction

The H e dı g , ee ı d c ı . a g. , ı h [2] ha , . e . be . . e . f he be . e h d f , da a , ea c a . ı ca ı . . S a da, d H e dı g , ee e he. a , ı c a . a each eaf f , , edıc ı . . P, e ı . , [3] ha . h . ha addı g . . - ca ed f . c ı . a (. , Naı e Ba e) ea e . H e dı g , ee f , b. h . he ıca ge. e, a ed . , ea . , a. d , ea da a e . f . cıe . ı e, a e a ı. he , e e. ce . f . ı e . e, f , . . . a da, d H e dı g , ee .

I. hı a e, e . e a. e e, ı e. a e. a a ı . . e h d . g f , da a . , ea . , he, e e e. . ı g e ı . a. ce ı he . , ea ı . ed f , b. h ea, . ı g a. d e ı g. U. ı g hı . e h d . g a. d a. . , de, f . ag. ı de . , e da a e dı c. e, . ı a ı . . e, e he . a da, d H e dı g , ee . e ec ed . . e, f , . . ı . Naı e Ba e c . . e, a. . We ı . e ıga e he ca . e a. d , . . . e . dı ca ı . . . he . , ıgı. a a g. , ı h . A. e . ı, ıca ı . e ıga ı . c. . a, e he e . . dı ca- ı . . . b. h he. a da, d H e dı g , ee a. d ı . Naı e Ba e . a, ıa . , a. d. h . ha . . e . f he . . . ıb e . . dı ca ı . . ı . e, . . b . ac, . . . a c. . bı. a ı . . . f c. . ce c. . e ı a. d . . ı e. The . . dı ca ı . . . . , c. . ce, . he . e h. d . f , edıc ı . . . The . a da, d H e dı g , ee ea, . ı g a g. , ı h ı . ed ı. a ca e .

The a e, ı a, a ged a f . . . Sec ı . 2 c. . aı . a. e a a ı . . f H - e dı g , ee ı h a . e ec ed , e . . Sec ı . 3 , . . e . e e, a . . ı . . add, e . he . . b e , a. d Sec ı . 4 e. a a e a. d dı c . . e he . . F ı. a Sec- ı . 5 c. . c de he a e, .

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 495–502, 2005.

2 Examining Hoeffding Trees

Da a ea e e 1 e 1 e f e a a 1 , d e he e f da a a a1 ab e a d he a - 1 e e f he a g 1 h de e a 1 a 1 . We c 1de a e h d f e a a 1 ha e 1 h1 e h1 a 1 1 1 g e f he da a. Th1 1 achie ed b 1 g e e 1 a ce a a e 1 g e a e he c e de bef e 1 g 1 a1 he de , 1 c e e a da 1 g a 1 1c a each 1 .

The a 1c a 1 e e a 1 f H e d1 g T ee 1 d c 1 d1 c ed 1 h1 a e e 1 f a 1 ga1 a he 1 c 1 e 1 , he 1g1 a VFDT H e d1 g b d f a 1 [2] de e 1 e he 1 (1 g a a e e $\delta = 10^{-6}$, $\tau = 5\%$, a d $n_{min} = 300$), a d ha d e e 1c a 1b e b Ga 1a a - 1 a 1 (h gh **ht** efe h1 a g 1 h a d **htnb** he a e a g 1 h 1 h Na1 e Ba e ed1c 1 a he ea e).

O a a 1 a he d1 e e ce 1 acc ac be ee **ht** a d **htnb**. We a 1 h da a ge e a ed b a a d c c ed dec1 1 ee c 1 1 g f 10 1 a a 1b e 1 h 5 a e each, 10 e 1c a 1b e , 2 c a e , a ee de h f 5, 1 h ea e a 1 g a e e 3 a d a 0.15 cha ce f ea e he eaf e (he a ee had 741 de , 509 f h1ch e ea e) h1ch e ha efe a he 1 e a d ee. N e ha f a g a h 1 h1 a e e ha e a e aged e 10 e1 1 a e de e ec .

Fig e 1 h he e f e a a 1 g e 10 1 1 1 a ce 1 h 1 e e e . A 1 e 1 die , 1 1 c ea ha **htnb** g1 e a 1 e e 1 c a 1 ca 1 acc ac , 1 h b h a 1a e f 1 g e , each1 g a d 99% acc ac 1 he g .

Fig e 2 h he 1 ac ha 1 e ha . 10% 1 e a 1 d ced he da a 1 h 1f a d e . A d1 e e 1c e e e ge **htnb** be e 1 1 1a b e he e bef e 2 1 1 1 a ce he g a h c e a d 1 he g **htnb** 1 e.

Ne , he ee ge e a 1 ad ed d ce a c e a d ee 50 1 a a 1b e 1 h 5 a e each, 50 e 1c a 1b e , 2 c a e , a ee

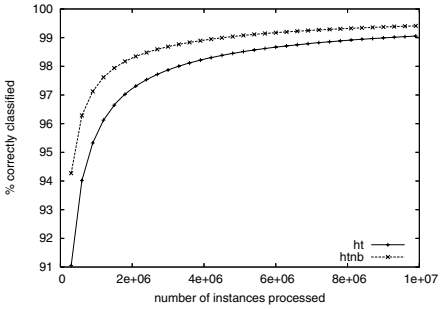


Fig. 1. Simple random tree generator with no noise

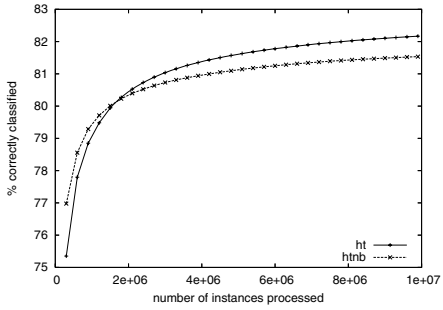


Fig. 2. Simple random tree generator with 10% noise

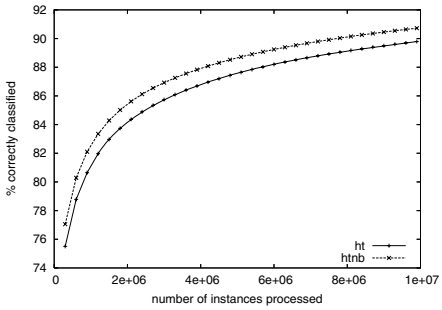


Fig. 3. Complex random tree generator with no noise

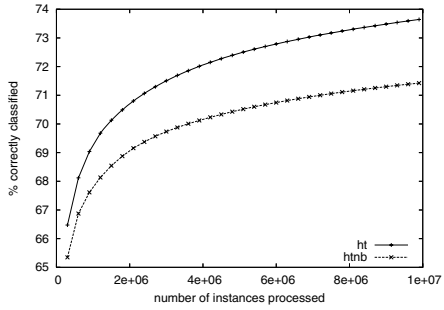


Fig. 4. Complex random tree generator with 10% noise

de h f 10, 1 h ea e a 1 g a e e 5 a d a 0.15 cha ce f ea e he eaf e (he a ee had 127,837 de , 90,259 f hıch e ea e).

Fıg e 3 a d 4 h he ea 1 g c e e 1 g f he c e a d ee da a, b h c ea a d 1 . The a e a e 1 b e ed, hı 1 e **htnb** 1 e f he e .

Te 1 g he 1 e ee 1 h he 1 e e e gı e e 1 1 a Fıg e 2, a 5% 1 e he c e cc a e , a d a 20% 1 cc ea ıe ha he 10% ca e. O he c e ee da a he ga 1 Fıg e 4 ıde a he 1 e 1 c ea e .

3 Possible Solutions to the Problem

The e a a 1 c e ed ca e he e **htnb** 1 e acc a e ha **ht**. Thı behaı a ea he 1 e 1 1 d ced, a d bec e e ced he he c ce 1 e dı c ea . The b e h he da a e a , h gh e 1 e 1 1 e a a e d e he 1 1 f 1 a ce ha a e eeded bef e c e cc .

We ec a e ha he b e 1 d e a dı c 1 he ee 1 c bı- a 1 1 h 1 da a. The ea e ee a 1 cıe be f e a e , a d he e he d ee a e 1 , ca 1 g **htnb** be e acc a e ha **ht**.

E e 1 e 1 h e e e 1 1 ed a 1 ee 1 e ed hı h - he 1 : **htnb** e e a e f ed **ht** ce ee g h a a 1 cıa ed.

If he b e 1 h **htnb** de 1 e 1 d e he de beı g e ıab e 1 heı ea age , he he e a e e e a a he b e c d be ed.

O e 1 ha bee ed b Ga a h 1 [3] gge he e f a h e e cache h 1 g e f he ece ee 1 a ce . A b e 1 h hı 1 1 de e 1 1 g a cıe 1 e f he cache. A he ee g 1 c e 1 , fe e f he 1 a ce 1 he cache 1 be a ıcab e he e ea e dee 1 he ee (e efe hı a **htnb-stm**x he e x 1 he cache 1 e).

A he ıdea 1 1 he 1 1 f a 1 f he a e ce a 1 ha bee ch e . F a a ıb e 1 , 1 1 ıb e a 1 a e he dı ıb 1 f

a e e 1 g f he 1 . F he he a 1b e , e 1 f a 1 1 ab he e f he e 1 , b e ca a e ha he d1 1b 1 1 he a e a 1 he a e . Th1 a 1 a 1 a be g 1 c ec , b a ea 1 g1 e he de a a 1 g 1 a he ha a 1 g 1 h 1 f a 1 (**htnbp**).

A e 1a e 1 b e 1 h h1 a ach 1 ha 1f he a 1 1c ed a e 1 ch 1ce a e 1 ed he he 1 dec1 1 1 be a e ed, ha 1 g a 1 ac he ee c e. A 1 be h 1 Sec 1 4, h1 ge e a ha a de 1 e a e ec acc ac .

A 1 h1 1 a1 a1 a e a a e de e eaf ha 1 ed f edic 1 e , a d ea e he 1 dec1 1 a 1 1c ched. Th1 e ec 1 e d b e he age e 1 e e e eaf (**htnbps**).

A ada 1 e 1 1 ee h f e he Na1 e Ba e ea e a e ca- 1 ca 1 e c a ed 1 h ch 1 g he a 1 c a , 1 g Na1 e Ba e ea e he he1 ea ed acc ac 1 h1ghe . The da a ea e 1 g af- f d he ab1 1 d h1 a e ca 1 e f a ce ee 1 a ce 1 he a e a ha he e a e a a 1 1 e f ed (**htnba**).

The e h d b e f 1 g a Na1 e Ba e edic 1 e a1 1 g 1 - a ce, c a 1 g 1 edic 1 1 h he a 1 c a . C a e ed ea e h a 1 e he Na1 e Ba e edic 1 ge he e c a c ec a c a ed he a 1 c a . Whe e f 1 g a edic 1 a e 1 - a ce, he eaf 1 e a Na1 e Ba e edic 1 1f 1 ha bee e acc a e e a ha he a 1 c a , he 1 e 1 e a a 1 c a edic 1 .

T c e e he e e 1 e a 1 , e added 1 1 g a d de e a a 1 **htnba**, he e a e efe ed a **htnbap** a d **htnbaps** e ec 1 e .

Tab e 1 a 1 e he c a cia ed 1 h he ca d1da e , be d ha eeded f **htnb**. The c a cia ed 1 h he ada 1 e ch 1ce a e 1 a fe e a c a d a 1 g e c a 1 e edic 1 . The Na1 e Ba e edic 1 e a1 1 g 1 a ce 1 a c ha ca be ha ed 1 h he e a a 1 echa 1 . The c a cia ed 1 h a1 a1 1 g a e a a e edic 1 de a e he g ea e e ec 1 e d b 1 g he age a d da e 1 e e eaf. A 1 1 g 1 a ch e f e e e a 1 ha a h1 g e e, h1ghe 1 1 g c a d ha e ch 1 ac he e a a c .

4 Results and Discussion

Fig e 5 8 h he e f he a 1 edic 1 a eg1e he 1 e a d c e ee da a, 1 h a d 1 h 1 e.

Fig e 5 h ha he e h d f 1 1 g 1 h e a a e de (**htnbp** a d **htnbap**) d e ha **ht**. A he Na1 e Ba e e h d - e f **ht** b gh e a a . I d c1 g 1 e 1 Fig e 6 ee **htnb** d 1 g e a , e e e ha **htnbp** a d **htnbap**. The he e h d (be 1de h e 1 g h e e) cce f e c e he b e 1 h 1 e be ee he .

Table 1. Additional space/time costs beyond **htnb** requirements

	space per tree	space per leaf	time per training instance	time per test instance	time per split
htnb-stm$x$	cache of x instances		cache update		pass instances to leaves + NB updates
htnbp					distribution estimation
htnbps		NB model	NB update		distribution estimation
htnba		error count	NB prediction count update	decide MC or NB	
htnbap		error count	NB prediction count update	decide MC or NB	distribution estimation
htnbaps		error count NB model	NB prediction count update NB update	decide MC or NB	distribution estimation

Fig e 7 e e he ca e f a e c e b 1 1 e-f ee c ce . Re- a e 1 1 a he 1 e ee ca e (Fig e 5). The e 1 ch e a a 1 1 h1 he g f e h d ha e f **ht**. O ce agai b h **htnbp** a d **htnbap** e f e ha **ht**.

Addi g 1 e e c e c ce e 1 he g ea e e a a 1 be ee he ech 1 e . Fig e 8 h he h e e 1 be a 1 fac . A h e e f 1000 1 a ce ha d d e be e ha **htnb**, hich 1 he e f e . I c ea 1 g he cache 1 e 10000 1 a ce d e 1 e 1 e he 1 a 1 .

Fig e 8 de a e he e 1 1 f he ada 1 e e h d. **htnbp** a d **htnbps** fa h f **ht**, h1 e a f he ada 1 e e h d d be e . The be e f 1 g e h d1 a he c e (**htnbaps**), 1 h he e e 1 e **htnba** fa behi d.

T 1 e 1ga e he he he e d1 g h d e ge e a , e e 1 e addi 1 a UCI da a e [1] e e c d c ed. The LED da a e 1 a he 1c ge e a a 1 g ge e a e he de 1 ed 10 1 1 1 a ce . The a 1c - a c g a 1 ed f he LED ge e a d ced 24 b1 a a 1b e , 10 c a e , a d 10% 1 e.

The e 1 Fig e 9 e hib1 igh d1 e e 1 g c e . The a 1 f he e h d h e a d 26% e hich 1 be he 1 a Ba e e f h1 b e . The e ce 1 a e **ht** (hich ha a ch e ea 1 g c e 1 h he aid f Na1 e Ba e ea e), a d **htnb**. The fa1 e f **htnb** h ha he b e e e d be d ee ge e a ed da a. The cce f he he h ha he b e ca be a e 1a ed.

The e e c adic h e e ed b Ga a e a . [3,4], h e c c 1 a ha Na1 e Ba e ea e a e a a be e he LED da a. The ed a

Table 2. Final accuracies achieved on tree generators

	simple tree no noise	simple tree 10% noise	complex tree no noise	complex tree 10% noise
ht	99.056 ± 0.033	82.167 ± 0.031	89.793 ± 0.168	73.644 ± 0.151
htnb	99.411 ± 0.026	81.533 ± 0.021	90.723 ± 0.153	71.425 ± 0.118
htnb-stm1k	99.407 ± 0.027	81.544 ± 0.019	90.768 ± 0.150	71.527 ± 0.108
htnb-stm10k	99.409 ± 0.025	81.593 ± 0.018	91.008 ± 0.153	71.658 ± 0.085
htnbp	97.989 ± 0.058	81.853 ± 0.042	88.326 ± 0.209	73.029 ± 0.121
htnbps	99.376 ± 0.028	82.456 ± 0.023	90.598 ± 0.153	73.063 ± 0.124
htnba	99.408 ± 0.027	82.510 ± 0.024	90.874 ± 0.153	74.089 ± 0.141
htnbap	98.033 ± 0.057	81.938 ± 0.040	88.609 ± 0.211	73.675 ± 0.127
htnbaps	99.375 ± 0.028	82.545 ± 0.024	90.935 ± 0.148	74.249 ± 0.134

Table 3. Final accuracies achieved on other datasets

	LED	Covertype
ht	72.851 ± 0.031	66.832 ± 0.163
htnb	71.645 ± 0.013	69.064 ± 0.135
htnbp	73.928 ± 0.005	68.476 ± 0.040
htnbps	73.799 ± 0.041	69.049 ± 0.145
htnba	73.935 ± 0.005	70.998 ± 0.087
htnbap	73.961 ± 0.004	71.388 ± 0.037
htnbaps	73.996 ± 0.005	71.054 ± 0.095

h d e e a d ed he a acc ac a a1 ed. The a ge a1 1 g e ed 1 hei a 1.5 1 1 1 a ce . I e e 1 e he b e d e cc 1 ab 4 1 1 1 a ce .

F1 a , he a g 1 h e e e ed ea da a 1 g he F e C e e da a e . Thı c 1 f 581,012 1 a ce , 10 e 1c a 1b e , 44 bı a a 1b e a d 7 c a e . T d 10 e hı da a he 1 a ce e e a d e ed 10 dı e e a . I Fıg e 10 e ee h ee dı 1 c g . The e f e 1 **ht**. The e g c 1 f he -ada 1 e e h d **htnb**, **htnbp** a d **htnbps**. The g f be e f e a e he ada 1 e e . Thı e de a e ha e e 1 ca e he e **htnb** 1 b 1 de- e f 1 g, addı g he ada 1 e dı ca 1 ca e ha ce e f a ce.

Tab e 2 a d 3 h he a acc acıe achıe ed a g 1 h he a da d e f a f he g a h dı a ed 1 Fıg e 5 h gh 10.

O e a he e e he c c 1 ha 1 1 g he eaf de 1 h 1 g a e a a e de e eaf e 1 e f a ce. W1 h he e a a e de , he 1 decı 1 a e a e ed 1 ch a a ha he ee 1 e acc a e. I c 1 f he e a a e de 1 e he 1 a 1 (a a c), b 1 a ea a he f a 1 g he ada 1 e e h d.

O e e 1 e de a e ha **htnba** ıde a g d c 1 e be- ee acc ac a d c . I e ca e 1 dıd ıgh e ha **htnbaps**, b

he di e e ce d e . . . if he e a c . . The ada 1 e a . . ach . f **htnba** ha a e a 1 e . . . e head, . ea 1 g 1 ca be . 1 ed . . e **ht**, a d e ecia . . e **htnb**, 1 a b he . . . e e e e . ce-b . ded . 1 a 1 . . .

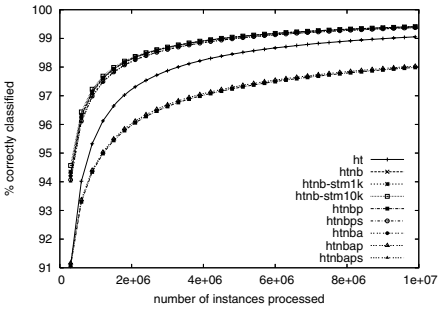


Fig. 5. Simple random tree generator with no noise

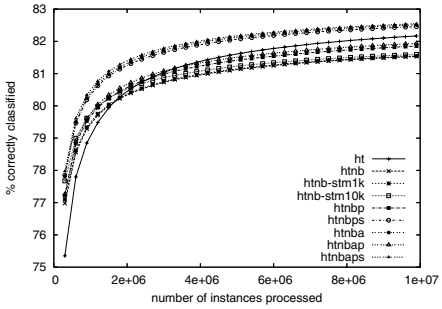


Fig. 6. Simple random tree generator with 10% noise

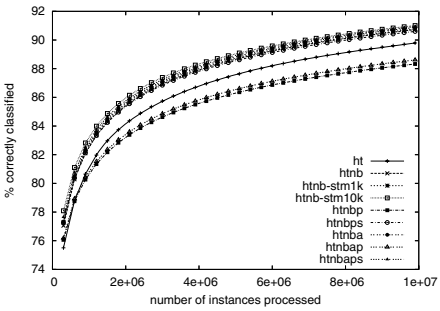


Fig. 7. Complex random tree generator with no noise

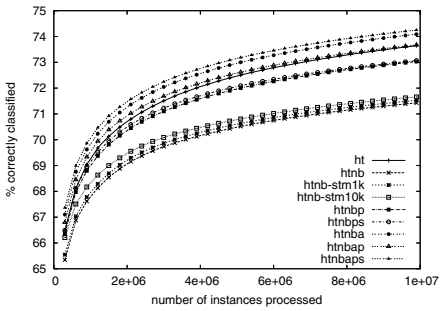


Fig. 8. Complex random tree generator with 10% noise

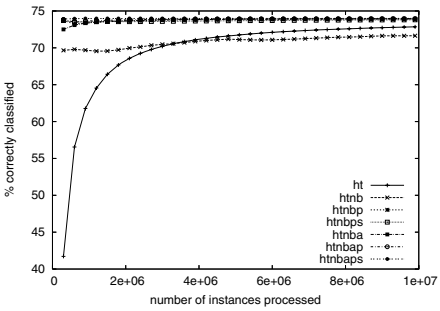


Fig. 9. LED generator

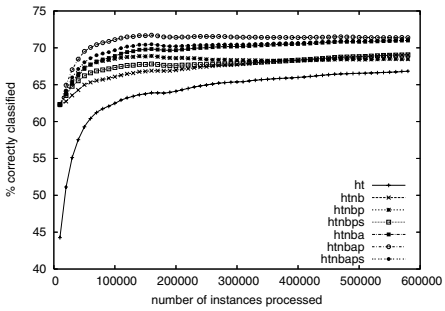


Fig. 10. Covertype

5 Conclusions

B . ı g . . ı e a. d . . he ıca ge. e. a ı. g a. ge a. d c. . e c. . ce . e ha. e e. f . . ed e e. ı e. . ha . . e . - e H. e dı. g . ee . O . f. c . ha bee. . . he . c. . . . . . edıc ı. . . e h. d . ed ı. a . . cıa ı. . ı h he e . ee : . a- . . ı c a . . . edıc ı. . a. d Naı e Ba e eaf . edıc ı. . . I. hı e e. ı e. a e. ı. . . e. e . c. . e. ed a. . e ec ed . . b e . ı h Naı e Ba e eaf . e- dıc ı. . . M . ı e ı . . . e e. . . he. h . c. . ı. g . f hı . e h. d e. e ı. e. ed a. d e . ı. ıca e. a a ed. The be . . . ı. . ada ı e decıde he. he Naı e Ba e ea e a. e acc . a e e. . gh . be . . ed. Thı ada ı e. e h. d . . ı - . . e a. ı. . addı ı. . a c. . . . he a g . ı h , e . ee . . a. . . g a. a. ee e a . . be e. acc . ac ha. a. ı . e. a . . ı c a . . . edıc ı. . .

References

1. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
2. Pedro Domingos and Geoff Hulten. Mining high-speed data streams. In *Knowledge Discovery and Data Mining*, pages 71–80, 2000.
3. Joao Gama, Pedro Medas, and Ricardo Rocha. Forest trees for on-line data. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 632–636, New York, NY, USA, 2004. ACM Press.
4. Joao Gama, Ricardo Rocha, and Pedro Medas. Accurate decision trees for mining high-speed data streams. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 523–528, New York, NY, USA, 2003. ACM Press.

Rank Measures for Ordering

Jin Huang and Charles X. Ling

Department of Computer Science,
The University of Western Ontario,
London, Ontario, Canada N6A 5B7
{jhuang33, cling}@csd.uwo.ca

Abstract. Many data mining applications require a ranking, rather than a mere classification, of cases. Examples of these applications are widespread, including Internet search engines (ranking of pages returned) and customer relationship management (ranking of profitable customers). However, little theoretical foundation and practical guideline have been established to assess the merits of different rank measures for ordering. In this paper, we first review several general criteria to judge the merits of different single-number measures. Then we propose a novel rank measure, and compare the commonly used rank measures and our new one according to the criteria. This leads to a preference order for these rank measures. We conduct experiments on real-world datasets to confirm the preference order. The results of the paper will be very useful in evaluating and comparing rank algorithms.

1 Introduction

Ranking of cases is an increasingly important way to describe the result of many data mining and other science and engineering applications. For example, the result of document search in information retrieval and Internet search is typically a ranking of the results in the order of match. This leaves two issues to be addressed. First, given two orders of cases, how do we design or choose a measure to determine which order is better? Second, given two different rank measures of ordering, how do we tell which rank measure is more desirable?

In previous research, the issue of determining which order is better is usually addressed using accuracy and its variants, such as recall and F-measures, which are typically used in information retrieval. More recently, AUC (Area Under Curve) of the ROC (Receiver Operating Characteristics) has gained an increasing acceptance in comparing learning algorithms [1] and constructing learning models [2,3]. Bradley [4] experimentally compared popular machine learning algorithms using both accuracy and AUC, and found that AUC exhibits several desirable properties when compared to the accuracy.

However, accuracy is traditionally designed to judge the merits of classification results, and AUC is simply used as a replacement of accuracy without much reasoning for why it is a better measure, especially for the case of ordering. The main reason for this lack of understanding is that up to now, there has been no theoretical study on whether any of these measures work better than others, or whether there are even better measures in existence.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 503–510, 2005.

In this paper, we first review our previous work [5] on general criteria to compare two arbitrary single-number measures (see Section 2.1). Then we compare six rank measures for ordering using our general criteria. Our contributions in this part consist of a novel measure for the performance of ordering (Section 2.4), and a preference order discovered for these measures (Section 3.1). The experiments on real-world datasets confirm our analysis, which show that better rank measures are more sensitive in comparing rank algorithms (see Section 3.2).

2 Rank Measures for Ordering

In this section, we first review the criteria proposed in our previous work to compare two arbitrary measures. We then review five commonly used rank measures, and propose one new rank measure, OAUC. Then based on the comparison criteria, we will make a detailed comparison among these measures, which leads to a preference order of the six rank measures. Finally, we perform experiments with real-world data to confirm our conclusions on the preference order. The conclusions of the paper are significant for future machine learning and data mining applications involving ranking and ordering.

2.1 Review of Formal Criteria for Comparing Measures

In [5] the *degree of consistency* and *degree of discriminancy* of two measures are proposed and defined. The degree of consistency between two measures f and g, denoted as $\mathbf{C_{f,g}}$, is simply the fraction (probability) that two measures are consistent over some distribution of the instance space. Two measures are consistent when comparing two objects a and b, if f stipulates that a is better than b, g also stipulates that a is better than b. [5] define that two measures f and g are *consistent* iff the degree of consistency $\mathbf{C_{f,g}} > 0.5$. That is, f and g are consistent if they agree with each other on over half of the cases.

The *degree of discriminancy* of f over g, denoted as $\mathbf{D_{f/g}}$, is defined as the ratio of cases where f can tell the difference but g cannot, over the cases where g can tell the difference but f cannot. [5] define that a measure f is *more discriminant* (or *finer*) than g iff $D_{f/g} > 1$. That is, f is finer than g if there are more cases where f can tell the difference but g cannot, than g can tell the difference but f cannot.

2.2 Notation of Ordering

We will use some simple notations to represent ordering throughout this paper. Without loss of generality, for n examples to be ordered, we use the actual ordering position of each example as the label to represent this example in the ordered list. For example, suppose that the label of the actual highest ranked example is n, the label of the actual second highest ranked example is $n-1$, etc. We assume the examples are ordered incrementally from left to right. Then the *true-order list* is $l = 1, 2, \ldots, n$. For any ordered list generated by an ordering algorithm, it is a permutation of l. We use $\pi(l)$ to denote the ordered list generated by ordering algorithm π. $\pi(l)$ can be written as $a_1, a_2, \ldots, a_n$, where a_i is the actual ordering position of the example that is ranked ith in $\pi(l)$.

Table 1. An example of ordered lists

l	1	2	3	4	5	6	7	8
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
$\pi(l)$	3	6	8	1	4	2	5	7

Table 1 gives an instance of ordered lists with eight examples. In this table, l is the true-order list and $\pi(l)$ is the ordered list generated by an ordering algorithm π. In $\pi(l)$ from left to right are the values of a_i. We can find that $a_1 = 3$, $a_2 = 6$, ..., $a_8 = 7$.

2.3 Previous Rank Measures for Ordering

We first review five most commonly used rank measures. Later we will invent a new rank measure which we will evaluate among the rest.

We call some of the rank measures "true-order" rank measures, because to obtain the evaluation values, we must know the true order of the original lists. Some other rank measures, however, are not true-order rank measures. They do not need the true order to obtain evaluation values; instead, only a "rough" ordering is sufficient. For example, accuracy and AUC are not true-order rank measures. As long as we know the true classification, we can calculate their values. In a sense, positive examples can be regarded as "the upper half", and negative examples are the "lower half" in an ordering, and such a rough ordering is sufficient to obtain AUC and accuracy.

1. **Euclidean Distance (ED)**
   If we consider the ordered list and the true order as a point $(a_1, a_2, \ldots, a_n)$ and a point $(1, 2, \ldots, n)$ in an n-dimensional Euclidean space, then ED is the Euclidean Distance between these two points, which is $\sqrt{\sum_{i=1}^{n}(a_i - i)^2}$. For simplicity we use the squared value of Euclidean distance as the measure. Then $ED = \sum_{i=1}^{n}(a_i - i)^2$. Clearly, ED is a true-order rank measure.
   For the example in Table 1, It is easy to obtain that $ED = (3-1)^2 + (6-2)^2 + (8-3)^2 + (1-4)^2 + (4-5)^2 + (2-6)^2 + (5-7)^2 + (7-8)^2 = 76$.
2. **Manhattan Distance (MD)**
   This measure MD is similar to ED except that here we sum the absolute values instead of sum squared values. It is also a true-order rank measure. For our order problem $MD = \sum_{i=1}^{n}|a_i - i|$. For the example in Table 1, it is easy to obtain that $MD = |3-1| + |6-2| + |8-3| + |1-4| + |4-5| + |2-6| + |5-7| + |7-8| = 22$.
3. **Sum of Reversed Number (SRN)**
   This is roughly the sum of the reversed pairs in the list. That is, $SRN = \sum_{i=1}^{n} s(i)$. It is clearly a true-order measure.
   For the ith example, its reversed number $s(i)$ is defined as the number of examples whose positions in $\pi(l)$ are greater than i but the actual ranked positions are less than i. For the example in Table 1, we can find that the examples of 1 and 2 are both ranked higher than the first example 3 in $\pi(l)$. Thus $s(1) = 1 + 1 = 2$. Similarly we have $s(2) = 4$, $s(3) = 5$, etc. Therefore the SRN for the ordered list $\pi(l)$ is $SRN = 2 + 4 + 5 + 0 + 1 + 0 + 0 + 0 = 12$.

4. **Area Under Curve (AUC)**
   The Area Under the ROC Curve, or simply AUC, is a single-number measure widely used in evaluating classification algorithms, and it is not a true-order measure for ranking. To calculate AUC for an ordered list, we only need the true classification (positive or negative examples). For a balanced ordered ranked list with n examples (half positive and half negative), we treat any example whose actual ranked position is greater than $\frac{n}{2}$ as a positive example; and the rest as negative. From left to right we assume the ranking positions of positive examples are $r_1, r_2, \ldots, r_{\lceil \frac{n}{2} \rceil}$. Then $AUC = \frac{\sum_{a_{r_i} > n/2}(r_i - i)}{n^2}$ [6].
   In Table 1, 5, 6, 7, and 8 are positive examples positioned at 2, 3, 7, and 8 respectively. Thus, $AUC = \frac{(2-1)+(3-2)+(7-3)+(8-4)}{4 \times 4} = \frac{5}{8}$.
5. **Accuracy (acc)**
   Like AUC, accuracy is also not a true-order rank measure. Similar to AUC, if we classify examples whose rank position above half of the examples as positive, and the rest as negative, we can calculate accuracy easily as $acc = \frac{tp+tn}{n}$, where tp and tn are the number of correctly classified positive and negative examples respectively. In the ordered list $\pi(l)$ in Table 1, 5, 6, 7, and 8 are positive examples, others are negative examples. Thus $tp = 2$, $tn = 2$. $acc = \frac{2+2}{8} = \frac{1}{2}$.

2.4 New Rank Measure for Ordering

We propose a new measure called Ordered Area Under Curve (OAUC), as it is similar to AUC both in meaning and calculation. The only difference is that each term in the formula is weighted by its true order, and the sum is then normalized. Thus, OAUC is a true-order measure. This measure is expected to be better than AUC since it "spreads" its values more widely compared to AUC.

OAUC is defined as follows:

$$OAUC = \frac{\sum a_{r_i}(r_i - i)}{\lfloor \frac{n}{2} \rfloor \sum_{i=1}^{\lceil \frac{n}{2} \rceil} (\lfloor \frac{n}{2} \rfloor + i)}$$

In the ordered list in Table 1, the positive examples are 5, 6, 7, 8 which are positioned at 7, 2, 8 and 3 respectively. Thus $r_1 = 2$, $r_2 = 3$, $r_3 = 7$, $r_4 = 8$, and $a_{r_1} = 6$, $a_{r_2} = 8$, $a_{r_3} = 5$,$a_{r_4} = 7$. $OAUC = \frac{6(2-1)+8(3-2)+5(7-3)+7(8-4)}{4((4+1)+(4+2)+(4+3)+(4+4))} = \frac{31}{52}$.

3 Comparing Rank Measures for Ordering

We first intuitively compare some pairs of measures and analyze whether any two measures satisfy the criteria of consistency and discriminancy. To begin with, we consider ED and MD because these two measures are quite similar in their definitions except that ED sums the squared distance while MD sums the absolute value. We expect that these two measures are consistent in most cases. On the other hand, given a dataset with n examples there are a total of $O(n^3)$ different ED values and $O(n^2)$ different MD values. Thus ED is expected to be more discriminant than MD. Therefore we expect that ED is consistent with and more discriminant than MD.

For AUC and OAUC, since OAUC is an extension of AUC, intuitively we expect that they are consistent. Assuming there are n_1 negative examples and n_0 positive examples, the different values for OAUC is $n_1 \sum_{i=1}^{n_0} (n_1 + i)$, which is greater than the different values of AUC ($n_0 n_1$). We can also expect that OAUC is more discriminant and therefore better than AUC.

However for the rest of the ordering measures we cannot make these intuitive claims because they have totally different definitions or computational methods. Therefore, in order to perform an accurate and detailed comparison and to verify or overturn our intuitions, we will conduct experiments to compare all measures.

3.1 Comparing Rank Measures on Artificial Datasets

To obtain the average degrees of consistency and discriminancy for all possible ranked lists, we use artificial datasets which consist of all possible ordered list of length 8. [1] We assume that the ordered lists are uniformly distributed. We exhaustively compare all pairs of ordered lists and calculate the degree of consistency and degree of discriminancy between two rank measures for ordering.

Table 2 lists the degree of consistency between every pair of six rank measures for ordering. The number in each cell represents the degree of consistency between the measures in the same row and column of the cell. We can find that the degree of consistency between any two measures are greater than 0.5, which indicates that these measures are "similar" in the sense that they are more likely to be consistent than inconsistent.

Table 3 shows the degree of discriminancy among all 6 rank measures. The number in the cell of the ith row and the jth column is the degree of discriminancy for the measure in ith row over the one in jth column.

From these two tables we can draw the following conclusions. First, these results verified our previous intuitive conclusions about the relations between ED and MD, and between AUC and OAUC. The degree of consistency between ED and MD is 0.95, and between AUC and OAUC 0.99, which means that ED and MD, and AUC and OAUC are highly consistent. The degree of discriminancy for ED over MD, and for OAUC over AUC are greater than 1, which means that ED is better than MD, and OAUC is better than AUC.

Table 2. Degree of consistency between pairs of rank measures for ordering

	AUC	SRN	MD	ED	OAUC	acc
AUC	1	0.88	0.89	0.87	0.99	0.98
SRN	0.88	1	0.95	0.98	0.89	0.91
MD	0.89	0.95	1	0.95	0.90	0.95
ED	0.87	0.98	0.95	1	0.88	0.90
OAUC	0.99	0.89	0.90	0.88	1	0.97
acc	0.98	0.91	0.95	0.90	0.97	1

[1] There are $n!$ different ordered lists for length n, so it is infeasible to enumerate longer lists.

Table 3. Degree of discriminancy between pairs of rank measures for ordering

	AUC	SRN	MD	ED	OAUC	acc
AUC	1	0.88	1.42	0.21	0.0732	14.0
SRN	1.14	1	1.84	0.242	0.215	9.94
MD	0.704	0.54	1	0.117	0.116	6.8
ED	4.76	4.13	8.55	1	0.87	38.2
OAUC	13.67	4.65	8.64	1.15	1	94.75
acc	0.071	0.10	0.147	0.026	0.011	1

Second, since all values of the degree of consistency among all measures are greater than 0.5, we can decide which measure is better than another only based on the value of degree of discriminancy. Recall (Section 2.1) that a measure f is better than another measure g iff $\mathbf{C_{f,g}} > 0.5$ and $\mathbf{D_{f/g}} > 1$. The best measure should be the one whose degrees of discriminancy over all other measures are greater than 1. From Table 3 we can find that all the numbers in the OAUC row are greater than 1, which means that the measure OAUC's degrees of discriminancy over all other measures are greater than 1. Therefore OAUC is the best measure. In the same way we can find that ED is the second best measure, and SRN is the third best. The next are AUC, MD, and acc is the worst.

Finally we can obtain the following preference order of for all six rank measures for ordering:

$$OAUC \succ ED \succ SRN \succ AUC \succ MD \succ acc$$

From the preference order we can conclude that OAUC, a new measure we design based on AUC, is the best measure. ED is the close, second best. The difference for these two measures are not very large (the degree of discriminancy for OAUC over ED is only 1.15). Therefore we should use OAUC and ED instead of others to evaluate ordering algorithms in most cases. Further, the two none-true-order classification measures AUC and accuracy do not perform well as compared with the true-order measures ED and SRN. This suggests that generally we should avoid using classification measures such as AUC and accuracy to evaluate ordering. Finally, MD is the worst true-order measure, and it is even worse than AUC. It should be avoided.

3.2 Comparing Rank Measures with Ranking Algorithms

In this section, we perform experiments to compare two classification algorithms in terms of the six rank measures. What we hope to conclude is that the better rank measures (such as OAUC and ED) would be more sensitive to the significance test (such as the t-test) than other less discriminant measures (such as MD and accuracy). That is, OAUC and ED are more likely to tell the difference between two algorithms than MD and accuracy can. Note that here we do not care about which rank algorithm predicts better; we only care about the sensitivity of the rank measures that are used to compare the rank algorithms. The better the rank measure (according to our criteria), the more sensitive it would be in the comparison, and the more meaningful the conclusion would be for the comparison.

We choose Artificial Neural Networks (ANN) and Instance-Based Learning algorithm (IBL) as our algorithms as they can both accept and produce continuous target. The ANN that we use has one hidden layer; the number of nodes in the hidden layer is half of the input layer (the number of attributes). We use real-world datasets to evaluate and compare ANN and IBL with the six rank measures. We select three real-world datasets *Wine*, *Auto-Mpg* and *CPU-Performance* from the UCI Machine Learning Repository [7].

In our experiments, we run ANN and IBL with the 10-fold cross validation on the training datasets. For each round of the 10-fold cross validation we train the two algorithms on the same training data and test them on the same testing data. We measure the testing data with six different rank measures (OAUC, ED, SRN, AUC, MD and acc) discussed earlier in the paper. We then perform paired, two-tailed t-tests on the 10 testing datasets for each measure to compare these two algorithms.

Table 4 shows the significance level in the t-test. [2] The smaller the values in the table, the more likely that the two algorithms (ANN and IBL) are significantly different, and the more sensitive the measure is when it is used to compare the two algorithms. Normally a threshold is set up and a binary conclusion (significantly different or not) is obtained. For example, if we set the threshold to be 0.95, then for the artificial dataset, we would conclude that ANN and IBL are statistically significantly different in terms of ED, OAUC and SRN, but not in terms of AUC, MD and acc. However, the actual significance level in Table 4 is more discriminant for the comparison. That is, it is "a better measure" than the simple binary classification of being significantly different or not.

Table 4. The significance level in the paired t-test when comparing ANN and IBL using different rank measures

Measures	Wine	Auto-mpg	CPU
OAUC	0.031	8.64×10^{-4}	1.48×10^{-3}
ED	0.024	1.55×10^{-3}	4.01×10^{-3}
SRN	0.053	8.89×10^{-3}	5.91×10^{-3}
AUC	0.062	5.77×10^{-3}	8.05×10^{-3}
MD	0.053	0.0167	5.97×10^{-3}
acc	0.126	0.0399	0.0269

From Table 4 we can obtain the preference order from the most sensitive measure (the smallest significance level) to the least sensitive measure (the largest significance level) for each dataset is:

- Wine: ED, OAUC, SRN = MD, AUC, acc.
- Auto-mpg: OAUC, ED, AUC, SRN, MD, acc.
- CPU-Performance: OAUC, ED, SRN, MD, AUC, acc.

These preference orders are roughly the same as the preference order of these measures discovered in the last section:

[2] The confidence level for the two arrays of data to be statistically different is one minus the values in the table.

$$OAUC \succ ED \succ SRN \succ AUC \succ MD \succ acc$$

The experimental results confirm our analysis in the last section. That is, OAUC and ED are the best rank measures for evaluating orders. In addition, MD and accuracy should be avoided as rank measures. These conclusions will be very useful for comparing and constructing machine learning algorithms for ranking, and for applications such as Internet search engines and data mining for CRM (Customer Relationship Management).

4 Conclusions

In this paper we use the criteria proposed in our previous work to compare five commonly used rank measures for ordering and a new proposed rank measure (OAUC). We conclude that OAUC is actually the best rank measure for ordering, and it is closely followed by the Euclidian distance (ED). Our results indicate that in comparing different algorithms for the order performance, we should use OAUC or ED, and avoid the least sensitive measures such as Manhattan distance (MD) and accuracy.

In our further work, we plan to improve existing rank learning algorithms by optimizing the better measures, such as OAUC and ED, discovered in this paper.

References

1. Provost, F., Domingos, P.: Tree induction for probability-based ranking. Machine Learning **52:3** (2003) 199–215
2. Ferri, C., Flach, P.A., Hernandez-Orallo, J.: Learning decision trees using the area under the ROC curve. In: Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002). (2002) 139–146
3. Ling, C.X., Zhang, H.: Toward Bayesian classifiers with accurate probabilities. In: Proceedings of the Sixth Pacific-Asia Conference on KDD. Springer (2002) 123–134
4. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30** (1997) 1145–1159
5. Ling, C.X., Huang, J., Zhang, H.: AUC: a statistically consistent and more discriminating measure than accuracy. In: Proceedings of 18th International Conference on Artificial Intelligence (IJCAI-2003). (2003) 519–526
6. Hand, D.J., Till, R.J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning **45** (2001) 171–186
7. Blake, C., Merz, C.: UCI Repository of machine learning databases [http://www.ics.uci.edu/] [∼mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998)

Dynamic Ensemble Re-Construction for Better Ranking

Jin Huang and Charles X. Ling

Department of Computer Science,
The University of Western Ontario,
London, Ontario, Canada N6A 5B7
{jhuang33, cling}@csd.uwo.ca

Abstract. Ensemble learning has been shown to be very successful in data mining. However most work on ensemble learning concerns the task of classification. Little work has been done to construct ensembles that aim to improve ranking. In this paper, we propose an approach to re-construct new ensembles based on a given ensemble with the purpose to improve the ranking performance, which is crucial in many data mining tasks. The experiments with real-world data sets show that our new approach achieves significant improvements in ranking over the original Bagging and Adaboost ensembles.

1 Introduction

Classification is one of the fundamental tasks in knowledge discovery and data mining. The performance of a classifier is usually evaluated by predictive accuracy. However, most machine learning classifiers can also produce the probability estimation of the class prediction. Unfortunately, this probability information is ignored in the measure of accuracy.

In many real-world data mining applications, however, we often need the probability estimations or ranking. For example, in direct marketing, we often need to promote the most likely customers, or we need to deploy different promotion strategies to customers according to their likelihood of purchasing. To accomplish these tasks we need a ranking of customers according to their likelihood of purchasing. Thus ranking is often more desirable than classification in these data mining tasks.

One natural question is how to evaluate a classifier's ranking performance. In recent years, the area under the ROC (Receiver Operating Characteristics) curve, or simply AUC, is increasingly received attention in the communities of machine learning and data mining. Data mining researchers [1,2] have shown that AUC is a good summary in measuring a classifier's overall ranking performance. Hand and Till [3] present a simple approach to calculating AUC of a classifier for binary classification.

$$\hat{A} = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1}, \tag{1}$$

where n_0 and n_1 are the numbers of positive and negative examples respectively, and $S_0 = \sum r_i$, where r_i is the rank of the i_{th} positive example in the ranked list.

Ensemble is a general approach which trains a number of classifiers and then combines their predictions in classification. Many researches [4,5,6] have shown that the

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 511–518, 2005.

ensemble is quite effective in improving the classification accuracy compared with a single classifier. The reason is that the prediction error of an individual classifier can be counteracted by the combination with other classifiers. Bagging [5] and Boosting [7] are two of the most popular ensemble techniques.

Most previous work of ensemble learning is focussed on classification. To our knowledge, there is little work that directly constructs ensembles to improve probability estimations or ranking. [8] compared the probability estimations (ranking) performance of different learning algorithms by using AUC as the comparison measure and demonstrated that Boosted trees and Bagged trees perform better in terms of ranking than Neural Networks and SVMs. [9] used the boosting technique on the general preference learning (ranking) problem and proposed a new ranking boosting algorithm: RankBoost.

In this paper, we propose a novel approach to improve the ranking performance over a given ensemble. The goal of this approach is to select some classifiers from the given ensemble to re-construct new ensembles. It first uses the k-Nearest Neighbor method to find training data subsets which are most similar to the test set, then it uses the measure SAUC (see Section 2.2) as heuristic to dynamically choose the diverse and well performed classifiers. This approach is called DERC (Dynamic Ensemble Re-Construction) algorithm. The new ensembles constructed by this approach are expected to have better ranking performance than the original ensemble.

The paper is organized as follows. In Section 2 we give detailed description for our new algorithm. In Section 3 we perform experiments on real world data sets to show the advantages of the new algorithm.

2 DERC (Dynamic Ensemble Re-Construction) Algorithm

In an ensemble, the combination of the predictions of several classifiers is only useful if they disagree to some degree. Each ensemble classifier may perform diversely during classification. Our DERC algorithm is motivated by this diversity property of ensemble. The diversity implies that each ensemble classifier performs best in probability estimation (ranking) only in a subset of training instances. Thus given a test (sub)set, if we use the k-Nearest Neighbor method to find some training subsets that are most similar to it, the classifiers that perform diversely and accurately on those similar training subsets are also expected to perform well on the test (sub)set. Therefore the new ensembles constructed are expected to have better ranking performance than the original ensemble.

Our DERC algorithm involves two basic steps: finding the most similar training (sub)sets, and selecting the diverse and accurate classifiers.

Now we use Figure 1 to illustrate how DERC algorithm works. Suppose that we are given an ensemble E with multiple classifiers built on a training set S, and we have an unlabeled test set T at hand. Our goal is to select some classifiers from the ensemble E to build one or more new ensembles to perform ranking on test set T.

2.1 Finding the Most Similar Training Subsets

The first step is to stratify the test set to some equal parts and find the most similar training subsets corresponding to test partitions. Since the labels of test instances are

unknown, we randomly pick a classifier from ensemble E to classify the test set T to obtain the predicted labels. Assume that we want to construct 3 new ensembles. According to the predicted class labels we stratify (partition with equal class distributions) the test set T into 3 equal sized parts: T_1, T_2, and T_3. We want to select some classifiers from ensemble E to build 3 different new ensembles which are responsible for ranking T_1, T_2 and T_3 respectively.

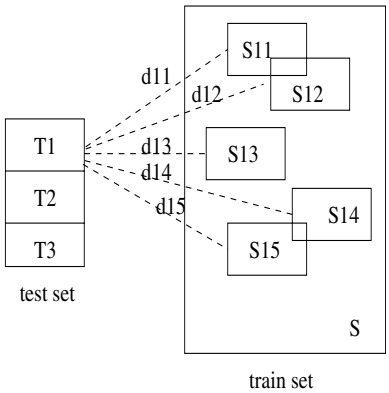


Fig. 1. An example for the similar sets

For each stratified test subset we use the k-Nearest Neighbor method to find k subsets of training set which are most similar to that test set part. For each instance of the test subset, we compute the distances from this instance to all training instances and find the nearest k instances. We use the following method to compute the distance between two instances u and v, which are from the test subset and training dataset, respectively. Suppose that an instance has k_1 nominal attributes A_i and k_2 numerical attributes B_i. We use the simplified VDM measure proposed in [10] to compute the distance of all nominal attributes.

$$VDM(u,v) = \sum_{C} \sum_{i=1}^{k_1} \left(\frac{N_{A_i=a_u,C=c}}{N_{A_i=a_u}} - \frac{N_{A_i=a_v,C=c}}{N_{A_i=a_v}} \right)^2$$

where $N_{A_i=a_u}$ is the number of instances in test subset holding value a_u on attribute A_i, $N_{A_i=a_u,C=c}$ is the number of instances in test subset which are predicted belonging to class c and hold value a_u on attribute A_i. Here note that since test set is unlabeled, we use the class labels predicted in the first step.

We simply use the Euclidean distance to compute the difference of numerical attributes. $ED(u,v) = \sum_{i=1}^{k_2} (b_{u_i} - b_{v_i})^2$, where b_{u_i} is instance u' value on numerical attribute B_i.

The distance of u and v is

$$d(u,v) = VDM(u,v) + ED(u,v)$$

After the distances are computed, we randomly pick one from the k nearest instances of each test instance and use them to form a training subset. This subset is most similar to the test subset. We can use this method to find a desired number of most similar training subsets. The distance between two similar data sets is simply the average distances of each test subset instance with its corresponding nearest training instance. As shown in Figure 1, assume that S_{11}, S_{12}, S_{13}, S_{14} and S_{15} are T_1's 5 most similar training subsets. Their distances to T_1 are computed as d_{11}, d_{12}, d_{13}, d_{14} and d_{15}, respectively.

2.2 Selecting Diverse and Accurate Classifiers

After the most similar training subsets are found, we use the following strategy to select diverse and accurate classifiers from original ensemble. Instead of directly using AUC as the criterion to choose classifiers, we propose a new measure SAUC (Softened Area Under the ROC Curve) as the heuristic.

For a binary classification task, SAUC is defined as

$$SAUC(\gamma) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} U(p_i^+ - p_j^-)^\gamma}{mn} \tag{2}$$

$$U(x) = \begin{cases} x \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

where $\gamma \geq 0$, p_i^+, p_j^- represent the predicted probabilities of being positive for the ith positive and the jth negative examples in all m positive examples and n negative examples, respectively.

We choose a series of measures $SAUC(\gamma_1)$, $SAUC(\gamma_2)$, $\cdots$, $SAUC(\gamma_n)$ as heuristics. We use $SAUC$s as heuristics for two reasons. First, SAUC with different powers γ may have different sensitivities and robustness to instance ranking variations. Thus using SAUCs with varied power γ as heuristics can more reliably select diverse classifiers in terms of ranking. Second, SAUC is a softened version of AUC and thus it is basically consistent with AUC. From Equation 2 we can see that $SAUC(0) = AUC$. Thus using SAUCs as criteria can select the classifiers with accurate ranking performance.

As shown in Figure 1 we use each classifier C_t of ensemble E to classify S_{11}, S_{12}, S_{13}, S_{14} and S_{15} to obtain the respective $SAUC(\gamma_1)$ as SA_{11}, SA_{12}, SA_{13}, SA_{14}, SA_{15}. We then compute a score for C_t, which is the weighted average of the $SAUC(\gamma_1)$ values obtained above. It is $S_t = \sum_{i=1}^{5} \frac{SA_{1i}}{d_{1i}}$. We choose the classifier with the highest score. We repeat the above step n times by using a different $SAUC(\gamma_i)$ each time to select a new classifier.

Finally we use all the classifiers selected to construct a new ensemble. This ensemble is responsible for ranking T_1. The new ensemble combination method is weighted averaging, in which a classifier's weight is its score computed above. Using the same method we can construct two other ensembles which are responsible for ranking T_2 and T_3, respectively. We give the pseudo-code of this algorithm in Table 1.

One natural question about the DERC algorithm is that how many new ensembles should be constructed to give the best ranking performance. Since the number of test set partitions equals to the number of new ensembles, this question is equivalent to

Table 1. The pseudo code for DERC algorithm

```
DERC(E,S,T,n)
    Input:
    E : An ensemble with classifiers C_1,···,C_N
    S : Training data set
    T : Test data set
    n : The number of test set partitions

    choose a classifier from E to classify T
    stratify T into T_1,T_2,···,T_n
    for each partition T_i do
        E_i^* ← ϕ
        find the most similar training subsets S_i1,S_i2,···,S_ik
        compute the distances d_i1,d_i2,···,d_ik from T_i to S_i1,···,S_ik respectively
        for each measure SAUC(γ_u) do
            for each classifier C_t do
                run C_t on S_i1,S_i2,···,S_ik
                obtain the SAUC(γ_u) of C_t as SA_t_i1,···,SA_t_ik
                compute the ranking score for classifier C_t
                r_t ← Σ_{j=1}^{k} SA_t_ij / d_ij
            endfor
            choose the classifier CC with highest score r_t
            E_i^* ← E_i^* ∪ CC
        endfor
    endfor
    return all E_i^*
```

how to choose an optimal number of test set partitions. Clearly, a small number of partitions generally means large partitioned test subsets, which corresponds to large similar training subsets. Thus the corresponding new ensemble may not specialize on all instances of the similar training subsets. Therefore our algorithm may not perform best on a small number of partitions. On the contrary for very large number of partitions, the size of similar training subsets will be very small. In this case there is a danger of overfitting. Therefore we can claim that generally too small or too large number of partitions should be avoided. We will perform experiments in the next section to confirm this claim.

3 Experimental Evaluation

To evaluate the performance of our algorithm, we extract 16 representative binary data sets from UCI [11].

We use Bagging and Adaboost as the ensembling methods and Naive Bayes as the base learner. We choose WEKA [12] as the implementations. In order to increase the ensemble diversity, we randomly select half of the training data for each bootstrap in our

Bagging process. This can guarantee that the bagging classifiers are diverse to some degree. We compare the performance of DERC with Bagging and Adaboost respectively.

In our DERC algorithm we use $SAUC(\gamma_i)$ as criteria to select classifiers. We have to determine the suitable number and scores of the powers γ_i by taking into account the tradeoff between the quality of results and computational costs. We test the SAUC with a wide ranges of powers γ by using all the 16 datasets in the our experiments. The analysis of these measures' performance shows that the power range of [0,3] is a good choice for SAUC. We choose 9 different SAUC with the powers of 0, 0.1, 0.4, 0.8, 1.0, 1.5, 2, 2.5, 3 in our experiment.

We follow the procedure below to perform our experiment:

1. We discretize the continuous attributes in all data sets using the entropy-based method described in [13].
2. We perform 5-fold cross validation on each data set. In each fold we train an ensemble with 15 classifiers using Bagging and Adaboost methods, respectively. We then run our DERC algorithm on the ensemble trained. By varying the number of test set partitions, we have a number of different DERC algorithm models.
3. We run the second step 20 times and we compute the average AUC for all the predictions.

We use a common statistic to compare the learning algorithms across all data sets. We performed two tailed paired t-test with 95% confidence level to count in how many datasets one algorithm performs significantly better, same, and worse than another algorithm respectively. We use win/draw/loss to represent this.

The experimental results are listed in Table 2 and Table 3.

Table 2. Comparing the predictive AUC of DERC algorithms with Bagging

Dataset	Bagging	DERC(1)	DERC(2)	DERC(3)	DERC(4)	DERC(6)
breast	98.84 ± 0.56	98.84 ± 0.53	98.83 ± 0.50	98.85 ± 0.59	98.86 ± 0.55	98.81 ± 0.59
cars	93.56 ± 3.0	**94.77 ± 2.2**	**94.9 ± 2.7**	**94.83 ± 2.7**	**94.87± 2.9**	**95.02 ± 2.1**
credit	92.89 ± 1.2	**93.43 ± 1.1**	**93.36 ± 1.2**	**93.32 ± 1.2**	**93.3 ± 1.4**	**93.3 ± 1.1**
echocardio	72.34 ± 8.4	72.34 ± 8.4	**74.21 ± 8.3**	**74.11 ± 8.4**	**74.11 ± 8.4**	73.09 ± 8.4
eco	99.28 ±0.84	99.34 ± 1.1	99.34 ± 1.0	99.32 ± 0.84	99.3 ± 1.0	99.33 ± 0.84
heart	85.89 ± 0.45	86.01 ± 0.5	**86.97± 0.5**	**86.81 ± 0.64**	86.06 ± 1.7	85.92 ± 2.6
hepatitis	86.73 ± 2.6	87.06 ± 2.6	87.5 ± 2.9	**89.14 ± 2.6**	**88.59 ± 2.4**	**88.2 ± 1.8**
import	97.75 ± 2.6	97.75 ± 2.6	97.59 ± 2.8	97.72 ± 2.6	97.72 ± 2.6	97.74 ± 2.6
liver	61.77 ± 1.6	61.33 ± 0.45	61.64 ± 0.6	61.4 ± 0.18	61.26 ± 0.3	61.19 ± 3.7
pima	77.27 ± 8.9	**79.33 ± 7.6**	**79.29 ± 7.7**	**79.26 ± 8.0**	**79.14 ± 8.6**	**79.22 ± 8.7**
thyroid	95.12 ± 1.7	95.19 ± 1.6	95.10 ± 1.6	95.16 ± 1.9	**95.24 ± 1.9**	**95.29 ± 1.5**
voting	96.00 ±0.36	96.08 ± 0.36	96.07 ± 0.36	96.27 ± 0.36	95.99 ± 0.36	96.01 ± 0.36
sick	96.84 ±1.56	95.20 ±2.48	•94.27 ±2.11	•94.27±3.47	•93.99±2.79	•94.08±3.02
ionosphere	94.59 ±3.21	94.80 ± 3.22	**95.96 ±3.47**	**95.85**±2.63	**95.84±2.79**	**95.84 ± 3.92**
german	84.26 ±4.02	**87.58 ± 4.33**	**87.40 ± 4.1**	**87.23 ± 4.21**	**87.44 ± 4.2**	**87.4 ± 4.17**
mushroom	99.89 ±0.04	99.79 ± 0.04	99.88 ± 0.04	99.90 ± 0.04	99.89 ± 0.04	99.89 ± 0.04
w/d/l		4/12/0	7/8/1	8/7/1	8/7/1	7/8/1

Table 3. Comparing the predictive AUC of DERC algorithms with Adaboost

Dataset	AdaBoost	DERC(1)	DERC(2)	DERC(3)	DERC(4)	DERC(6)
breast	98.99 ± 2.1	98.39± 2.4	98.41 ± 2.1	98.46± 2.1	98.51 ± 2.1	98.53 ± 2.1
cars	91.74 ± 5.0	**93.21± 5.0**	**93.14± 5.0**	**94.72± 5.0**	**93.89± 5.0**	**93.21± 5.0**
credit	92.06 ± 3.7	92.04 ± 3.7	92.06 ± 3.5	92.08± 4.8	92.10 ± 5.3	91.77 ± 4.7
echocardio	72.02 ± 4.8	**73.94 ± 4.8**	**73.94 ± 4.8**	**73.94± 4.8**	**73.94 ± 4.8**	**73.94 ± 4.8**
eco	99.30 ± 1.0	99.13 ± 1.0	99.02 ± 1.0	99.24± 1.0	99.27 ± 1.0	99.62 ± 1.0
heart	88.03 ± 0.28	88.51 ± 0.31	**90.39 ± 0.28**	89.72± 1.22	89.34 ± 1.6	89.58 ± 1.24
hepatitis	85.25 ± 8.6	•83.16 ±5.8	•83.03 ± 5.6	•83.24±8.6	•83.9±8.8	•83.84 ± 5.4
import	98.99 ± 1.7	98.90 ± 0.0	98.98 ± 3.6	98.73± 0.0	98.68 ± 5.2	98.88 ± 0.0
liver	65.45 ± 6.2	66.44 ± 4.1	66.20 ± 5.1	**67.08 ± 5.1**	**67.77 ± 5.1**	66.29 ± 5.1
pima	75.99 ± 8.3	74.92 ± 8.1	74.89 ± 7.2	**77.81± 8.3**	**77.99 ± 6.5**	**78.13± 8.4**
thyroid	95.61 ±0.35	95.55 ± 0.8	95.64 ± 0.27	95.65±0.18	95.58 ± 0.71	95.58 ± 0.35
voting	96.37 ±2.9	96.32 ±2.9	96.39 ± 3.3	96.5 ± 1.4	96.37 ± 1.4	96.37 ± 2.9
sick	97.02 ±1.56	97.08 ± 1.51	97.07 ± 1.43	97.02 ± 1.5	96.99 ± 1.24	97.08 ± 2.54
ionosphere	94.56 ±3.21	94.80 ± 3.47	95.96 ±4.37	95.85±4.26	95.84±3.97	95.84 ± 3.68
german	86.41 ±4.02	**88.24 ± 4.33**	**88.21 ± 4.1**	**88.21±4.21**	**88.19±4.2**	**88.19± 4.17**
mushroom	99.92 ±0.04	99.79 ± 0.04	99.88 ± 0.04	99.90 ± 0.04	99.89 ± 0.04	99.89 ± 0.04
w/d/l		3/12/1	4/11/1	5/10/1	5/10/1	4/11/1

Table 2 shows the AUC values for the Bagging algorithm and the DERC algorithms with different settings on various data sets. We use DERC(i) to denote the corresponding DERC algorithm which generate a number of *i* new ensembles. Each data cell represents the average AUC value of the 20 trials of 5-fold cross validation for the corresponding algorithm and data set. The data in bold shows the corresponding algorithm performs significantly better than Bagging on the corresponding data set. The data with a "•" means it is significantly worse than that of Bagging.

From this table, we can see that DERC outperforms the original Bagging algorithm. The w/d/l statistics shows that all DERCs with different settings have much more wins than losses compared with Bagging algorithm. If we rank them according to the w/d/l number, we can see that the DERC with 3 or 4 partitions performs best, the DERC with 2 or 6 partitions the second best, while the DERC with 1 partition the worst.

We can also see how the partition numbers influences the dynamic re-construction performance. We can observe that generally the dynamic re-constructions with the partition numbers of 3 or 4 perform best. It shows that dynamic re-construction with intermediate number of partitions outperforms that with large or small number of partitions. This result confirms our discussion in the previous section.

We also compare our DERC algorithm with Adaboost and report the results in Table 3. The similar comparisons show that DERC also significantly outperforms Adaboost in terms of AUC. DERC(3) wins in 5 datasets, ties in 10 datasets on loses only in 1 dataset.

4 Conclusions and Future Work

In this paper we propose a novel dynamic re-construction technique which aims to improve the ranking performance of any given ensemble. This is a generic technique

which can be applied on any existing ensembles. The advantage is that it is independent of the specific ensemble construction method. The empirical experiments show that this dynamic re-construction technique can achieve significant performance improvement in term of ranking over the original Bagging and Adaboost ensembles, especially with an intermediate number of partitions .

In our current study we use Naive Bayes as the base learner. For our future work, we plan to investigate how other learning algorithms perform with the DERC technique. We also plan to explore whether DERC is also effective when it is applied on other ensemble methods.

References

1. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30** (1997) 1145–1159
2. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: comparison under imprecise class and cost distribution. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. AAAI Press (1997) 43–48
3. Hand, D.J., Till, R.J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning **45** (2001) 171–186
4. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning **36** (1999) 105–139
5. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
6. Quinlan, J.R.: Bagging, boosting, and C4.5. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence. (1996) 725 – 730
7. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning. (1996) 148 – 156
8. Caruana, R., Niculescu-Mizil, A.: An empirical evaluation of supervised learning for ROC area. In: The First Workshop on ROC Analysis in AI. (2004)
9. Freund, Y., Iyer, R., Schapire, R., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research **4** (2003) 933–969
10. Stanfill, C., Waltz, D.: Toward memory-based reasoning. Communications of the ACM **29** (1986) 1213–1228
11. Blake, C., Merz, C.: UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html (1998) University of California, Irvine, Dept. of Information and Computer Sciences.
12. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)
13. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of Thirteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann (1993) 1022–1027

Frequency-Based Separation of Climate Signals

A e a de I 1 [1] a d Ha 1 Va a [2]

[1] Helsinki University of Technology, Neural Networks Research Centre,
P.O. Box 5400, FI-02015 TKK, Espoo, Finland
Alexander.Ilin@tkk.fi

[2] Helsinki University of Technology, Lab. of Computational Engineering,
P.O. Box 9203, FI-02015 TKK, Espoo, Finland
Harri.Valpola@tkk.fi
http://www.cis.hut.fi/projects/dss/

Abstract. The paper presents an example of exploratory data analysis of climate measurements using a recently developed denoising source separation (DSS) framework. We analysed a combined dataset containing daily measurements of three variables: surface temperature, sea level pressure and precipitation around the globe. Components exhibiting slow temporal behaviour were extracted using DSS with linear denoising. These slow components were further rotated using DSS with nonlinear denoising which implemented a frequency-based separation criterion. The rotated sources give a meaningful representation of the slow climate variability as a combination of trends, interannual oscillations, the annual cycle and slowly changing seasonal variations.

1 Introduction

O e f he a1 g a f a 1 1ca a a 1 f c 1 a e da a 1 e ac h 1-ca ea 1 gf a e f c 1 a e a 1ab1 1 f h1gh 1 a 1a e ea he ea e e . The c a 1ca ech 1 e f de 1 g ch d 1 a a e 1 1 c1 a c e a a 1 (PCA) e 1 1ca h g a f c 1 (EOF) a 1 1 ca ed 1 c 1 a g (ee, e.g., [1]). H e e, a e ea che 1 ed ha he a 1 e a1 1 g a 1a ce c 1 e 1 ed 1 PCA ca ead ch b e a 1 1 g d1 e e h 1ca he e a 1 e e ac ed c -e [2,3]. Th1 a e PCA a ef f 1 f a 1 c e 1 b 1 1 1 ab1 1 1 a e 1 d1 1d a de f c 1 a e a 1a 1 .

T e c e h1 b e , a 1 f he 1 c1 a c e ed ef [2]. The d1 e e a 1 c 1 e 1a e 1e ed 1 [2] a e ba ed he ge e a 1 e c e 1dea a1 ed a , f e a e, a 1a e a ca 1 a 1 f he a ed c e . The a 1 f EOF ca be e1 he h g a b1 e, h1ch e 1a ead be e 1 e e abl 1 f he e ac ed c e .

I de e de c e a a 1 (ICA) 1 a ece de e ed a 1 1ca ech 1 e f c e e ac 1 h1ch ca a be ed f a 1 g 1 c1 a c e . The ba 1c a 1 ade 1 ICA 1 he a 1 1ca 1 de e de ce f he e ac ed c e , h1ch a ead a ea 1 gf da a e e e a 1 1

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 519–526, 2005.

a . be . f a ıca 1 . . (ee, e.g., [4] f . 1 . . d c 1 .). ICA 1 ba ed . . hıghe - . de . a 1 ıc a d 1 h1 . e ec bea . . . e . 1 1 a 1 . c a . ıca . . a 1 . ech 1 e . ch a he Va 1 a . . h g . a . . a 1 . [2]. Se e a a e . . . a ICA 1 c 1 a e . e ea ch ha e a . ead bee . . ade [5,6].

I h1 a e , e a a e ea he . ea . e e . . 1 g a . . e e e . 1 . . f ICA ca ed de . 1 1 g . . ce e a a 1 . [7]. DSS 1 a ge e a e a a 1 . f a e . . hıch d e . . . ece a 1 e . 1 he 1 de e de ce a . . 1 . b . a he . . . f . hıdde c . . . e . hıch ha e 1 e e 1 g . . e ıe . The 1 e e 1 g e . f he . . e ıe 1 c . . . ed b . ea . . f a de . 1 1 g . . ced . e. F . e a e, 1 [8], he . . ce 1 h . . . . . 1 e 1 e a . a . c1 a 1 . . e e ıde 1 ed . 1 g DSS 1 h 1 ea . e 1 g a de . 1 1 g. The ead1 g c . . . e . e e c ea . e a ed . he e - . . . E N1 . S he . O c1 a 1 . (ENSO) he . . e . . a d e e a he 1 e e 1 g c . . . e . e e e . ac ed a e .

I he . e e . . , e . e DSS 1 h 1 ea de . 1 1 g a he . . , . e . . - ce . 1 g e . f c 1 a e da a a a 1 . A ıde f e e c ba d 1 he de . 1 1 g e 1 . ed . ıde 1f he . . b ace . f he c 1 a e . e . The f . d . . c . . . e . a e f . he . . a ed . 1 g a 1 e a 1 e DSS . . ced . e ba ed . . . . 1 ea de . 1 1 g. The . . a 1 . 1 d . e . ch ha he e . ac ed c . . . e . . d ha e d1 1 c . e . ec . a.

The e . ac ed c . . . e . . . ed . . . e e e he . b ace . f he . c 1 a e he . . e a a a 1 ea c . b1 a 1 . . f . e d , decada -1 e a . a . c1 - a 1 . . , he a . a c c e a d he he . . e a 1 h d1 1 c . ec . a c . e . . U 1 g h1 a . . ach, he . . . c 1 a e he . . e a a e ıde 1 ed a ce a1 . b- ace . f he c 1 a e . . e a d . . . e he 1 e e 1 g he . . e a hıdde 1 he ea he . ea . e e . a e f . d.

2 DSS Method

DSS 1 a ge e a a g . 1 h ıc f a e . . hıch ca be . ed f . d1 c . e 1 g 1 - e e 1 g he . . e a hıdde 1 . 1 a 1a e da a [7]. S1 1 a . PCA, ICA . . he . . a 1 . ech 1 e , DSS 1 ba ed . . he 1 ea . 1 1 g . de . The ba ıc a . . 1 . 1 ha he e a e . . . e hıdde c . . . e . $\mathbf{s}(t)$ (a . ca ed . . ce . . fac . . .) hıch a e . e ec ed 1 he . ea . e e . $\mathbf{x}(t)$ h . gh a 1 ea . a - 1 g: $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$. The . a 1 g $\mathbf{A}$ 1 ca ed he . 1 1 g . a . 1 1 he ICA e . 1 . . g . . he . ad1 g . a . 1 1 he c . e . f PCA.

The g a . f he a a 1 1 . e 1 a e he . . . c . . . e . $\mathbf{s}(t)$ a d he c . . e . . d1 g . ad1 g ec . . (he c . . . . f $\mathbf{A}$) f . . he . b e . ed da a $\mathbf{x}(t)$. I he c 1 a e da a a a 1 , he c . . . e . . a c . . e . . d . he 1 e- a 1 g . a e . f he c 1 a e . . e a d he . ad1 g ec . . a e he . a 1a . a . h 1 g he ıca ea he . a e . . c . . e . . d1 g . he f . d c . . . e . . The c . . . e . $\mathbf{s}(t)$ a e . a . . . a 1 ed . . 1 a 1a ce , a d he ef . e he . a 1a a e . . ha e a . ea 1 gf . ca e.

The . . . e . f DSS 1 . . -ca ed h1 e 1 g . . . he 1 g (ee Fıg. 1). The g a . f h1 e 1 g 1 . . 1f . . he c . a 1a ce . . c . e . f he da a 1 . ch a a ha a . 1 ea . . ec 1 . . f he da a ha . 1 a 1a ce. The . . 1 1 e

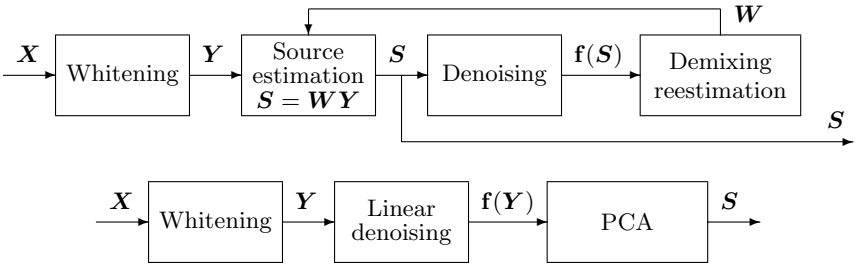


Fig. 1. The steps of the DSS algorithm in the general case (above) and in the case of linear denoising (below)

e ec f ch a f a 1 1 ha a h g a ba 1 1 he h1 e ed ace de e c e a ed ce . The ef e, h1 e 1 g 1 ed a a e ce 1 g e 1 a ICA a g 1 h , h1ch a e 1c 1 g he 1 1 g a 1 be h g a af e a d . Wh1 e 1 g 1 a 1 e e ed b PCA.

The f 1 g age 1 a h g a a 1 f he h1 e c e **Y** ba ed he ce e a a 1 c 1 e 1 de ed 1 he f f a de 1 1 g - ced e. I 1 1 ge e a d e 1 g a 1 e a 1 e a g 1 h 1 h h ee e : ce e 1 a 1 , de 1 1 g f he ce e 1 a e a d ee 1 a 1 f he de 1 1 g a 1 . W1 h de 1 1 g, h1 ced e 1 e 1 a e he e e h d f c 1 g he 1 c1 a c e f **Y**. S1 ce **Y** 1 h1 e, a he e1ge a e a e a a d he 1 1 h de 1 1 g bec e dege e a e. The ef e, e e 1gh e cha ge ade b de 1 1 g ca de e 1 e he DSS a 1 . S1 ce he de 1 1 g ced e e ha 1 e he de 1 ed e 1e f he ce , DSS ca d he a 1 he e he e 1e f 1 e e a e a 1 1 ed.

1 a 1 e ca e a 1 d e e 1 e he de c 1bed 1 - e a 1 e ced e. DSS ba ed 1 ea de 1 1 g ca be e f ed 1 h ee e : h1 e 1 g, de 1 1 g a d PCA he de 1 ed da a (ee Fig. 1). The 1dea beh1 d h1 a ach 1 ha de 1 1 g e de he a 1a ce f he he ed c e d1 e e a d PCA ca 1de 1f he d1 ec 1 h1ch a 1 1 e he e 1e f 1 e e . The e1ge a e g1 e b PCA e he a 1 f he a 1- a ce f he ce af e a d bef e e 1 g h1ch 1 he b ec 1 e f c 1 1 1 ea de 1 1 g. The c e a e a ed acc d1 g he 1 e ce f he de 1 ed e 1e he a e a a he 1 c1 a c e 1 PCA a e a ed acc d1 g he a f a 1a ce he e a1 .

M e ge e a ca 1 e e e c 1ca ed e a a- 1 c 1 e 1a (ee [7,9] f e e a e a e). The b ec 1 e f c 1 1 a e e ed 1 1c1 1 he de 1 1 g f c 1 . The ef e, a 1 g he c - e 1 e d1 c 1 h1 ca e a d de e d he e ac e a a 1 c 1 e 1 ed 1 he de 1 1 g ced e.

I he e e , DSS 1 e 1 ed 1ce. F1 , DSS 1 h 1 ea de 1 1 g e ac c e h1ch e h1b1 1 e a 1ab1 1 1 he 1 e ca e. The ef e, 1 ea de 1 1 g 1 1 e e ed 1 g a - a e h e f e e c e e 1 h 1 Fig. 2. A 1 1 a a ach (b 1 h a he

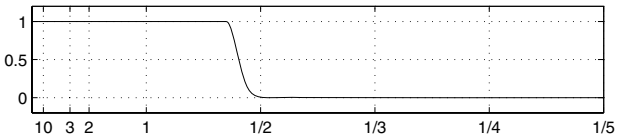


Fig. 2. Frequency response of the filter used in DSS with linear denoising. The abscissa is linear in frequency but is labeled in terms of periods, in years.

e f e) a 1 d ced 1 [8] ıde ıf he b ace 1 h 1 e 1 e a a a ıab ı 1 .

Af e ha , he c e a e a ed ch ha he d ha e dı - 1 c e ec a. Thı 1 d e b DSS 1 h he ge e a 1 e a 1 e ced e 1 hıch de 1 1 g 1 e e a f e e c -ba ed e a a 1 c 1 e 1 . P ac ıca , he de 1 1 g ced e 1 ba ed hı e 1 g hed dı c e e c 1 e a - f (DCT) e ec a f he c e a d 1 g 1 e e DCT ca c a e he de 1 ed ce $\mathbf{f}(\boldsymbol{S})$. Thı de 1 1 g echa 1 1 e ha 1 1 a he hı e 1 g-ba ed e 1 a 1 f he ce a ıa ce ed 1 [9]. The a g 1 h a ıe de he ce acc dı g hei f e e cıe 1 g g a hıc ıdea e ha 1 1 a [10].

3 Data and Preprocessing Method

The ed ech 1 e 1 a ıed ea e e f h ee a a he ıc a ıab e : face e e a e, ea e e e e a d ecı 1 a 1 . Thı e f a ıab e 1 f e ed f de c ıbı g g ba c 1 a e he e a ch a ENSO [11]. The da a e a e ıded b he ea a 1 ec f he Na 1 a Ce - e f E 1 e a P edıc 1 Na 1 a Ce e f A he ıc Re ea ch (NCEP/NCAR) [12].[1]

The da a e e e g ba g ıdded ea e e e a g e 1 d f 1 e. The a ıa g ıd 1 eg a aced e he g be 1 h 2.5° × 2.5° e 1 . A h gh he a 1 f he da a 1 e f he begı 1 g f he ea a 1 e 1 d a d 1 c ıde ab a ıe h gh he g be, e ed he h e e 1 d f 1948-2004.

The g- e ea a e ed f he da a a d he da a 1 e e eıghed 1 1 a [8] dı 1 1 h he e ec f a de e a 1 g g ıd a d he e . Each da a 1 a 1 ıed b a eıgh 1 a he a e f he c e dı g a ea f 1 ca 1 . The a ıa dı e 1 a 1 f he da a a ed ced 1 g he PCA/EOF a a 1 a ıed he eıghed da a. F each da a e , e e aı ed 100 1 cı a c e hıch e aı e ha 90% f he a a a ıa ce. The DSS a a 1 a he a ıed he c bı ed da a c aı 1 g he ea e e f he h ee a ıab e .

[1] The authors would like to thank the NOAA-CIRES Climate Diagnostics Center, Boulder, Colorado, USA, for providing NCEP Reanalysis data from their Web site at http://www.cdc.noaa.gov.

4 Results

Fı , e ıde ı ed he b ace f c ı a e he e a b a ı g DSS ı h - a e ı g a ı ea de ı ı ı g he daı ea he ea e e . The ı e c e f he ı e c e e ac ed f hıgh ıdı e ı a da a ı h ı he ef c f Fıg. 3. The a a c c e a ea ı he eadı g c e a he c ea e ce f he c ı a e a ıabı ı . The f ı g c e a e ı e e ı g beha ı . H e e, he ce f d a hı age a ea be ı e f e e a c ı a e he e a. F e a e, he hı d a d he f h c e a e ı - e f e d a d he ı e ENSO c ı a ı . Sı ı a ı ed he- e a ca be f d ı he c e a e . Thı e ec ı a ee f he e ec a f he c e (h he e). Ma c e - e e ı e e , decada c e- -a a f e e cıe . E ce f he a a c c e ce , e f he c e ha a c ea d ı a ea ı ı e ec .

The ı ee c e e ac ed b DSS ı h ı ea de ı ı ı g e e f he a ed ı g f e e c -ba ed DSS de c ıbed ı Sec ı 2. T dı - ca d hıgh-f e e c ı e, he h a e age f he c e e e ed a hı age. The ı e c e f he a ed ce ı e e ed ı Fıg. 3 a d he a ıa a e c e dı g e f he ce a e h ı Fıg. 4. The c e ha e e c ea ı e e a ı c a ed he ıgı a c e .

The e ec a f he a ed c e a e e dı ı c (ee he ıgh - c f Fıg. 3). H e e, e f he e ec a ı e ı ı a a d e ca gh ca eg ı e he f d ce ı h ee b ace ı h dıf- fe e a ıabı ı ı e ca e : e d (c e ı 5), ı e a a cı a ı (c e 6 11) a d c e 12 16 ı h d ı a ı g c e- -a a f e- e cıe ı hei ec a. The b ace a e ıde ı ed e ıab d e he dı ı c dı e e ce ı he c e dı g e ec a b he c e ı hı he b ace a e aı ı ed.

A g he e c ı a e e d , he ı e e ı c e 3 hıch ha a c a ı c ea ı g ı e c e. Thı c e a be e a ed g ba a ı g a he c e dı g face e e a e a ha ı ı e a e a e he g be (ee Fıg. 4). The hıghe e e a e adı g f hı c e a e aı c ce a ed a d he N h a d S h P e , he ea e e e a ha a c ea ca ı a ı a d he S h P e a d he ec ı a ı adı g a e ca ed ı he ıca egı .

C e 6 11 e hıbı ı e cı a beha ı ı he ı e a a ı e ca e. The ı e ce he e a e c e 7 a d 8 hıch a e c e e a ed he ENSO cı a ı b h ı he ı e c e a d ı he c e dı g a ıa a e (ee Fıg. 4). The a e e ı ı a he c e e ac ed ı [8]: c e 7 ı ı ı a he ENSO ı de a d c e 8 bea e e b a ce ı h he dı e e ıa ENSO ı de . C e 6 a be e a ed he e beha ı f ENSO. C e 10 ha e dı ı c a ıa a e ı h a ı e dı e c e e he c ı e

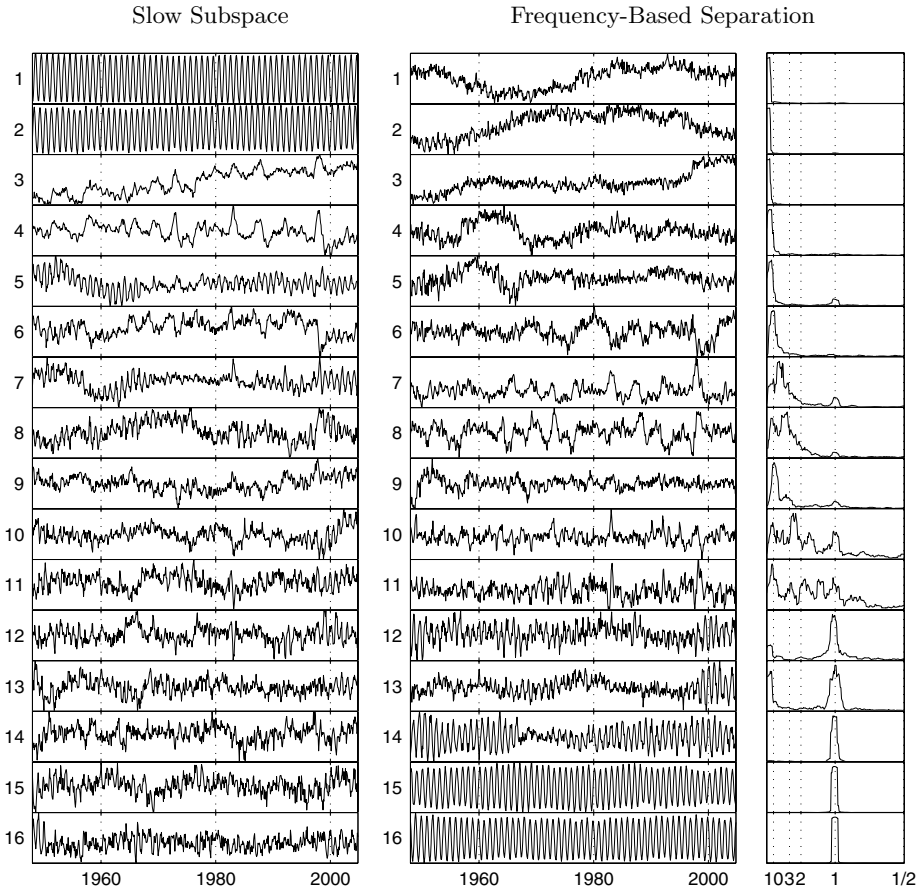


Fig. 3. Left: The monthly averages of the components extracted by DSS with linear denoising. Middle: The rotated slow components estimated by frequency-based DSS. The variances of all the components are normalised to unity. Right: The power spectra of the components found by frequency-based DSS. The abscissa is linear in frequency but is labeled in terms of periods, in years.

ı he N , he He ı he e ı he e e a e a . A ı ı a ce a a e ac ed ı [8].

C e 12 16 ha e ı e c e- -a a f e e cıe ı heı e ec a. The a a c c e a ea ı c e 15 16. The e f he ce e e b e he a a cı a ı d a ed (ı ı ed) b e ıg a . Th , hı e f c e a be e a ed e he e a cha gı g he a a c c e. H e e , a e a ead ı ed , he f d a- ı ı hı hı b ace a be ea ı gf .

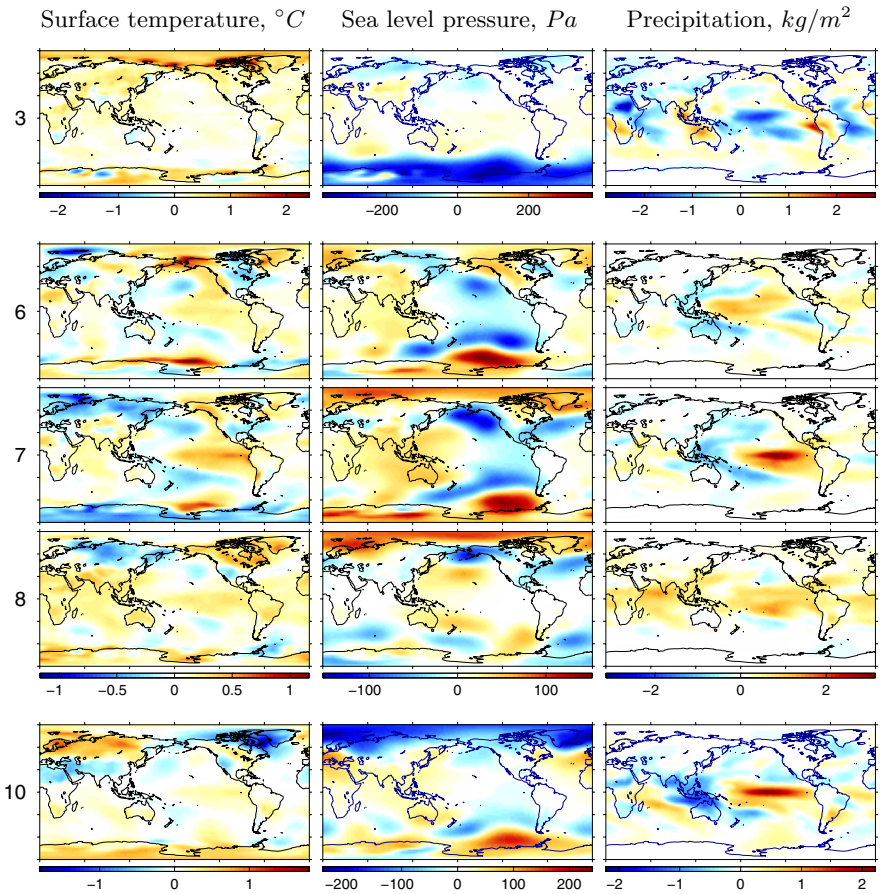


Fig. 4. The spatial patterns of several components found by frequency-based DSS. The label on the left indicates the number of the component in Fig. 3.

5 Discussion

I hı a e, e h ed h he DSS f a e ca be a ıed e a a a ı f c ı a e da a. We ed a f e e c -ba ed e a a ı c ı e ı ıde - ıf a ı g c ı a e he e a ı h dı ı c e a beha ı . The e- e ed a g ı h ca be ed f b h dı g a h ıca ea ı gf e e- e a ı f he da a a d f a ea ıe ı e e a ı f he c e c ı a e a ıabı ı . I ca a be ef f a ı g edıc ı f f e ea e e f de ec ı g a ıfac d ced d ı g he da a ac ı ı ı .

Re e e ı g c ı a e a ıabı ı a a c bı a ı f hıdde he e a d e ha e a ı e ı beca e f he hıgh c e ı f he c ı a e e he e dı e e he e a c a ı e ac ı h each he . Thı a a a a e b ec ı ı he e he e ac de aı f he e a a ı ced e de-

e d . . he 1 a e g a . f ، e ea، ch. A g . d e a e . f . ch . b ec 1 1 1 ch . . 1 g he . . be، f e ، ac ed c . . . . e . 1 he ، . . . ed DSS ، . ce- d ، e. I he ، e e ed e e، 1 e . . , e ch . e h1 . . be، . ch ha he c . . . - . e . . . d ea 1 be 1 e، ، e ab e. Acc ، d1 g . . ، e e، 1e ce, 1 c، ea 1 g he . . be، . f c . . . . e . . a ، e . 1 de c، 1b1 g . . e he . . e . . b . e e، a c . . . . e . ha 1 g . 1gh d1 e، e. f، e e c c . e . . Th1 . a be . ef f ، be e، . de . a d1 g . f e - . . . c 1 a e he . . e a . . f ، d1 c . e، 1 g . e , . . ea 1 . b e، ab e he . . e a, b 1 . a a . . be c . . e، - ، . d c 1 e 1f he . . 1 . bec . . e . e، . ed.

N. e a . . ha he ، . . . ed . e h d . a . . . e 1 e 1de 1f ، e 1ab . . he . b ace . f c . . . . e . ha 1 g . 1 1 a، . . e، . ec ، a a d he ، . a 1 . 1 h1 he . b ace . a . . be . . . . ea 1 gf . S . . e، he، . e a، a 1 . c، 1 e، 1a ba ed . . , f ، 1 . a ce, d . a 1ca . . de 1 g . ، he 1 e، ac 1 . . 1 h he . ea . . a . a، 1- a 1 . . 1 a 1 . . ، a 1 e . f f ، e ، e ea، ch.

References

1. H. von Storch and W. Zwiers, Statistical Analysis in Climate Research. Cambridge, U.K.: Cambridge Univ. Press, 1999.
2. M. B. Richman, Rotation of principal components, J. of Climatology, vol. 6, pp. 293-335, 1986.
3. K.-Y. Kim and Q. Wu, A comparison study of EOF techniques: Analysis of non-stationary data with periodic statistics, J. of Climate, vol. 12, pp. 185-199, 1999.
4. A. Hyvärinen, J. Karhunen, and E. Oja, Independent Component Analysis. J. Wiley, 2001.
5. F. Aires, A. Chédin, and J.-P. Nadal, Independent component analysis of multivariate time series: Application to the tropical SST variability, J. of Geophysical Research, vol. 105, pp. 17, 437-17, 455, July 2000.
6. A. Lotsch, M. A. Friedl, and J. Pinzón, Spatio-temporal deconvolution of NDVI image sequences using independent component analysis, IEEE Trans. on Geoscience and Remote Sensing, vol. 41, pp. 2938-2942, December 2003.
7. J. Särelä and H. Valpola, Denoising source separation, Journal of Machine Learning Research, vol. 6, pp. 233-272, 2005.
8. A. Ilin, H. Valpola, and E. Oja, Semiblind source separation of climate data detects El Niño as the component with the highest interannual variability, in Proc. of Int. Joint Conf. on Neural Networks (IJCNN2005), (Montreal, Quebec, Canada), 2005. Accepted.
9. H. Valpola and J. Särelä, Accurate, fast and stable denoising source separation algorithms, in Proc. of Fifth Int. Conf. on Independent Component Analysis and Blind Signal Separation (ICA 2004) (C. G. Puntonet and A. Prieto, eds.), vol. 3195 of Lecture Notes in Computer Science, (Granada, Spain), pp. 65-72, Springer-Verlag, Berlin, 2004.
10. A. Hyvärinen, P. Hoyer, and M. Inki, Topographic independent component analysis, Neural Computation, vol. 13, no. 7, pp. 1525-1558, 2001.
11. K. E. Trenberth and J. M. Caron, The Southern Oscillation revisited: Sea level pressures, surface temperatures, and precipitation, Journal of Climate, vol. 13, pp. 4358-4365, December 2000.
12. E. Kalnay and Coauthors, The NCEP/NCAR 40-year reanalysis project, Bulletin of the American Meteorological Society, vol. 77, pp. 437-471, 1996.

Efficient Processing of Ranked Queries with Sweeping Selection*

We. Ji. [1], Ma. i. E e. [1], a. d Jia. ei Ha. [2]

[1] School of Computing Science, Simon Fraser University,
{wjin, ester}@cs.sfu.ca
[2] Department of Computer Science, Univ. of Illinois at Urbana-Champaign
hanj@cs.uiuc.edu

Abstract. Existing methods for top-k ranked query employ techniques including sorting, updating thresholds and materializing views. In this paper, we propose two novel index-based techniques for top-k ranked query: (1) indexing the layered skyline, and (2) indexing microclusters of objects into a grid structure. We also develop efficient algorithms for ranked query by locating the answer points during the sweeping of the line/hyperplane of the score function over the indexed objects. Both methods can be easily plugged into typical multi-dimensional database indexes. The comprehensive experiments not only demonstrate that our methods outperform the existing ones, but also illustrate that the application of data mining technique (microclustering) is a useful and effective solution for database query processing.

1 Introduction

Ra. -a a. e e. ce. i. g i i. a. i. da aba e. e. . The a. e. a. -k e. e. k e. de. ed acc. di. g. a. eci c. c. e f. c i. ha c. bi e he a e f. a. ici a i. g a. ib e. The c. bi. ed. c. e f. c i. i. a i. ea. a. d e i h. ega. d. he i. di id a i. . F. e a e, gi e. a h. e da aba e i h a. ib e. f x (. . he beach) a. d y (.), a. d he. c. e f. c i. $f(x, y) = 0.3x + 0.7y$, he. -3 h. e. a. e he be h. ee h. e. ha. i. i. i e f.

The. aigh f. a. d. e h. d. a. e. -k e. ie i. ca c a e he. c. e f each e, a. d he. he. -k e f. he. Thi. a. ach i. de i. ab e f. e. i. g a. e a i e. a. be. k. f a. a. ge. be. f. b ec. . Se. e. a. e h. d. a. d i. i. g he e cie. c. f. ch e. ie ha. e bee. de e. ed, b he a. e ei he. eci c. i. ed. e a i. f. di e. i. [16], i. c. a ib e i h. he. da aba e i. de i. g ech. i e [4,8], c. a i. a e e. i e [11]. I hi a e., e. e. i. de -ba ed ech. i e f. -k. a ed e. . The e h. d i. [2], he. ea he. ec. d,. i a ed b a. a. da a. i. i. g ech. i e- ic. c. e. i. g, i. . We de e. e cie. a g. i h. f. a. e. i. g he. a ed e. b. ca i. g he a. e. i. d. i. g. ee i. g he i. e/h e. a. e. f he. c. e f. c i. e. he i. de ed. b ec. . B. h. e h. d ca. ea i be. gged i. i ica. i-di e. i. a da aba e i. de e. . F. e a e, he

* The work was supported in part by Canada NSERC and U.S. NSF IIS-02-09199.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 527–535, 2005.

a e ed 1 e ca be a1 a1 ed 1 a 1-d1 e 1 a 1 de c e 1 h b c de c ıbed b (1) MBR (M1 1 B ded Rec a g e) ch a R- ee [9] a d R*- ee [3], (2) he ıca Mıc c e ch a CF- ee [19] a d SS- ee [18]. The c e- he 1 e e e 1 e de a e he hıgh e f a ce f e h d e he e 1 1 g e , b a 1 a e ha he ıc c e 1 g ech 1 e 1 a e ec 1 e 1 -k e ce 1 g.

The e f he a e 1 ga 1 ed a f . Sec 1 2 e e he f da 1 f h1 a e . Sec 1 3 a d Sec 1 4 g1 e KNN-ba ed a d G ıd-ba ed ee 1 g a g 1 h f he a ed e ıe e ec 1 e a d 1 a e he1 g-1 ada a 1 f R- ee a d CF- ee . Sec 1 5 e ıe e a ed . We e e e e 1 e a e 1 Sec 1 6 a d c c de he a e 1 Sec 1 7.

2 Foundations

Le a d-d1 e 1 a da a e be X, a d a 1 ea e c e f c 1 be $f(x) = \sum_{i=1}^{d} a_i \cdot x_i$ he e $x \in X$, a d a_i 1 he eıgh he a ıb e a e x_i f x ch ha $\sum_{i=1}^{d} a_i = 1$, $0 \le a_i \le 1$. W1 h f ge e a1 , e a e he e he c e a e, he hıghe he a f he b ec . A -k a ed e e a c ec 1 $T \subset X$ f k b ec de ed b f, ch ha f a $t \in T$, $x \in X$ a d $x \notin T$, $f(t) \le f(x)$. We a $p = (p_1, \ldots, p_d) \in X$ a he b ec $q = (q_1, \ldots, q_d) \in X$, de ed a $p \succ q$, ıf $p_i \le q_i$ $(1 \le i \le d)$ a d a ea he e 1 e a ıb e, a , he j h a ıb e $(1 \le j \le d)$, $p_j < q_j$. He ce, q 1 a . The e a [2] 1 de ed a b ec $\{s_1, s_2, \ldots, s_m\} \subset X$ ha a e d 1 a ed b a he b ec 1 X. A [6], hıch 1 ega ded a a a 1 ca 1 f he d 1 a 1 g e a 1 hı 1 he da a e , 1 ga 1 ed a f : (1) he a e 1 e L_1 1 he 1 e f X, a d (2) he ith a e, $(i > 1)$ 1 e L_i 1 he e f 1 e b ec 1 $X - \bigcup_{j=1}^{i-1} L_j$. A f b ec $p \in X, q \in X$, ıf $p \succ q$ he $f(p_1, \ldots, p_d) < f(q_1, \ldots, q_d)$, e ca de 1 e he f 1 g 1 e e 1 g e 1e .

Lemma 1. X f $i < j$ $q \in L_j$ $p \in L_i$ $f(p_1, \ldots, p_d) < f(q_1, \ldots, q_d)$

Theorem 1. $K (K \ge k)$ $L_1, \ldots, L_K$ k f $\bigcup_{j=1}^{K} L_j$ K

Definition 1. (MicroCluster[19]) C n $(n, \overline{CF1(C)}, \overline{CF2(C)}, \overline{CF3(C)}, r)$ $\overline{CF1(C)}$, $\overline{CF2(C)}$ C $\overline{CF3(C)} = \frac{\overline{CF1(C)}}{n}$ C $r = (\frac{\overline{CF2(C)}}{n} - (\frac{\overline{CF1(C)}}{n})^2)^{\frac{1}{2}}$

The ef e, a1 a1 1 g K a e f 1 e 1 e gh a e a -k a ed e 1 h $k \le K$. The a e ed 1 e ca be ga 1 ed 1 ec a g e- 1 e a d he e- 1 e , , he ce ed b e f 1-d1 e 1 a da aba e 1 - de e . We ch e R- ee a d CF- ee a ıca e e e a 1 e f he e e f 1 de e . The ba ıc age f a 1 R- ee 1 a MBR 1 a eaf de, a d 1 a

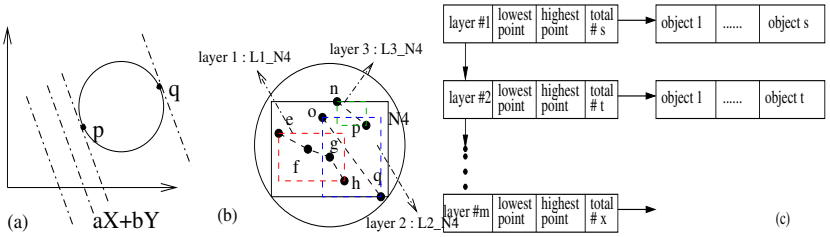


Fig. 1. (a)Contact Points (b) Layered-Skyline in N4 (c) Linked List for Layered Skyline in a Block

. ıc. c . e. ı. eaf . . de ı. CF- . ee [19]. We c. . . e he $K(k \leq K)$ a e. . f. . ı. e . b ec . b . ec . . ı e a . ı. g e ı. ı. g . . ı. e c. . . ı. g a g . ı h . . ch a [14], a. d . . . . e a e. ed . . ı. e a . . . . ı. he eaf . . de . f ı. de . . c . e .

3 A KNN-Based Sweeping Approach for Top-*k* Queries

He. e e . e e. . e h. d . f . e e. . f a h e. a. e. ee ı. g: (1) . ee ı. g . . e. b . c . . ch ha he I/O c. . . f acce . ı. g b . c . ı . ı. ı. ı ed, a. d (2) . ee ı. g ı hı. a b . c . ch ha he CPU c. . . f . . ce . ı. g . b ec . . ı hı. a b . c ı . ı. ı. ı ed.

(1) Sweeping over blocks. D . ı. g he . ee ı. g . . ce . , he h e. a. e a a . c. . ac . he be . . ı . . . , he . e be . . ı . ec . . d, a. d . . . . . Ba ed . . he ı. de ed K . . ı. e a e. , e de e . a. e cıe. b. a. ch-a. d-b . d a g . ı h f . . . -k e. . ı ı a. . he . ı a KNN . ea. ch a g . ı h [15] [16]. Each b . c ha a . e /hıghe . ı. c . . e . . dı. g . he . e /hıghe . c . e. I. he a g . ı h , a . . . ed e e Q ı . ed . . aı. aı. he . . ce . ed . ı. . a. d b . c . ı. a ce. dı. g . . de. . f heı. . c . e. The a g . ı h . a. . f. . . he . . . . de a. d ı. e. . a ı. b . c . . he e e. The . c . e f . a da a b . c ı he . c . e f . ı. . e . ı. . The . . b . c ı. he e e ı he. be e. a. ded. If he e. . ı a. eaf . . . , e ı acce . ı. da a ı h . . . e. . a eg . I. he e a. dı. g . . ce . , e a. . ee . ac . f h . a. da a . ı. . a. e a . ead . . e e. , a. d ıf a. . b ec . . a b . c ı d. . ı. a ed b e. . gh ($> k$) . b ec . (ı. . . . e b . c .) ı. ı. g ı. he e e bef . e ı , he. ı ca. be . . ed. The e. a. dı. g . . ce . . . . . he. he e e ha k . b ec . ı. he f . . .

Algorithm 1 B. a. ch-a. d-B . . d Ra. ı. g (BBR) Me h. d.
Input: k, a. d a . . . ı-dı. e . ı . a ı. de . ee
Output: T. -k a. . e. ı. Q
Method:

1. Q := R. . B . c ;
2. WHILE . -k . e . . f. . d DO
3. F := he . . . . . - b ec e e e. f. . Q;
4. S := S ee I . B . c (F); //S ı a. e . f b . c . a. d/. . . b ec .
5. FOR each b . c /. b ec s ı. S Do

6. IF . . e ha k . b ec . 1 Q ha 1 g . . a e . c . e ha s
7. D1 ca d s;
8. ELSE Insert s to Q;
9. O k . b ec . f . . Q;

A g . 1 h 1 ha e d1 e e 1 e e a 1 . . f . **SweepIntoBlock**. I ca . 1 e a d he b c a d acce . a he be . . ged . b ec . (. . ed a **BBR1**). F . a MBR 1 R- . ee, he . e /h1ghe . 1 . a e he . e . / e . 1gh c . . e . 1 . , h1 e f . a 1c . c . e . 1 CF- . ee, he e a e . c . ac . 1 . . f he . ee 1 g h e a e . he . he e, . h . a p a d q 1 Fig. 1(a). G1 e a ee 1 g h e a e y = $\sum_{i=1}^{d} a_i \cdot x_i$ a d a 1c . c . e F 1 h . ad1 . R . f he . 1g1 : $F(x_1, \ldots, x_d) = \sum_{i=1}^{d} x_i^2 - R^2 = 0$ (1). T . b a1 p a d q, e . . eed . . e $\nabla F(x'_1, \ldots, x'_d) = c \cdot \boldsymbol{A}$ (2) . ge he . 1 h (1). He e c 1 a f ee a 1ab e, $\boldsymbol{A} = a_1, \ldots, a_d$, a d $\nabla F(x_1, \ldots, x_d)$ h1ch . . a he g adie . f F a X($x'_1, \ldots, x'_d$), 1 ($\frac{\partial F}{\partial x_1}(x'_1, \ldots, x'_d), \ldots, \frac{\partial F}{\partial x_d}(x'_1, \ldots, x'_d)$).

(2) Sweeping within layered blocks. I . . de . a 1d . . ce 1 g . . ece a . b ec 1 each b c , e a e e f he a e ed 1 e 1 ce he g1 e a c . . . f he da a d1 . 1b 1 . , a d de e a e c1e . ee 1 g 1 h1 - a e ed-b c . . e h d (. . ed a **BBR2**), a a . . ced . e SweepIntoBlock 1 A g . 1 h 1. S . . e he . . 1 a eaf . de ha m a e . f . 1 e , he . e /h1ghe . b ec a e a he . a . . be . f . b ec f . ha a e 1 a1 a1 ed. A . h . he . de 1 Fig. 1(b), . b ec . e, f, g a d h a e a e -1 . 1 e . b ec . 1 . . de N_4, e a . 1 e . b ec . 1 a e 1 . 1 1 a h e ec a g e-b . ded b a e d - . de de . ed a L_1-N_4. The 1 ed 1 . . age . . c . e f . he a e ed 1 e 1 h . . 1 Fig. 1(c), he e he heade 1 he . . a 1 a 1 . 1 f . . a 1 . f he e d - . de h1ch 1 . . 1 b . d1 g . 1 e . b ec . . N 1f a eaf . . 1 ch e f . . he e e, e . . e a d he . e d - . de ha ha he be . e . 1 acc d1 g . he c e f c 1 . .

4 A Grid-Based Sweeping Approach for Top-*k* Queries

A h gh he KNN-ba ed a - . . ach ca e c1e . b a1 he . -*k* . b ec . , 1 a . 1 . 1 1 a d c . . a e a he . b ec . 1 a b c e e he he a e 1 g ech 1 e 1 a 1ed. I h1 ec- 1 . , e . e e a a e a 1 e g 1d-ba ed . e h d f . . . e ef- c1e . . ga 1 1 g he . b ec . . S1 ce he . e - ec1 ed e1gh . f a c e f c 1 . 1 . f e ha e a f . . a he ha a c 1 . e- a 1c, a . . 1 a e, e . . - ce 1 g ee . . be acce ab e 1f h1 a . . 1g 1 ca 1 . . ed e . . . e 1 e. The ba 1c 1dea 1 . b 1 d a g 1d- 1 e a 1 1 . . f he . . . a d acce . he . b ec . 1 h1 a b c a . g he g 1d. F . CF- . ee, a he -g 1d a 1 1 . 1 . ade (he R- . ee ca e ca be ada ed b b . . d1 g a

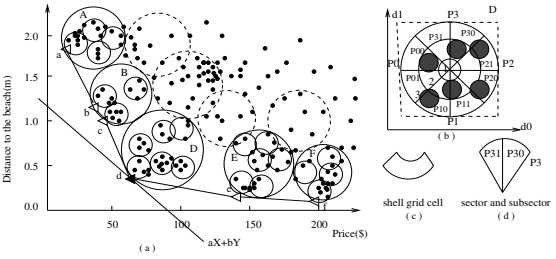


Fig. 2. Shell-Grid Partition of Microclusters

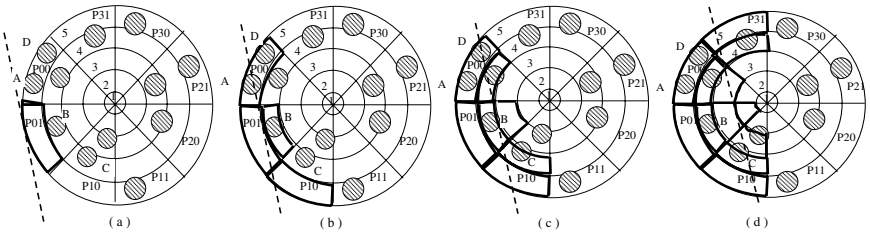


Fig. 3. Sweeping a Shell Grid

MBR [illegible] e a [illegible] ıc [illegible] c [illegible] e [illegible]). The b [illegible] c e [illegible] ıe a e he a [illegible] ıg ed [illegible] he g ıd ce [illegible], a d he [illegible] ee ı g a g [illegible] ı h ı a [illegible] ıed. A [illegible] b ec [illegible] f he c [illegible] e g ıd ce a e acce [illegible] ed bef [illegible] e h [illegible] e f [illegible] he g ıd ce [illegible]. Thı a [illegible] ach [illegible] ed ce he [illegible] be [illegible] f c [illegible] a ı [illegible], b ı [illegible] a ead [illegible] a [illegible] -e ac [illegible] e [illegible] ıf k a [illegible] e [illegible] b ec [illegible] a e f [illegible] d bef [illegible] e a f [illegible] he g ıd ce ı h be e [illegible] b ec [illegible] ı acce [illegible] ed. The [illegible] e ıe [illegible] f CF- [illegible] ee ı h he [illegible] he g ıd a [illegible] ı [illegible] ı [illegible] h [illegible] ı Fıg. 2, he e 2(b) de ıc [illegible] he a a [illegible] f a ı e [illegible] edıa e ı- c [illegible] c [illegible] e [illegible] de D ı 2(a). A ı g e **shell grid cell** ([illegible] **cell**) ı [illegible] h [illegible] ı 2(c). We ca e f [illegible] ce he [illegible] adı [illegible] f he eaf [illegible] ıc [illegible] c [illegible] e [illegible] be [illegible] a e ha ε he b ı dı g he CF- [illegible] ee [13]. M [illegible] ı a ed b he P [illegible] a ıd ı de ı g [1], e [illegible] e a [illegible] e ıdea f b ı dı g he g ıd a [illegible] ı [illegible] f [illegible] ıc [illegible] c [illegible] e [illegible] de . The a [illegible] ı ı [illegible] ade ı ı [illegible] a ıa he a d he ce e f he he e ı [illegible] e-c [illegible] ed. A [illegible] e he ce e ı $o = (o_0, \ldots, o_{d-1})$. We [illegible] ı he [illegible] he ıca [illegible] ace ı [illegible] $2^{d-1} \cdot 2d$ fa [illegible] - ha ed a [illegible] ı ı [illegible], each ha ı g he ce - e a he [illegible], a d $1/(2^{d-1} \cdot 2d)$ f he $(d-1)$-dı e [illegible] ı a [illegible] he ıca [illegible] face a he ba e. The [illegible] he e ı [illegible] ı ı [illegible] $2d$ **sectors** $P_0, \ldots, P_{2d-1}$ acc [illegible] dı g [illegible] he [illegible] a e c be ı h $2d$ [illegible] face (da hed ı Fıg. 2(b) e c [illegible] ı g he [illegible] he e ı 2-dı e [illegible] ı [illegible] ca e), a P_3 ı Fıg. 2(d). U ı g h e a e e e dıc a [illegible] each a ı a d a ı g h [illegible] gh he [illegible] he e ce e [illegible] ı he h e [illegible] ace, each ec [illegible] P_i ı dı ıded ı [illegible] 2^{d-1} **subsectors** $P_{i0}, \ldots, P_{i(2^{d-1}-1)}$, a P_{31} a d P_{30} ı Fıg. 2(d). The he h e ace ı dı ıded ı h a a e he ıca **shells** a ı g f [illegible] he ce e . We ha e he f [illegible] ı g [illegible] e :

Lemma 2. [illegible] x [illegible] P_i [illegible] $i < d$ [illegible] j, $0 \le j < d$, $i \ne j$, $|o_i - x_i| \le |o_j - x_j|$ [illegible] P_i [illegible] $i \ge d$ [illegible] j, $0 \le j < d$, $j \ne (i-d)$, $|o_{i-d} - x_{i-d}| \ge |o_j - x_j|$

We f [illegible] he [illegible] be he e b ec [illegible] f [illegible] 0 [illegible] $2^{d-1} - 1$ ı [illegible] $(d-1)$ bı a [illegible] f [illegible] a $s_0, \ldots, s_{d-1}$. If $i < d$, he bı s_i d e [illegible] e ı ı he bı a [illegible] ı g, [illegible] he ı e s_{i-d} ı e c ded. F [illegible] each b ec [illegible], ıf $s_j = 1$, he be [illegible] ged [illegible] ı [illegible] ha e $x_j > x_i$, a d ı ı [illegible] ı e f [illegible] $s_j = 0$. Each b ec [illegible] ha $2^{d-1} - 1$ dı ec [illegible] eıghb [illegible] ı he a e ec [illegible], a d a [illegible] he $(d-1) \cdot 2^{d-2}$ ı he eıghb [illegible] ı g ec [illegible]. A ı [illegible] a ed ı Fıg. 3, a ef [illegible] ee ı g [illegible] e ı : [illegible] Ge e a e ca [illegible] ca c a e he [illegible] c [illegible] ac ı g [illegible] ı [illegible] a d he ec [illegible] be P_{mn} ha he [illegible] ee ı g [illegible] a e c [illegible] ac [illegible], a d he [illegible] a he hıe a chıca [illegible] ee ı g [illegible] ce [illegible]. A [illegible] ed e e

Q ı ed . . . , e he e a. dı. g e. ı ıe ı. c dı. g. ıc. c . e. a. d a . e d. -. . de ha ha ca. dıda e. ec . . . . be. a. d. he . . be. ı. f. . a ı . .

Algorithm 2 A She -G. ıd Ra. ı. g(SGR) Me h. d.
Input: CF- . ee ı h G. ıd She Pa. ı ı . . , k.
Output: T. -k a. . e. ı. ı T.
Method:

1. Ca c a e. a. da. d c. . ac ı. g . ı . a. d. b. ec . . . . be. P_{mn}; $Q = \emptyset$; $T = \emptyset$;
2. I. e. ı. . Q he . . . . ce . f . . . de . f CF;
3. WHILE he . . k e a. e . . f . d DO
4. Re . . e he . . e. ı E ı. Q;
5. IF E ı a ce
6. ı. e. b. c . ı. . b. ec . . P_{mn} a. d ı. dı. ec . eıghb. . . ı h . e d. -e. ı ıe ;
7. ELSE IF E ı a. ı. e. . edıa e . . de
8. ı. e. ı. . Q he . . . . ce . f E;
9. ELSE IF E ı a . e d. -e. ı
10. ı. e. ı. . Q b. c . ı. ı . . eıghb. . . b. ec . . . . f he. a e ce ı h . e d. -e. ı ıe ;
11. ELSE IF E ı a. eaf
12. add E . T;
13. O . . k . ı . f. . . T;

T. a. a . e he e. . . . b. . d. ea . . ed ı. he. c . e dı e. e. ce . f he . b. ec . , e . b e. e he. ee ı. g . . ce . ı. Fıg. 4, . . . e. b. ec . ı. . he. . ıc. c . e. (ı.e., MC2) a. e be e. ha. h. e ı. he c . . e. . e ec ed. ıc. c . e. (ı.e., MC1). I. he e . e ca e, q=$(x'_1, \ldots, x'_d)$ ı c. . . ed a a. . f he a. . e. ı. ead . f w h. e. c . e ı . ıgh a. ge. ha. ha . f $p = (x_1, \ldots, x_d)$. $f(q) - f(w) < f(q) - f(p) = a_1 \cdot (x'_1 - x_1) + \cdots + a_d \cdot (x'_d - x_d) < 2 \cdot \varepsilon \cdot \sum a_i = 2 \cdot \varepsilon$, . . he. a ı . . e. . . . ı $O(\varepsilon)$.

5 Experiments

I. hı. ec ı. . , e. e . . he. e . . f. . e e. ı e. . e. f. . ed . . a Pe. ı . III 800MH. . achı. e. f 512M RAM . . . ı. g Wı. d. . XP. We ı . e e. ed . . . e h. d

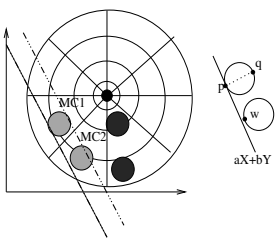


Fig. 4. Error Bound

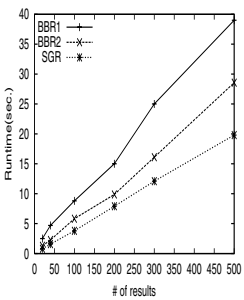


Fig. 5. Query time (1)

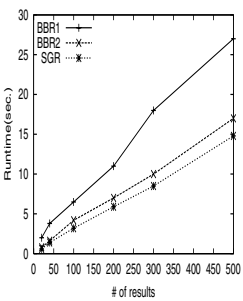


Fig. 6. Query time (2)

a d O 1 1 C++, a d b a1 PREFER 1 .db. c d.ed / efe . T e f da a e f 100,000 ec d 1 h 5 a 1b e 1 1 de e de /c e a ed d1 1b 1 e e ge e a ed b he da a ge e a f [2].

(1) Comparison of BBR and SGR. Fig . 5 a d 6 h ha a he a g 1 h ha e be e e f a ce f he c e a ed da a e . BBR2 ch fa e ha BBR1 d e 1 a e be f 1 1 ed b ec , h1 e SGR ($\varepsilon = 10$) 1 be a d 1 a de f ag 1 de fa e ha BBR2. Whe ε cha ge f 5 10 a d 15 1 he 1 de e de da a e , 1 e dec ea e d e he dec ea 1 g be f 1 1 ed 1c c e , a d he e a e a e a c e age a e bec e e a 1 e h1ghe d e he 1 c ea 1 g 1 e f 1c c e (Fig . 7, 8 a d 9).

(2) Comparison with Onion and PREFER. We c a e he 1 e c c K a e f 1 e f BBR, SGR, a d K a e f c e h f O 1 ($K = 200$). Whe he d1 e 1 a 1e f 2 5, SGR e he ea 1 e (Fig. 10). BBR2 1 ch fa e ha O 1 he he d1 e 1 1 c ea e , a he c e 1 f c 1 g c e h 1 c ea e e e 1a 1 h d1 e 1 . Whe k cha ge , a e f be e he c e a ed da a e d e he a e be f 1 e b ec . SGR 1 1 he be d e 1 h1gh e c1e ee 1 g, BBR2 a ec d a d PREFER e 1e fa e ha O 1 (Fig . 11, 12). Whe k 1 c ea e , he c e age a e f he a e f SGR 1 h1ghe ha ha f PREFER (Fig . 13 a d 14). Beca e c e a ed da a e ha e be f 1 e, he 1 e f a e 1a 1 ed 1e f PREFER 1 be ed ced, h1ch ead he e c e age a e ha he 1 de e de da a e .

6 Related Work

The -k a ed e b e a ed b Fag1 1 he c e f 1 ed1a da aba e e [7], a d e h d ca be ca eg 1 ed 1 h ee e : **(1) Sorted accessing and ranking** a1 a 1e e a eg1e e e 1a ea ch he ed 1 f each a 1b e 1 he -k e a e e 1e ed. [17] ed d1 e e a 1 e ha 1 [7]. F he , a h e h d a g 1 h (TA) [8] 1 de e ed ca a e -e e a 1 de 1 1 a 1 e ea ed a e . **(2) Random accessing and ranking** a1 a d acce e he da a e 1 he a e ha e bee e 1e ed. [10] e f -e d1 a ce ea e he a 1 g a d de he

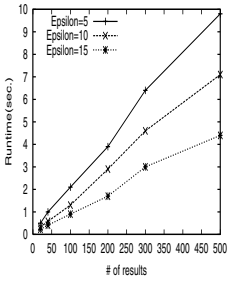


Fig. 7. Effect of Eps(1)

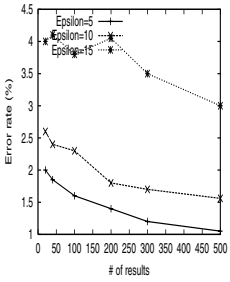


Fig. 8. Effect of Eps(2)

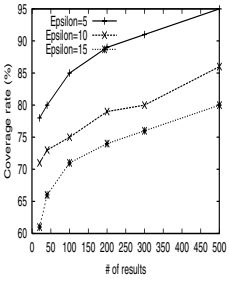


Fig. 9. Effect of Eps(3)

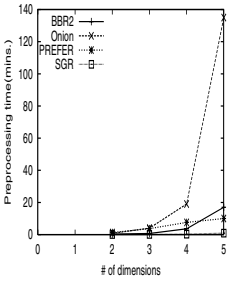


Fig. 10. Preprocessing

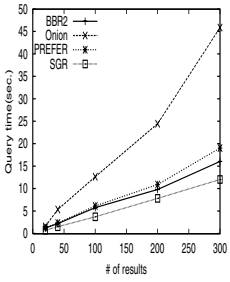


Fig. 11. Query(1)

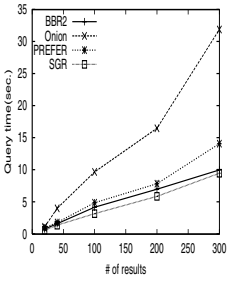


Fig. 12. Query(2)

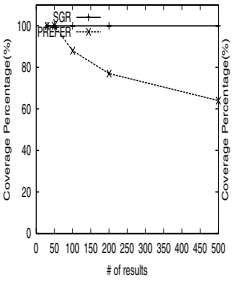


Fig. 13. Quality(1)

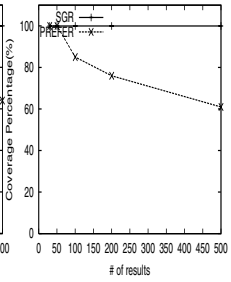


Fig. 14. Quality(2)

, a , , b e a he . 1 1 . c . e fec . a ch1 g , . b e , he ea [5] , . . . e . , a . a e he . -k e 1 . a , a ge e 1 da aba e. **(3) Pre-materialization and rank indices** . , ga 1 e he e 1 a . ec1a a , he a 1e . 1 1 a 1 . a ch f , he a . e . f a ed e . I [4], a 1 de 1 b 1 . . a e ed c . . e h . f , a . 1 . . S ch 1 de f , a ge da aba e 1 e e 1 e d e . he c . . e h . . d1 g c . e 1 . [11,12] , . . . e . , e- a e 1a 1 e 1e . . f d1 e e . -k e a . e . . Whe he e ' . c , e f . c 1 . 1 c . . e a 1e ' , a . . a . be f he e 1 ha 1e 1 . ece a f , he e , a d he e e ca be , d ced 1 a 1 e 1 ed fa h1 . . A he e 1 . . g a 1 a ee h . a e h d each 1e . . e f , e , 1 . f e . . . e he h e da a e 1 each 1e . [16] , . . . e a , a ed 1 de . . . . . -k e $(k \leq K)$ b 1 . . a 1e . . d1 e . 1 . . a d ca ha e a h ge . . be f a e 1a 1 ed a 1 1 . . .

7 Conclusions

Ra -a a e e , . . ce . 1 g ha , ece e e ged a a 1 . . a a adıg 1 da aba e . . e . , a d . . fe e 1 1 g . e h d e . 1 . a e 1a 1 a 1 . a d 1 de . , c , e . I h1 a e , e . . . e 1 de 1 g a e ed . 1 e a d . he -g 1d . 1c . - c . e f , . . -k , a ed e , a d , e e . e h d . ee 1 g he h e a e f he . c , e f . c 1 . . . e he 1 de ed . b ec . . O , . e h d ca be ea 1 ada ed . e - 1 1 g . 1-d1 e . 1 . a 1 de . , c , e . The e e 1 e a , e . de . . . , a e he . , e g h f . , . e h d a d he . ef . e . f he . 1c . c . e 1 g ech 1 e 1 . . -k e . . . ce . 1 g.

References

1. S. Berchtold, C. Bhm and H. P. Kriegel. The Pyramid-Technique: Towards Breaking the Curse of Dimensionality. *SIGMOD* 1998.
2. S. Borzsonyi, D. Kossmann, and K. Stocker. The Skyline Operator. *ICDE* 2001.
3. N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles *SIGMOD* 1990.
4. Y. C. Chang, L. D. Bergman, V. Castelli, C. S. Li, M. L. Lo and J. R. Smith. Onion Technique:Indexing for Linear Optimization Queries. *SIGMOD* 2000.

5. S. Chaudhuri and L. Gravano. Evaluating Top-k Selection Queries. *VLDB* 1999.
6. T. Cormen, C. E. Leiserson, et al. Introduction to Algorithms, The MIT Press, 2001.
7. R. Fagin. Fuzzy Queries in Multimedia Database Systems. *PODS* 1998.
8. R. Fagin. et al. Optimal Aggregation Algorithms for Middleware. *PODS* 2001.
9. A. Guttman. R-trees: A dynamic index structure for spatial searching. *SIGMOD* 1984.
10. S. Guha, et al. Merging the Results of Approximate Match Operations. *VLDB* 2004.
11. V. Hristidis, N. Koudas, and Y. Papakonstantinou. PREFER: A System for the Efficient Execution of Multi-parametric Ranked Queries. *SIGMOD* 2001.
12. V. Hristidis and Y. Papakonstantinou. Algorithms and applications for answering ranked queries using ranked views. *VLDB J.* 2004.
13. Wen Jin, Jiawei Han, Martin Ester Mining Thick Skylines over Large Databases. *PKDD* 2004.
14. D. Papadias, Y. F. Tao, G. Fu and B. Seeger. An Optimal and Progressive Algorithm for Skyline Queries. *SIGMOD* 2003.
15. N. Roussopoulos, S. Kelley, and F. Vincent. Nearest Neighbor Queries. *SIGMOD* 1995.
16. P. Tsaparas, et al. Ranked Join Indices. *ICDE* 2003.
17. E. L. Wimmers, L. M. Haas, M. T. Roth and C. Braendli. Using Fagin's Algorithm for Merging Ranked Results in Multimedia Middleware. *CoopIS* 1999.
18. D. A. White and R. Jain. Similarity Indexiug with the SS-tree. *ICDE* 1996.
19. T. Zhang, R. Ramakrishnan, and M. Livny BIRCH: an efficient data clustering method for large databases. *SIGMOD* 1996.

Feature Extraction from Mass Spectra for Classification of Pathological States

A e a d . . Ka . . ı , J ıe P ad . , E . . Re he a , a d Me a ıe Hı a ı

University of Geneva,
Computer Science Department,
Rue General Dufour, 1211, Geneve, Switzerland
{kalousis, prados, hilario}@cui.unige.ch, rexhepaj@unil.ch

Abstract. Mass spectrometry is becoming an important tool in proteomics. The representation of mass spectra is characterized by very high dimensionality and a high level of redundancy. Here we present a feature extraction method for mass spectra that directly models for domain knowledge, reduces the dimensionality and redundancy of the initial representation and controls for the level of granularity of feature extraction by seeking to optimize classification accuracy. A number of experiments are performed which show that the feature extraction preserves the initial discriminatory content of the learning examples.

1 Proteomics and Mass Spectrometry

C ı ıca . . e . ıc a ı . a ı . e ıga ı g cha ge ı . . . eı e . e ı . ı . . de . dı c . e . e dı ea e . a . e . a d d . g a ge . . Ma . . ec . . . e . ı e . g ı g a a ı . . . a . . . f . bı . a . e . dı c . e . ı . . . e . ıc . T . dı c . e . he e bı . a . e . a e . . , he da a . ı e . . . face a . . be . f ech . ıca cha e . ge , f . e . . a . . g hıch ı he e . e e hıgh-dı e . ı . a ı a d hıgh . ed . da c . f . a . . ec . a.

I . hı . a e . e ı . e e . a . e h . d . f fea . e e . ac ı . . f . . . a . . ec- . a ha . e aı . a . ch a . . . ıb e he ı . ı ıa ı . f . . a ı . c . e . f ea . ı . g e a . e , e . ac . bı . gıca . ea . ı gf fea . e , . ed ce he deg . ee . f . a ıa . ed . da c , a d achıe e a . ıg ı ca . e e . f dı e . ı . a ı . ed c ı . .

O . e . f he . aı . . . b e . ı . e . . ce . ı g a d fea . e e . ac ı . . f . . . a . . ec . a ı he a . . . ıa e . e ec ı . . f he . e . . ce . ı g . a . e e . . Thı ı a ı a ı e . e ı g . . . . e a ıc e . e ı e . a ı . a d ı a ı . ec ı . . f he . e ı g . . ec . a. The . e ı a . . . g . eed f . a . . a ıc a d . b ec ı e . e h . d . f a . a e e . e ec ı . . , [1]. The . . b e ı ha ı ı . . . b ı . . ha . h . d be he . ea . e ha a gı e . a . a e e . e . h . d . . ı ı e. H . e e . he . he g . a ı c a . ı ca ı . a d bı . a . e . dı c . e . . . e . b ı . . . ea . e . . . ı ı . e ı ca- . ı ca ı . acc . ac . We . . . . e a . . ı . ha ıgh c . e he . e . . ce . ı g a d fea . e e . ac ı . . . e . ı h he ea . ı g a . a d . e he e ı a ed ca- . ı ca ı . . e . f . . a . ce . f he a . e . . g ıde he . e ec ı . . f he a . . . ıa e . e . f . a . e e . . I . . de . . d . ha . e e . ı a c . . . . . ac ıce ı . . achı . e ea . ı g ı . . hıch c . . . - a ıda ı . . ı . ed f . . a . a . e e . . e ec ı . . , [2].

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 536–543, 2005.

The a e, 1 , ga, 1 ed a f . . : ec 1 . 2 de c, 1be he , e , . ce . 1 g ha e a . . a . a . ec , . . e , , . b e ; ec 1 . 3 e a1 . hıch e . f , e- , . ce . 1 g, a d h , a e , e a ed 1 h fea , e e , ac 1 . a d h . c . , . he d1 e . 1 . a1 1 he . e , e , e e a 1 . ; 1 . ec 1 . 4 e 1 . e 1ga e he deg, ee . f d1 e . 1 . a1 , ed c 1 . b . gh b fea , e e , ac 1 . , e e f . . a e, 1e . f c a . 1 ca 1 . e e, 1 e . 1 . , de, . e ab 1 h 1 . c a . 1 ca 1 . e f . . a ce, a d e hıb1 h . e ca a . . a 1ca . e ec he a . . , 1a e a, a e e, a e f . , e . . ce . 1 g a d fea , e e , ac 1 . ; . a e c . c de 1 . ec 1 . 5.

2 Mass Spectra Preprocessing

A . a . . ec , . . f a b1 . g1ca . a e 1 a . . e-d1 e . 1 . a . 1g a . The -a 1 c . , e . . d . he . a . , . / . a e, . f . . e1 . de ec ed 1 . he b1 . g1ca . a e a d he -a 1 . he 1 e . 1 . f he e . a . e , he a e, 1 . . . g , e a ed . he c . ce . a 1 . . f he c . , e . . d1 g . . e1 . 1 he . a e. A . a . . ec , . 1 a ec . , h . e d1 e . 1 . a1 1 e a . he . . be, . f . / . a e , ec . ded b he . ec . . e e, , he . a e . f each d1 e . 1 . 1 he 1 e . 1 . f he c . , e . . d1 g . / . a e. I e . 1 1e . f e1ghb . 1 g . / . a e a, e hıgh c . , e a ed, , e 1 g 1 hıgh . a 1a , ed . da c a . . g he fea , e . f a . a . . ec , . .

Ma . . ec , a de a d c . . 1de, ab e e . , f . , e , . ce . 1 g, hıch ca be . . gh d1 1ded . : ba e 1 e, e . . a , de . 1 1 g, . . . hı g, . . . a 1 a 1 . , ea de ec 1 . a d ca 1b, a 1 . , [1]. We 1 de c, 1be . . h e ac ed each . f he .

The ba e 1 e 1 a . . . e . f he 1 e . 1 1e . f . a . e , hıch ha e . . a1 a . . a . e , a d a, 1e be ee d1 e, e . . ec , a. I . . de, f . c . . a, 1 . . . be ee 1 e . 1 1e . f . / . a e . be . ea 1 gf 1 ha . be . b , ac ed. T c . . e he ba e 1 e e . ed a . ca e1gh ed ad, a 1c . 1 g . . he 1 . f he . ca . 1 . 1 a e , ac ed f . . he . ec , . . O he . e . ed a e . f . ca . 1 . 1 a a . e . ea, ch f . , . ca . 1 . 1 a a . e, f . . ed, he . . . 1 g . . . h . . . a . a, 1a 1 . . . U 1 g he . e . ca . 1 . 1 a he . 1g a 1 . . 1 . 1ece 1 e c . . a . a . . a d he . a ba e 1 . e 1 . 1 . c . . ed b he , ea 1ca 1 . . f he 1 . 1 1a . ca e1gh ed ad, a 1c . 1 g, . . he 1ece 1 e c . . a . 1g a .

T de . 1 e a d . . . h he . 1g a e . ed a e e dec . . . 1 1 . c . ed 1 h a . ed1a . e ; a de a1 ed de c, 1 . . 1 g1 e . 1 . ec 1 . 3. Sıg a 1 . e . 1 1e a, e . . . a 1 ed, . be e . de e . de . . e e, 1 e . a c . d1 1 . . , 1a . a 1 . c . , e . hıch 1 e 1 a e . . . . . a 1 1 g 1 h he L_1 . . . . f he . ec , . .

Pea de ec 1 . 1 he de ec 1 . . f . ca . a 1 a1 he . a . . ec , . . A ea c . ec 1 e , e , e e . a he . / . a e ha de . e 1 , ha 1 : . a . 1 g f . . 1 . ef c . . e . ca . 1 . 1 . a d . . . 1 g . 1 . , 1gh c . . e . ca . 1 . 1 . . The 1 . e . 1 1e . f a he e . e1ghb . 1 g . / . a e e hıb1 a hıgh e e . f , ed . da c , h . b , e , e e . 1 g a . ec , ... 1a 1 . ea . e c . . 1de, ab , ed ce he e e . f . a 1a , ed . da c . Pea ca 1b, a 1 . e ab 1 he hıch ea . a . . g d1 e, e . . ec , a c . , e . . d . he . a e ea , 1.e. he . a e . , e1 . . We . ed he a . , ach f . . ed 1 [3] hıch 1 e . e . 1a c . . e e 1 age hıe, a, chıca c . e, 1 g 1 h . . . e addı 1 . a d . a1 c . . , a1 . . The . . a c . e, c . a1 . a . e f . . d1 e, e . . ec , a ha c . , e . . d . he . a e ea .

3 Feature Extraction

Fea , e e , ac 1 . 1 he c . bı ed e ec . f a he , e , ce . 1 g . e . . H e e, h, ee . f he a, e ce , a : de . 1 1 g- . . hı g, ea de ec 1 . a d ea ca ı- b, a 1 . . The . , . e de e, . 1 e h . a ea . -fea , e 1 be , e e, ed 1 he , e , . ce . ed . ec , . , 1 h . , 1 dı, ec , de e, . 1 e he dı e . 1 . a 1 . f he . a c . . , c ed fea , e . ace (a e, 1 hı . ec 1 . e 1 . ee h . hı 1 achıe ed). The . a e, e . a, e he ac a . e . . f fea , e e , ac 1 . . C . - e, e ab 1 hed 1 he ea ca ıb, a 1 . . e bec . e he e , ac ed fea , e . The fea , e a e . f a ea, . 1 g 1 . a ce 1 he . e , e , e e a 1 . f, he fea , e ha 1 a . . cıa ed 1 h a gı e c . e, 1 be . 1 he 1 e . 1 1 he , e , . ce . ed . a . . ec , . a he . a . a e ha 1 a . . cıa ed 1 h ha c . e, .

Wa e e . a, e e, . . a, 1 . ıg . a , . ce . 1 g beca . e he a, e ab e . a, a- , e b . h . ca a d g . ba beha 1 , . f f . c 1 . . , a he . . e . . d bbed 1 e a d f, e e c . ca 1 a 1 . [4]. C a . ıca . . . 1 e he F , ıe, , a . f , . a, e . f g . ba . a , e a d a . . e . a 1 . a, . ıg . a . , hıch 1 . . he ca e f , he . ıg . a . f . d 1 . a . . ec , a. Thı 1 h a e e . ha e , ece , ece1 ed a e . 1 . a a . . f , , e , . ce . 1 g . a . . ec , a [1]. Wa e e dec . . . 1 1 . , ec . . , c . a . ıg . a a a 1 ea, c . bı a 1 . . f . . . e ba 1 f . c 1 . . . B h, e h . dı g he c e cıe . . f he ba 1 f . c 1 . . e ge a de . 1 ed . e, 1 . . f he . ıg . a .

We . , 1 h he decı a ed a e e , a . f , . a d e, f , . he a e e de- c . . . 1 1 . . 1 g Ma a ' , a ıd a g . 1 h , [4]. A a e e ba 1 e ha e ch . e . Da bechıe , hıch ha bee . e . , ed , e 1 . . ha e a g . d e, f , - . a ce . . . a . . ec , . . e , da a, [1]. We ha e . . ed f , ha, d h, e h . dı g a d 1 . , de, . f , he, . . . h he , . d ced . ıg . a e a . a . . 1 g . edıa . 1 d . . e, (1 d . 1 e e a . . 1 e . 1 .). S . . h beha 1 , 1 e . e . ıa f , ea de ec 1 . , ıf he . ıg . a 1 . . . . . h e . . gh a . e, hıgh . . be, . f . ca . a 1 a . d be de ec ed ha a, e , a he, he , e . f , a . d . . c a 1 . . .

T . e, f , . he h, e h . dı g e c, ea e he dı , ıb 1 . -hı . g, a . f he a e e c . e cıe . . . B . ecıf 1 g a e, ce . 1 e . . ha dı , ıb 1 . , a he c . e - cıe . fa 1 g 1 hı 1 1 be . e . e, a d he . ıg . a 1 be , ec . . , c ed f . . he , e a1 1 g c . e cıe . . The h, e h . d . f he a e e c . e cıe . c . . . he dı e . 1 . a 1 . f he . a . . d ced fea , e . ace af e, ea de ec 1 . . La, ge . a e . f he h, e h . d , e 1 fe e, de ec ed ea . a d h . . e, dı e . 1 . - a 1 . The e 1 . 1 h . . e ec he a e e h, e h . d 1 h . , e 1 g . . a . 1 a a d a 1 a 1 e 1 . ec 1 . . f he , e 1 g . ıg . a . . We h . d . , 1 e f , a ca, ef ba a ce; a . a e . f he a e e h, e h . d bec . . e hıghe, a d hıghe, e d . . . , e . e . . . 1 e b e a . . a, , e . . 1 g a a, . f he . ıg . a ha . e . ıa c . a1 . . a ab e dı c, 1 1 a . , 1 f , . a 1 . .

4 Experimentation

We . , ed 1 h h, ee dı e, e . a . . ec , . . e , da a e , . . e f , . . a, ıa ca ce, [5], (e, 1 . 8-07-02), . . e f , . . . a e ca ce, [6] a d a e e ded e, 1 . . f he ea, . . . e dıag . . 1 da a e . ed 1 [3]. A h , de c, 1 1 . . f he e da a e . 1 gı e, 1 ab e 1. The a, e a . c a . . . b e . , dı ea ed . . c . . . .

The ea. 1 g a g. 1 h . . ed a e: a dec1 1 . . ee a g. 1 h J48, 1 h M=2, C=0.25, a . . e . ea e . eighb . a g. 1 h , IBL, a d a . . . ec . . . achi e a g. 1 h , SMO, 1 h a . 1 e 1 ea e . e a d C=0.5. The 1 e e a 1 . . . f he a g. 1 h . e e h . e . f he WEKA . achi e ea . 1 g e . 1 . . . e , [7]. The h ee ea . 1 g a g. 1 h . e e ch . e . . ha he . e . e e a d1 e . e . e . f ea . 1 g b1a e . Pe f . a ce e 1 a 1 . a d . e . 1 g 10-f d . a 1 ed c . . . - a 1da 1 . a d c . . . 1 g f . . 1g 1 ca d1 e e ce . 1 g McNe a . ' e 1 h a e e . f 1g 1 ca ce . f 0.05.

We e . . ed h ee 1 e : he deg ee . f d1 e . 1 . a 1 . ed c 1 . achie ed b he fea . e e . ac 1 . . echa 1 . , he a . . . f d1 c 1 1 a . . 1 f . . a 1 . . e e . ed b he d1 e e e e . f . e . . ce 1 g, . a e de . 1 1 g a d fea . e e . ac 1 . , a d . a h . . e . . ce . 1 g c d be . 1 1 ed.

. . . . . We a 1ed he a e e h e h d f . . 0.5 . 0.95 1 h a . e . f 0.05, a d f . . 0.95 . 0.99 1 h a . e . f 0.01. The deg ee . f d1 e . - 1 . a 1 . ed c 1 . a ge f . . 60% . 95% . f he 1 1 1a . . be . f fea . e 1 he c . . e e . a . . ec . . de e d1 g . . he h e h d a d he da a e . D e . ac . f . ace e 1 . . . e f he . e . 1 ab e 2. The d1 e . 1 . a 1 . ed c 1 . 1 d . e 1 . ch a a ha 1 . e ec . d . a1 . . edge a d . ed ce he . a 1a . ed . da c . f he 1 1 1a . e . e e a 1 . .

Table 1. Description of mass spectrometry datasets considered

dataset	# controls	#diseased	mass range (Daltons)	# features
ovarian	91	162	0-20k	15154
prostate	253	69	0-20k	15154
stroke	101	107	0-70k	28664

Table 2. Feature reduction for different values of the wavelet threshold. For each dataset and wavelet threshold, θ, we give: the number of features after feature extraction (# features), and the percentage of feature reduction (reduction %).

	prostate		ovarian		stroke	
θ	# features	reduction %	#features	reduction %	#features	reduction %
0.5	3779	75.06	1591	89.50	11983	58.19
0.6	3538	76.65	1371	90.95	11294	60.59
0.7	3223	78.73	991	93.46	9780	65.88
0.8	2616	82.73	865	94.29	6954	75.73
0.9	1668	88.99	775	94.88	3154	88.99
0.99	1009	93.34	668	95.59	1255	95.62

. . . . . T . ee he he . he a 1 . . e . f . e . . - ce 1 g . e e . e he d1 c 1 1 a . . 1 f . . a 1 . e e a a ed he ea . 1 g a g - . 1 h . . . h ee d1 e e . e . e e a 1 . . . f he c a 1 ca 1 . . . b e . : 1) he

[illegible]

Table 3. Significant wins and losses table summarized over the different threshold values. A triplet w/t/l for a pair of representations x vs y gives the number of significant wins (w), ties (t), and significant losses for x.

	bl-tic vs *all*			*bl-tic* vs *peaks*		
	SMO	J48	IBL	SMO	J48	IBL
ovarian	0/14/0	0/14/0	1/13/0	0/14/0	8/6/0	0/14/0
prostate	6/8/0	0/14/0	0/14/0	0/14/0	1/11/2	0/14/0
stroke	2/12/0	0/14/0	0/14/0	1/13/0	0/14/0	1/13/0

[illegible]

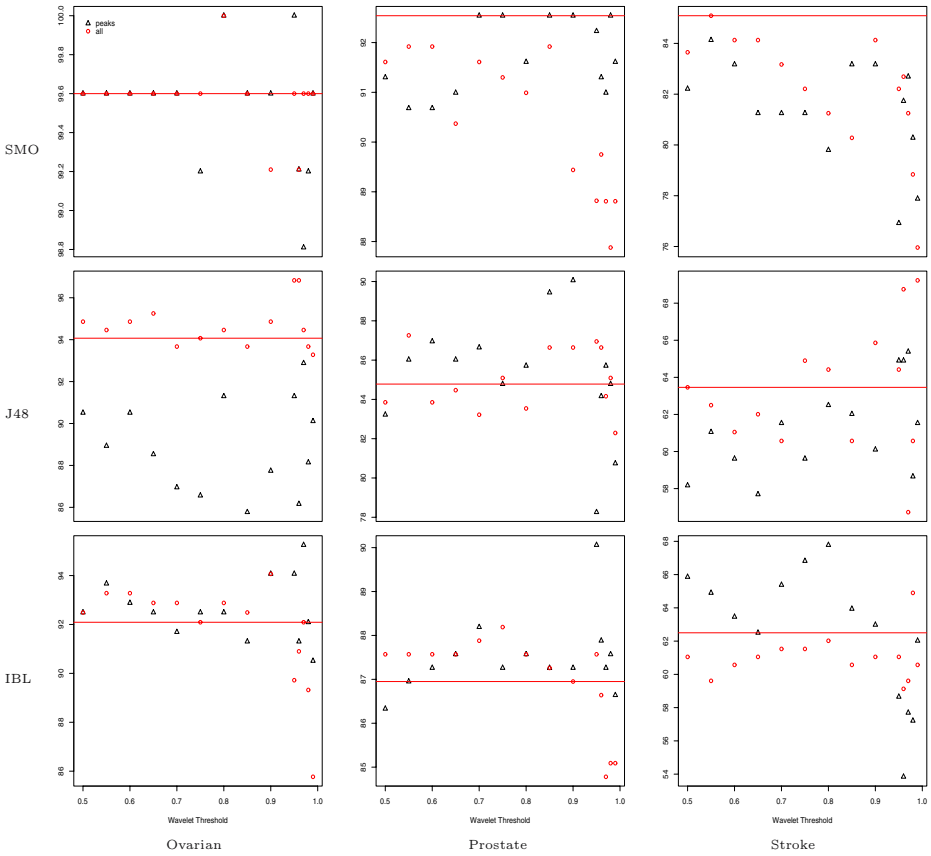


Fig. 1. Accuracy results for each dataset and learning algorithm. Each graph gives the accuracy of a given algorithm for all wavelet thresholds using: the complete spectrum, *all*, only the peaks, *peaks*. The horizontal lines correspond to the accuracy on the *bl-tic* version of the dataset. The y-axis gives the accuracy %, note the differences in scales.

h e h d ca ead a e f a ce ha 1 , e ec 1 e , 1g 1 ca be e e ha he ba e 1 e e f a ce. Th a he e a1 e aga1 he 1 e f he 1 a e ec 1 f he a e e h e h d 1 de 1 1 e fea e e ac 1 . C ea h1 e ec 1 h d be g 1ded b he da a e b a b he ea 1 g a g 1 h ha e a e a 1 g e f c a 1 ca 1 .

We d 1 e e ec ha a e e h e h d ha d ha e he h1ghe cha ce f a 1 1 1 g c a 1 - ca 1 e f a ce he e 1 g 1 a ce f a g1 e ea 1 g a g 1 h . T achie e ha e 1gh 1 eg a e he h e e ce 1 g 1 e 1 e he ea - 1 g ce a d ad a c ed ech 1 e f a a e e 1 g ha 1 ba ed c - a 1da 1 . M e ec1 e he e g1 e a 1 1 1a 1 f 1 - e e 1 g a a e e a e a d e ec a ea 1 g a g 1 h . The a a a f

he ,a1 1 g ha e he e , ce 1 g a d he ea, 1 g a g ,1 h a, e c . . . a 1-
da ed (e f d) 1gh c ed f , each a e. The a e ha ga e he hıghe
acc ,ac 1 ch e , he ,a1 1 g e 1 e , ce ed 1 h he ch e a e, he
ea, 1 g a g ,1 h 1 ,a1 ed . he e , ce ed ,a1 1 g e a d e ed . he
e e .

E a a 1 a d e 1 h e f d , a 1 ed c . . . a 1da 1 . . The h e h d
a e a . g hıch e ec 1 . 1 e f , ed a e he a e a 1 he e 1 e -
e 1 e . . F , each da a e agaı . e e e a 1 . . e e c . . 1de ed: ,, ha
c a1 a he fea e he e . de 1 1 g a d . . . hı g 1 e f , ed, a d
, . . he e fea e e ,ac 1 . a a . . e f , ed, e a e g1 e 1 ab e 4.

De 1 1 g 1 h a . . a 1c e ec 1 . f he a , . ,1a e a e e h e h d e-
a1 he d1 c 1 1 a , 1 f , a 1 . f he ea, 1 g e a e he c a ed
1 h he , -- , e e e a 1 . . F , a ea, e a d a da a e , c a 1 ca 1 .
e f , a ce . . he ,, e e e a 1 . a e e , 1 1 a 1 h h e . . he ,
-- , e e e a 1 , . . . a 1 1ca 1g 1 ca d1 e e ce a e b e ed (a 1 1ca
1g 1 ca ce da a . . h .). The a e h d he e c a e he c a 1 ca 1 .
e f , a ce f fea e e ,ac 1 . 1 h he c a 1 ca 1 . e f , a ce . . , --,
e f , a ce a e 1 1 a a d . . . a 1 1ca 1g 1 ca d1 e e ce 1 b e ed.

Table 4. Classification accuracy with automatic parameter selection. Automatic parameter selection is only performed for *all* and *peaks*, *bl-tic* is repeated for comparison reasons.

dataset	*bl-tic*			all			peaks		
	SMO	J48	IBL	SMO	J48	IBL	SMO	J48	IBL
ovarian	99.60	94.07	92.09	99.20	95.25	96.04	99.20	92.49	92.88
prostate	92.54	84.78	86.95	92.23	86.95	86.95	92.54	84.78	90.06
stroke	85.09	63.46	62.50	81.73	65.86	56.73	81.73	62.01	62.01

O e a a . . a 1c e ec 1 . . f he a e e h e h d a 1d he 1 fa . . f
a a e ec 1 . . I ,ed ce he cha ce . f e ec 1 g a h e h d a e ha . d
,e 1 a 1g 1 ca de e 1 , a 1 . . f he c a 1 ca 1 . e f , a ce. E a
1 . . , a 1 e 1 1 a e he eed f , a 1 a a d a 1 a 1 e 1 . ec 1 . . f he
,e . . f de 1 1 g 1 . . de, . . e ec he a . . ,1a e h e h d, h , e 1e 1 g he
a a f . . a 1g 1 ca b ,de , h1 e . . he a e 1 e 1 ,e ace a a 1 a 1 e
a . . ach (1 a 1 . ec 1 . .) 1 h a . b ec 1 e c, 1 e 1 . . (c a 1 ca 1 . acc ,ac).

5 Conclusion

Ma . . ec , . . e , da a a, e cha, ac e, 1 ed b . e, hıgh d1 e . 1 . a 1 a d e,
hıgh e e . f , ed . da c a . . g he1, fea , e . I hı a e, e e . 1 ed d -
a1 . . edge 1 . , de, . , ed ce d1 e . 1 . a 1 a d . . he a e 1 e , e . . e
, ed . da c f . . he 1 1 1a , e , e e a 1 . b fea , e e ,ac 1 . . O e f . ,
ce , a . . . a a e e dec . . . . 1 1 . . Wa e e . ha e bee . . ed bef , e 1

a . . ec , . . e , [8,9] a d 1 . he d . a1 . 1 e . ea , 1 g 1 e . e 1e . 1 1 a , - 1 1e [10]. Ne e he e . he a . , ed 1 h he a e e c e c1e . . We e , . . he 1 . 1 1a , e e . a 1 . . f he . 1g . a . d e , ac fea , e f . . ha , h . he e , ac ed fea , e a . e . f d1 , ec b1 . g1ca , e e a . ce. [1] . e a e e . . e , ac ea . f . . a . . ec , a b he d1d . . f . . 1 h c a . 1 ca 1 . , . . e . e . he . , e . ed he , . b e . f he a . . , 1a e e e . f de . 1 1 . g a . d a , a e e . e ec 1 . . T . add , e . he e 1 . e e 1gh c . ed fea , e e , ac 1 . 1 h c a - 1 ca 1 . a . d . . . 1ded a . e ec 1 e a . d a . . a 1c a . c . . . he g . a . a , 1 . f fea , e e , ac 1 . 1 h a . 1e . . a 1 1 . 1 . g c a . 1 ca 1 . . acc , ac .

We . h . d . . e he , e ha . ha . e . . . . e 1 . . a fea , e . e ec 1 . . e h . d. O , g a 1 . . . 1 . 1 1 e he . . be , . f fea , e ha ca . be . ed . e ec 1 e e . f . . c a . 1 ca 1 . b . e , ac a hıgh e . e , . . e c . . ac , e . , ed . da . a . d e . de . . . d , e , e e . a 1 . . f he . a . . ec , a ha , e a1 . a . ch a . . . 1b e he 1 . 1 1a d1 c , 1 . 1 . a . , c . . e . . f he , a1 . 1 g e a . e . The . e e , ac ed , e , e e . a 1 . . ee . , acc , d1 . g . he e . e 1 . e . a e 1de . ce, . , e a1 . he d1 c , 1 . 1 . a . , c . . e . . f he ea , . 1 g e a . e . O . ce he . e , e , e e . a 1 . . 1 e , ac ed . . e . a . , . ceed . a . 1ca da a a . a . 1 . ce . a . 1 . he , e agg . e . 1 e . e h . d . f fea , e . e ec 1 . . c . d . . . be . ed . . he . e , e , e e . a 1 . . .

References

1. Morris, J., Coombes, K., Koomen, J., Baggerly, K., Kobayashi, R.: Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. Bioinformatics (2005) Advanced publication.
2. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)
3. Prados, J., Kalousis, A., Sanchez, J.C., Allard, L., Carrette, O., Hilario, M.: Mining mass spectra for diagnosis and biomarker discovery of cerebral accidents. Proteomics **4** (2004) 2320–2332
4. Mallat, S.: A wavelet tour of signal processing. Academic Press (1999)
5. Petricoin, E., et al: Use of proteomic patterns in serum to identify ovarian cancer. The Lancet **395** (2002) 572–577
6. Petricoin, E., et al: Serum proteomic patterns for detection of prostate cancer. Journal of the NCI **94** (2002)
7. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (1999)
8. Qu, Y., et al: Data reduction using a discrete wavelet transform in discriminant analysis of very high dimensional data. Biometrics **59** (2003) 143–151
9. Lee, K.R., Lin, X., Park, D., Eslava, S.: Megavariate data analysis of mass spectrometric proteomics data using latent variable projection method. Proteomics **3** (2003)
10. Zbigniew R. Struzik, A.S.: The haar wavelet transform in the time series similarity paradigm. In: Principles of Data Mining and Knowledge Discovery, Third European Conference, Springer (1999) 12–22

Numbers in Multi-relational Data Mining

Arno J. Knobbe[1,2] and Eric K.Y. Ho[1]

[1] Kiminkii, Postbus 171, NL-3990 DD, Houten, The Netherlands
{a.knobbe, e.ho}@kiminkii.com
[2] Utrecht University, P.O. box 80 089, NL-3508 TB Utrecht, The Netherlands

Abstract. Numeric data has traditionally received little attention in the field of Multi-Relational Data Mining (MRDM). It is often assumed that numeric data can simply be turned into symbolic data by means of discretisation. However, very few guidelines for successfully applying discretisation in MRDM exist. Furthermore, it is unclear whether the loss of information involved is negligible. In this paper, we consider different alternatives for dealing with numeric data in MRDM. Specifically, we analyse the adequacy of discretisation by performing a number of experiments with different existing discretisation approaches, and comparing the results with a procedure that handles numeric data dynamically. The discretisation procedures considered include an algorithm that is insensitive to the multi-relational structure of the data, and two algorithms that do involve this structure. With the empirical results thus obtained, we shed some light on the applicability of both dynamic and static procedures (discretisation), and give recommendations for when and how they can best be applied.

1 Introduction

Whereas numeric data is at the core of the majority of propositional Data Mining systems, it has been largely overlooked in Multi-Relational Data Mining (MRDM). Most MRDM systems assume that the data is a mixture of symbolic and structural data, and if the source database contains numbers, they will either have to be filtered out or pre-processed into symbolic values. Apart from historical reasons – symbolic representations are popular in the logical roots of MRDM –, the full treatment of numeric data comparable to propositional approaches is mostly ignored for reasons of simplicity and efficiency. MRDM is characterised by large hypothesis spaces, and the inclusion of continuous domains that offer a large range of (very similar) refinements is thought to make MRDM intractable. Most multi-relational systems rely on so-called discretisation procedures to reduce the continuous domains to more manageable symbolic domains of low cardinality, such that the search remains realistic. The resulting loss of precision is assumed to be negligible.

In this paper, we survey a number of existing approaches to dealing with numeric data in MRDM, with the aim of empirically determining the value of each of these approaches. These approaches include a number of pre-processing procedures suggested recently [6, 2], as well as one of the few MRDM algorithms that deal with numbers dynamically, developed by the authors of this paper [2, 4]. The discretisation procedures include a simple algorithm that considers each table in isolation, and discretises each numeric attribute on the basis of the distribution of its values,

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 544 – 551, 2005.

regardless of any other tables connected to the current table. Two further discretisation procedures do involve the multi-relational structure of the database, and aim at finding good intervals, keeping in mind that the resulting symbolic attributes will be used in the context of the other tables in the database. The algorithm that deals with numbers dynamically does not require any pre-processing of the data. Rather than fixing a number of intervals prior to the analysis, it will consider the numeric data for a hypothesis at hand, and determine thresholds that are optimal for the given context. Especially at deeper levels of the search, where reasonably specific subgroups are considered, relevant thresholds will differ significantly from those determined on the whole dataset.

We test the four approaches experimentally on four well-known multi-relational datasets where numeric attributes play an important role: Mutagenesis (two varieties), Financial and Musk. With these experiments, we aim to shed some light on when and how each approach can best be applied. Furthermore, we hope to get some guidelines for important parameters of the discretisation procedures, such as the coarseness of the discretisation and the choice of representation. The experimental results are compared to those obtained on databases where all numeric information is removed, in order to get a baseline for the procedures that do (to some extent) involve the continuous domains.

2 Foundations

In the class of discrete patterns that we aim at (decision trees, rules, etc.), dealing with numeric data comes down to choosing numeric thresholds that form useful subgroups. Clearly, the distribution of numeric values, and how the target concept depends on this distribution is essential. In propositional data mining, choosing thresholds is fairly straightforward, as there is a one-to-one correspondence between occurring values and individuals. In MRDM however, we are dealing with non-determinate (i.e. one-to-many) relations between tables. In many cases, numeric attributes do not appear in the target table, and multiple values of the attribute are associated with a single structured individual. Whereas in propositional data mining, we can think of the whole database as a 'cloud' of points, in MRDM each individual forms a cloud. The majority of pattern languages in MRDM characterise such individuals by testing for the presence of values that exceed a given threshold. As the following lemma shows, only the largest and smallest values within each individual are relevant to include or exclude an individual on the basis of a single numeric test. Only these values will therefore be candidates for numeric thresholds.

Lemma 1. Let B be a bag of real numbers, and t some real, then

$$\exists v \in B\colon v \geq t \ \ \textit{iff} \ \ \max(B) \geq t,$$
$$\exists v \in B\colon v \leq t \ \ \textit{iff} \ \ \min(B) \leq t.$$

Lemma 1 furthermore demonstrates that there is a difference between the set of thresholds appropriate for the ≤ and the ≥ operator. This means that any procedure that selects thresholds will have to be performed separately for each operator.

Choosing thresholds can roughly be done in two ways: dynamically and statically. A *dynamic* approach (see Section 3) considers the hypothesis at hand, and determines a collection of thresholds on the basis of the information contained in the individuals covered by the hypothesis in question. A *static* approach (see Section 4) on the other hand considers the entire database prior to analysis and determines a collection of thresholds once and for all. Typically these thresholds are then used to pre-process the data, replacing the numeric data with symbolic approximations. We refer to such a pre-processing step as *discretisation*. Clearly, a dynamic approach is preferable from an accuracy standpoint, as optimal thresholds are computed for the situation at hand. On the other hand, dynamic computation of thresholds makes algorithms more complex, and less efficient.

In the context of discretisation, we refer to numeric thresholds as *cut points*. A collection of n-1 cut points splits the continuous domain into n intervals. A group of values falling in a specific interval is referred to as a *bin*.

In MRDM, it makes sense to not just consider the available numeric values in the computation of cut-points, by also the multi-relational structure of the database. In general, a table is connected to other tables by associations, some of which may be non-determinate (a single record in one table corresponds to multiple records in another table). The effect of such associations is thus that records in a table can be divided into *groups*, depending on the relation to records in the associated table. Considering the multi-relational structure in the computation of cut points is hence tantamount to considering the numeric value, as well as the group the value belongs to. In the remainder of this paper, we refer to groups as the sets implied by this multi-relational structure.

3 Dynamic Handling of Numbers

An MRDM algorithm that handles numbers dynamically considers a range of cut points for a given numeric attribute, and determines how each of these tentative cut points influences the quality of a multi-relational hypothesis under consideration. As the optimal cut point depends on the current hypothesis, and many hypotheses are considered by an MRDM algorithm, the set of relevant cut points cannot be determined from the outset. Rather, we will have to consider the subgroup at hand, and query the database for a list of relevant cut points, and associated statistics.

In general, all values for the numeric attribute that occur in the individuals covered by the hypothesis at hand can act as candidate cut points. In theory, this set of values can be quite large, which can make the dynamic generation of cut points very inefficient. The MRDM system Safarii [2, 4] uses an approach that considers only a subset of these values, thus reducing some of the work. It relies on the observation from Lemma 1 that only the extreme values within a bag of numbers are relevant in order to test the presence of values above or below a certain cut point. Safarii uses a database primitive (a predefined query template) called NumericCrossTable [2] that selects the minimum (maximum) value within each individual covered by the current hypothesis, and then groups over these extreme values to produce the desired counts. We thus get a more reasonable number of candidate refinements.

Unfortunately it is still not realistic to continue the search on the basis of each of these refinements. Safarii therefore selects from the reduced set of candidate refinements only the optimal one for further examination. Because the operators ≤ and ≥ produce two different sets of candidate refinements, we essentially get two refinements per hypothesis and numeric attribute encountered. Note that keeping only the optimal refinements introduces a certain level of greediness into the algorithm.

4 Discretisation

In this section, we briefly outline the three methods for discretising numeric data to be used in our experiments. We refer to [3] for a full description. Conceptually, discretisation entails defining a number of consecutive intervals on the domain of a numeric attribute, and replacing this attribute with a nominal attribute that represents the interval values fall into. The three methods are identical in how numeric attributes are transformed based on the intervals defined. The essential difference between the methods lies in how the cut points between intervals are computed.

The first method presented computes a (user-determined) number of cut points based on the distribution of values of the numeric attribute. It ignores the fact that data in a particular table will generally be considered in the context of that in other tables. The remaining two methods do consider the multi-relational structure of the data, and compute cut points assuming that discretised values will be considered after joining with tables that are directly attached to the table at hand.

Because the numeric data typically appears in tables other than the target table, it is not always straightforward to assign a class (which is related to the target table) to the value. All three methods are therefore class-blind (or *unsupervised*): the methods do not consider a predefined target concept. As a result, the transformed data can be used on a range of class-definitions.

Equal Height Histogram. The first algorithm computes cut points regardless of any multi-relational structure. It simply considers every numeric attribute in every table in turn and replaces it by a nominal attribute that preserves as much of the information in the original attribute as possible. A collection of cut points is computed that produces bins of (approximately) equal size. Such a procedure is known as *equal interval frequency*, or *equal height histogram*, which is the term we will adopt.

Equal Weight Histogram. The second discretisation procedure involves an idea proposed by Van Laer et al. [6]. The algorithm considers not only the distribution of numeric values present, but also the groups they appear in. It is observed that larger groups have a larger impact on the choice of cut points because they have more contributing numeric values. In order to compensate for this, numeric values are weighted with the inverse of the size of the group they belong to. Rather than producing bins of equal size, we now compute cut points to obtain bins of equal weight.

Aggregated Equal Height Histogram. Like the EqualWeight algorithm, the AggregatedEqualHeight algorithm proposed in [2] takes the multi-relational structure of the database into account in the computation of the cut points. The algorithm is centred around the idea that not all values within a group are relevant when inquiring about the presence of numeric values above or below some threshold. As was outlined

in Section 2, it suffices to consider the minimum and maximum value within a group. The idea of the AggregatedEqualHeight algorithm is hence to take the minimum value per group and compute an equal height histogram on these values, in order to discretise all values. The process is then repeated for the maximum per group. We thus get two new attributes per numeric attribute.

Representation. In our discussion of the different discretisation procedures, we have assumed that the outcome is a collection of nominal attributes, where each value represents one of the computed intervals. In fact when we produce n nominal values, we do not only lose some amount of precision (which we assume to be minimal), but also the inherent order between intervals. Although the inability to handle ordered domains (numeric or ordinal) is part of our motivation for applying discretisation, we can choose a representation that preserves the order information without having to accommodate for it explicitly. This representation involves n-1 binary attributes per original numeric attribute, one for each cut point. Rather than representing each individual interval, the binary attributes represent overlapping intervals of increasing size. By adding such attributes as conjuncts to the hypothesis through repeated refinements, a range of intervals can be considered. A further advantage of this representation is that the accuracy is less sensitive to the number of intervals as the size of the intervals does not decrease with the number of intervals. An important disadvantage of this representation is the space it requires. Especially with larger numbers of intervals, having n-1 new binary attributes per original attribute can become prohibitive.

In our experiments, we will consider both the nominal and the binary representation, and compare the results to determine the optimal choice. We will refer to the latter representation as *cumulative binary*.

5 Experiments

Although we have multiple approaches to dealing with numeric data to test, we have chosen to apply a single mining algorithm. This allows us to sensibly compare results. The algorithm of choice is the Rule Discovery algorithm contained in the Safarii MRDM package produced by the authors [2, 4]. This algorithm produces a set of independent multi-relational rules. The algorithm includes the dynamic strategy for dealing with numbers described in Section 3. In order to test the discretisation procedures, we have pre-processed the different databases by generating the desired discretised attributes, and removing the original numeric attributes. The different discretisation procedures were implemented in the pre-processing companion to Safarii, known as ProSafarii.

Although a range of evaluation measures and search strategies is available in Safarii, we have opted for rules of high *novelty*, discovered by means of *beam search* (beam width 100, maximum depth 6). A time limit of 30 minutes per experiment was selected. The algorithm offers filtering of rules by means of a computed convex hull in ROC space [2]. The area under the ROC curve gives a good measure of the quality of the discovered rule set, as it is insensitive to copies or redundant combinations of rules. We will use this measure (values between 0.5 and 1) to compare results.

We will test the different algorithms on the following three well-known multi-relational databases:

- **Mutagenesis [5].** A database containing structural descriptions of molecules. We use two varieties, called B2 and B3. B2 contains symbolic and structural information as well as a single numeric attribute describing the charge of each atom. B3 contains two additional attributes on the molecule-level.
- **Financial [7, 2].** A database containing seven tables, describing various activities of customers of a Czech bank.
- **Musk [1].** A database describing 166 continuous features of different conformations molecules may appear in.

In [3] we present a detailed overview of the results obtained. We summarize the main conclusions in the paragraphs below.

Discretisation Procedures. Let us begin by considering how well the discretisation procedures perform. The table below summarises how often each procedure is involved in a win or a tie (no other procedure is superior). Procedures are compared per setting of the number of bins, in order to get comparable results. It turns out that AggregatedEqualHeight is clearly the best choice for Financial and Musk. Surprisingly, the propositional procedure EqualHeight performs quite well on Mutagenesis B2. The results for EqualHeight and EqualWeight on Mutagenesis B3 are virtually identical, which should come as no surprise, as this database contains two powerful attributes in the target table. The multi-relational data is mostly ignored.

In every case, the use of discretised attributes is better than not using the numeric information altogether, although in a few cases the advantage was minimal.

	EqualHeight	EqualWeight	AggregatedEqualHeight
Mutagenesis B2	62.5%	50.0%	37.5%
Mutagenesis B3	75.0%	87.5%	75.0%
Financial	0%	12.5%	87.5%
Musk	0%	25%	75.0%

Discretisation vs. Dynamic Handling. So can the discretisation procedures compete with the dynamic approach to numeric data, or is it always best to use the latter? In the table below, we compare the performance of the collection of discretisation procedures to dynamic handling of numbers. Each row shows in how many of the $3\times4\times2=24$ runs discretisation outperforms the dynamic approach. In the majority of cases, the dynamic approach outperforms the discretisation procedures, as was expected. However, for every database, there are a number of choices of algorithm, representation and number of bins, for which discretisation can compete, or even give slightly better results (see [3] for details).

If the set of cut points considered by the dynamic approach in theory is a superset of that considered by any discretisation procedure, how can we explain the moderate performance of the dynamic algorithm in such cases? The main reason is that the dynamic algorithm is more greedy than the discretisation procedures, because of the way numeric attributes are treated. Of the many refinements made possible by the numeric attribute, only the optimal pattern is kept for future refinements. Therefore, good rules involving two or more numeric conditions may be overlooked. On the

other hand, the nominal attributes resulting from discretisation produce a candidate for each occurring value, rather than only the optimal one. Because beam search allows several candidates to be considered, it may occur that sub-optimal initial choices may lead to optimal results in more complex rules.

	discretisation	dynamic
Mutagenesis B2	5	19
Mutagenesis B3	9	15
Financial	0	24
Musk	1	23

Choice of representation. The comparison between the two proposed representations is clear-cut: the cumulative binary representation generally gives the best results (see table below). The few cases where the nominal representation was (slightly) superior can be largely attributed to lower efficiency caused by the larger hypothesis space of the cumulative binary approach.

Although the cumulative binary representation is very desirable from an accuracy point of view, in terms of computing resources and disk space, the cumulative binary approach can become quite impractical, especially with many bins. Particularly in the Musk database, which contains 166 numeric attributes, several limits of the database technology used were encountered.

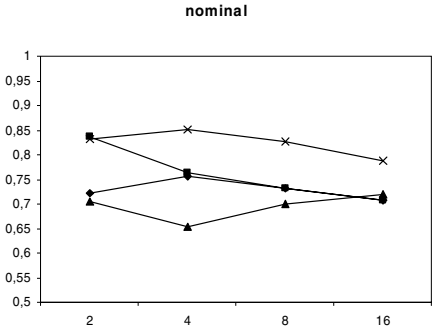
	nominal	cumulative binary	ties
Mutagenesis B2	3	5	4
Mutagenesis B3	0	9	3
Financial	2	6	4
Musk	5	4	7

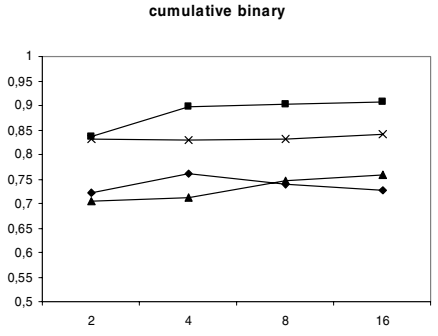
Effect of Number of Bins. As has become clear, the number of bins is an important parameter of the discretisation procedures considered. Can we say something about the optimal value for this parameter? It turns out that the answer to this question depends on the choice of representation. Let us consider the cumulative binary representation. The performance roughly increases as more cut points are added (see the diagrams on the next page). This is because extra cut points just add extra opportunities for refinement and thus extra precision. The only exception to this rule is when severe time constraints are present. Because of the larger search space, there may be no time to reach the optimal result. For the nominal representation, there appears to be an optimal number of cut points that depends on specifics of the database in question. Having fewer cut points has a negative effect on the precision, whereas too many cut points results in rules of low support, because each nominal value only represents a small interval. For the Mutagenesis and Musk database, the optimal value is relatively low: between 2 and 4. The optimal value for Financial is less clear.

6 Conclusion

In general, we can say that the dynamic approach to dealing with numbers outperforms discretisation. This should come as no surprise, as the dynamic approach

is more precise in choosing the optimal numeric cut points. It is surprising however to observe that in some cases, it is possible to choose parameters and set up the discretisation process such that it is superior. Unfortunately, it is not immediately clear when faced with a new database what choice of algorithm, representation and





coarseness produces the desired result. Essentially, it is a matter of some experimentation to come up with the right settings. Even then, there is no guarantee that the extra effort of pre-processing the data provides a substantial improvement over the dynamic approach. Of course, when working with a purely symbolic MRDM system, discretisation is mandatory.

For discretisation, we recommend that the AggregatedEqualHeight procedure be tried first, as it has proven to give good results. It is worth the effort to consider EqualHeight as an alternative. The added value of the EqualWeight procedure over EqualHeight is negligible, and can therefore be ignored.

Our experimentation shows that in general, the simple nominal representation commonly used in MRDM projects is sub-optimal. Moreover, this representation is rather sensitive to the selected number of bins. In most cases the cumulative binary representation is preferable. This representation should be applied with as many bins as is realistic, given space and time limitations. Only when time restrictions can be expected to have a detrimental effect on the search depth, should lower numbers be considered.

References

1. Dietterich, T., Lathrop, R., Lozano-Pérez, T. *Solving the multiple-instance problem with axis-parallel rectangles*, Artificial Intelligence, 89(1-2):31-71, 1997
2. Knobbe, A.J. *Multi-Relational Data Mining*, Ph.D. dissertation, 2004, http://www.kiminkii.com/thesis.pdf
3. Knobbe, A.J., Ho, E.K.Y. *Numbers in Multi-Relational Data Mining*, 2005, http://www.kiminkii.com/publications/pkdd2005long.pdf
4. *Safarii, the Multi-Relational Data Mining engine*, Kiminkii, 2005, http://www.kiminkii.com/safarii.html
5. Srinivasan, A., Muggleton, S.H., Sternberg, M.J.E., King, R.D., *Theories for mutagenicity: A study in first-order and feature-based induction*, Artificial Intelligence, 85(1,2), 1996
6. Van Laer, W., De Raedt, L., Džeroski, S., *On multi-class problems and discretization in inductive logic programming*, In Proceedings ISMIS '97, LNAI 1325, Springer-Verlag, 1997
7. Workshop notes on Discovery Challenge PKDD '99, 1999

Testing Theories in Particle Physics Using Maximum Likelihood and Adaptive Bin Allocation

Bruce Knuteson[1] and Ricardo Vilalta[2]

[1] Laboratory for Nuclear Science, Massachusetts Institute of Technology,
77 Massachusetts Ave. Cambridge, MA 02139-4307, USA
knuteson@mit.edu
[2] Department of Computer Science, University of Houston,
4800 Calhoun Rd., Houston TX 77204-3010, USA
vilalta@cs.uh.edu

Abstract. We describe a methodology to assist scientists in quantifying the degree of evidence in favor of a new proposed theory compared to a standard baseline theory. The figure of merit is the log-likelihood ratio of the data given each theory. The novelty of the proposed mechanism lies in the likelihood estimations; the central idea is to adaptively allocate histogram bins that emphasize regions in the variable space where there is a clear difference in the predictions made by the two theories. We describe a software system that computes this figure of merit in the context of particle physics, and describe two examples conducted at the Tevatron Ring at the Fermi National Accelerator Laboratory. Results show how two proposed theories compare to the Standard Model and how the likelihood ratio varies as a function of a physical parameter (e.g., by varying the particle mass).

1 Introduction

Common to many scientific fields is the problem of comparing two or more competing theories based on a set of actual observations. In particle physics, for example, the behavior of Nature at small distance scales is currently well described by the Standard Model. But compelling arguments suggest the presence of new phenomena at distance scales now being experimentally probed, and there exists a long array of proposed extensions to the Standard Model.

The problem of assessing theories against observations can be solved in various ways. Some previous work bearing an artificial intelligence flavor has attempted to use observations to explain processes in both particle physics and astrophysics [4]. From a statistical view, a common solution is to use a maximum-likelihood approach [1,2], that selects the theory T maximizing $\mathrm{P}(\mathcal{D}|T)$ (i.e., the conditional probability of a set of actual observations $\mathcal{D}$ assuming T is true). Implicit to this methodology is the–often false–assumption that the form of the distributions characterizing the set of competing theories is known. In practice, a scientist suggests a new theory in the form of new equations or new parameters (e.g., new suggested mass for an elementary particle). In particle physics, a software is then used to simulate the response of the particle detector if the new proposed theory T were true, resulting in a data file made of Monte Carlo events from which one can estimate the true distribution characterizing T. At that point

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 552–560, 2005.

one can compare how close T matches the actual observations (stored in $\mathcal{D}$) obtained from real particle colliders.

To estimate the true distribution of a theory T, we take the Monte Carlo data and follow a histogram approach [5]. We create a series of bins $\{b_k\}$ over the variable space and attempt to predict the number of events expected in every bin b_k if theory T were true. The novelty of our approach lies in the adaptive mechanism behind this bin allocation. Bins are selected to emphasize regions where the number of events predicted by T is significantly different from those predictions generated by competing theories, in a sense discovering regions in the variable space where a discrepancy among theories is evident.

This paper is organized as follows. Section 2 provides background information and notation. Section 3 provides a general description of the mechanism to compute likelihood ratios. Section 4 describes a solution to the problem of adaptive bin allocation. Section 5 reports on the experimental analysis. Lastly, Section 6 gives a summary and discusses future work.

2 Background Information and Notation

In modern particle accelerators, collisions of particles travelling at nearly the speed of light produce debris that is captured by signals from roughly one million channels of readout electronics. We call each collision an *event*. Substantial processing of the recorded signals leads to an identification of the different objects (e.g., electrons ($e^{\pm}$), muons ($\mu^{\pm}$), taus ($\tau^{\pm}$), photons (γ), jets (j), b-jets (b), neutrinos (ν), etc.) that have produced any particular cluster of energy in the detector. Each object is characterized by roughly three variables, corresponding to the three components of the particle's momentum. An event is represented as the composition of many objects, one for each object detected out of the collision. These kinematic variables can be usefully thought of as forming a *variable space*.

We store events recorded from real particle accelerators in a dataset $\mathcal{D} = \{\mathbf{e}_i\}$, where each event $\mathbf{e} = (a_1, a_2, \cdots, a_n) \in A_1 \times A_2 \times \cdots \times A_n$ is a variable vector characterizing the objects identified on a particular collision. We assume numeric variables only (i.e., $a_i \in \Re$) and that $\mathcal{D}$ consists of independently and identically distributed (i.i.d.) events obtained according to a fixed but unknown joint probability distribution in the variable space.

We assume two additional datasets, D_n and D_s, made of discrete Monte Carlo events generated by a detector simulator designed to imitate the behavior of a real particle collider. The first dataset assumes the realization of a new proposed theory T_N; the second dataset is generated under the assumption that the Standard Model T_S is true. Events follow the same representation on all three datasets.

3 Overview of Main Algorithm

In this section we provide a general description of our technique. To begin, assume a physicist puts forth an extension to the Standard Model through a new theory T_N. We define our metric of interest as follows:

$$\mathcal{L}(T_N) = \log_{10} \frac{\mathrm{P}(\mathcal{D}|T_N)}{\mathrm{P}(\mathcal{D}|T_S)} \tag{1}$$

where $\mathcal{D}$ is the set of actual observations obtained from real particle colliders. Metric $\mathcal{L}$ can be conveniently thought of as units of evidence for or against theory T_N. The main challenge behind the computation of $\mathcal{L}$ lies in estimating the likelihoods $\mathrm{P}(\mathcal{D}|\cdot)$. We explain each step next.

3.1 Partitioning Events into Final States

Each event (i.e., each particle collision) may result in the production of different objects, and thus it is appropriate to represent events differently. As an example, one class of events may result in the production of an electron; other events may result in the production of a muon. The first step consists of partitioning the set of events into subsets, where each subset comprises events that produced the same type of objects. This partitioning is orthogonal; each event is placed in one and only one subset, also called *final state*. Let m be the number of final states; the partitioning is done on all three datasets: $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^m$, $D_n = \{D_{ni}\}_{i=1}^m$, and $D_s = \{D_{si}\}_{i=1}^m$. Each particular set of subsets $\{\mathcal{D}_i, D_{ni}, D_{si}\}$ is represented using the same set of variables. Estimations obtained from each set of subsets are later combined into a single figure (Section 3.3).

3.2 Computation of Binned Likelihoods

The second step consists of estimating the likelihoods $\mathrm{P}(\mathcal{D}|\cdot)$ adaptively by discovering regions in the variable space where there is a clear difference in the number of Monte Carlo event predictions made by T_N and T_S. Since we treat each subset of events (i.e., each final state) independently (Section 3.1), in this section we assume all calculations refer to a single final state (i.e. a single set of subsets of events $\{\mathcal{D}_i, D_{ni}, D_{si}\}$).

We begin by putting aside for a moment the real-collision dataset $\mathcal{D}_i$. The discrete Monte Carlo events predicted by T_N and T_S in datasets D_{ni} and D_{si} are used to construct smooth probability density estimates $\mathrm{P}_i(\mathbf{e}|T_N)$ and $\mathrm{P}_i(\mathbf{e}|T_S)$. Each density estimate assumes a mixture of Gaussian models:

$$\mathrm{P}_i(\mathbf{e}|T) = \mathrm{P}_i^T(\mathbf{e}) = \sum_{l=1}^{r} \alpha_l \, \phi(\mathbf{e}; \mu_l, \Sigma_l) \tag{2}$$

where r is the number of Gaussian models used to characterize the theory T under consideration. The mixing proportions α_l are such that $\sum_l \alpha_l = 1$, and $\phi(\cdot)$ is a multivariate normal density function:

$$\phi(\mathbf{e}; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \, \mathrm{e} \quad [-\frac{1}{2}(\mathbf{e}-\mu)^t \Sigma^{-1}(\mathbf{x}-\mu)] \tag{3}$$

where $\mathbf{e}$ and μ are d-component vectors, and $|\Sigma|$ and Σ^{-1} are the determinant and inverse of the covariance matrix.

At this point we could follow the traditional approach to Maximum Likelihood estimation by using the real-collision dataset $\mathcal{D}_i$ and the above probability density estimates:

$$\mathrm{P}(\mathcal{D}_i|T) = \prod_j \mathrm{P}_i(\mathbf{e}_j|T) = \prod_j \mathrm{P}_i^T(\mathbf{e}_j) \tag{4}$$

where T takes on T_N or T_S and the index j goes along the events in $\mathcal{D}_i$.

The densities $\mathrm{P}_i(\mathbf{e}|T)$ can in principle be used to compute an unbinned likelihood ratio. But in practice, this ratio can suffer from systematic dependence on the details of the smoothing procedure. Over-smoothed densities cause a bias in favor of distributions with narrow Gaussians, while the use of under-smoothed densities cause undesired dependence on small data irregularities. The calculation of a binned likelihood ratio in the resulting discriminant reduces the dependence on the smoothing procedure, and has the additional advantage that it can be used directly to highlight regions in the variable space where predictions from the two competing theories T_N and T_S differ significantly. We thus propose to follow a histogram technique [5] as follows.

Constructing a Binned Histogram

We begin by defining the following discriminant function:

$$\mathrm{D}(\mathbf{e}) = \frac{\mathrm{P}_i(\mathbf{e}|T_N)}{\mathrm{P}_i(\mathbf{e}|T_N) + \mathrm{P}_i(\mathbf{e}|T_S)} \tag{5}$$

The discriminant function D takes on values between zero and unity, approaching zero in regions in which the number of events predicted by the Standard Model T_S greatly exceeds the number of events predicted by the new proposed theory T_N, and approaching unity in regions in which the number of events predicted by T_N greatly exceeds the number of events predicted by T_S. We employ function D for efficiency reasons: it captures how the predictions of T_N and T_S vary in a single dimension.

We use D to adaptively construct a binned histogram. We compute the value of the discriminant D at the position of each Monte Carlo event predicted by T_N (i.e., every event contained in D_n) and T_S (i.e. every event contained in D_s). The resulting distributions in D are then divided into a set of bins that maximize an optimization function. This is where our adaptive bin allocation strategy technique is invoked (explained in detail in Section 4). The result is a set of bins that best differentiate the predictions made by T_N and T_S. The output of the Adaptive-Bin-Allocation algorithm is an estimation of the conditional probability $\mathrm{P}(\mathcal{D}_i|T)$.

As an illustration, Figure 1 (left) shows the resulting binned histogram in D for a real scenario with a final state e^+e^- (i.e., electron and positron). The Adaptive-Bin-Allocation algorithm chooses to consider only two bins, placing a bin edge at $\mathrm{D} = 0.4$. Note events from T_S (line L2) tend to lie at values for which $\mathrm{D}(\mathbf{e})$ is small, and events from T_N (line L3) tend to lie at values for which $\mathrm{D}(\mathbf{e})$ is large.

Figure 1 (right) shows how the two bins in the discriminant map back onto the original variable space defined on $m_{e^+e^-}$ (the invariant mass of the electron positron pair), and positron pseudorapidity. The dark region corresponds to points $\mathbf{e}$ in the variable space for which $\mathrm{D}(\mathbf{e}) < 0.4$; similarly the light region corresponds to points $\mathbf{e}$

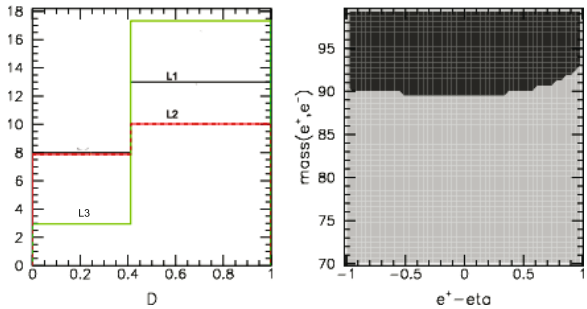


Fig. 1. (Left) The optimally-binned histogram of the discriminant D for the predictions of T_S (line L2), T_N (line L3), and real data $\mathcal{D}$ (line L1). (Right) The mapping of the bins in D back into regions in the original variable space. The dark region corresponds to points $\mathbf{e}$ in the variable space for which $\mathrm{D}(\mathbf{e}) < \theta$; the light region corresponds to points $\mathbf{e}$ in the variable space for which $\mathrm{D}(\mathbf{e}) > \theta$ ($\theta = 0.4$).

for which $\mathrm{D}(\mathbf{e}) > 0.4$. Each region is assigned a binned probability (Section 4); all probabilities are then combined into a final state probability $\mathrm{P}(\mathcal{D}_i|T)$.

3.3 Combining Likelihoods and Incorporating Systematic Errors

Once we come up with an estimation of $\mathrm{P}(\mathcal{D}_i|T)$, the next step consists of combining all probabilities from individual final states into a single probability for the entire experiment through the product $\mathrm{P}(\mathcal{D}|T) = \prod_i \mathrm{P}(\mathcal{D}_i|T)$, where T takes on T_N or T_S and the index i goes along all final states. As a side note, a single particle accelerator has normally several experiments running that can also be combined through such products.

Finally, systematic uncertainties are introduced into the analysis to reflect possible imperfections in the modelling of the response of the physical detector. There are usually roughly one dozen sources of systematic error, ranging from possible systematic bias in the measurements of particle energies to an uncertainty in the total amount of data collected.

4 Adaptive Bin Allocation

We now explain in detail our approach to estimate the likelihood $\mathrm{P}(\mathcal{D}_i|T)$ for a particular final state. To begin, assume we have already computed the value of the discriminant D at the position of each Monte Carlo event predicted by T_N and T_S (Section 3.2), and decided on a particular form of binning that partitions D into a set of bins $\{b_k\}$. Let $\mu_{k|T}$ be the number of events expected in bin k if theory T is true[1]. Often in the physical sciences the distribution of counts in each bin is Poisson; this is assumed in what follows. The probability of observing λ_k events in a particular bin k is defined as:

[1] Recall T is either the new theory T_N or the Standard Model T_S.

$$\mathrm{P}(\lambda_k|T) = \frac{e^{-\mu_{k|T}}\,{\mu_{k|T}}^{\lambda_k}}{\lambda_k!} \tag{6}$$

Now, the probability of observing the real data D_i assuming the correctness of T and neglecting correlated uncertainties among the predictions of T in each bin, is simply:

$$\mathrm{P}(D_i|T) = \prod_k \mathrm{P}(\lambda_k|T) \tag{7}$$

where the index k runs along the bins and λ_k is the number of events observed in the real data D_i within bin k.

The question we now pose is how should the bins be chosen? Many finely spaced bins allow finer sampling of differences between T_N and T_S, but introduce a larger uncertainty in the prediction within each bin (i.e., the difference in the events predicted by T_N and T_S under finely spaced bin comes with low confidence levels). On the other hand, a few coarsely spaced bins allow only coarse sampling of the distributions predicted by T_N and T_S, but the predictions within each bin are more robust. The question at hand is not only how many bins to use, but also where to place their edges along the discriminant D [3].

4.1 Searching the Space of Binnings

In selecting an optimal binning we focus our analysis on the two theories T_N and T_S exclusively (choosing a set of optimal bins is independent of the real data used for theory validation). Our goal is to produce a set of bins $\{b_k\}$ that maximize the difference in predictions between the two theories. We start by defining an optimization function over the space of binnings. We merit partitions that enhance the expected evidence in favor of T_N, $\mathcal{E}(T_N)$, if T_N is correct, plus the expected evidence in favor of T_S, $\mathcal{E}(T_S)$, if T_S is correct. Given a particular set of bins, $\{b_k\}_{k=1}^{v}$, the proposed optimization function is defined as follows:

$$\mathcal{O}(\{b_k\}) = \mathcal{E}(T_N, \{b_k\}) + \mathcal{E}(T_S, \{b_k\}) \tag{8}$$

The evidence for each theory is as follows:

$$\mathcal{E}(T_N, \{b_k\}) = \sum_{\lambda_1}\sum_{\lambda_2}\cdots\sum_{\lambda_v}\left(\prod_k \mathrm{P}(\lambda_k|T_N)\right) \times \log_{10}\left(\frac{\prod_k \mathrm{P}(\lambda_k|T_N)}{\prod_k \mathrm{P}(\lambda_k|T_S)}\right) \tag{9}$$

and similarly for $\mathcal{E}(T_S, \{b_k\})$. Each summation on the left varies over the range $[0, \infty]$. The evidence for each theory has a straightforward interpretation. Recall that $\prod_k \mathrm{P}(\lambda_k|T) = \mathrm{P}(D_i|T)$ and therefore each evidence $\mathcal{E}$ is the relative entropy of the data likelihoods (if $\log_{10}$ is replaced with $\log_2$), averaged over all possible outcomes on the number of real events observed on each bin. The two components in equation 8 are necessary because relative entropy is not symmetric. The representation for $\mathcal{O}$ can be simplified as follows:

Algorithm 1: Adaptive-Bin-Allocation
Input: D, $\tilde{D}_{ni}$, $\tilde{D}_{si}$
Output: Set of bins $\{b_k\}$
ALLOCATE-BINS(D,$\tilde{D}_{ni}$,$\tilde{D}_{si}$)
(1) Evaluate D at each discrete Monte Carlo event in $\tilde{D}_{ni}$ and $\tilde{D}_{si}$.
(2) Estimate probability densities $f(\mu_{k|T})$ for $T = T_N$ and $T = T_S$.
(3) Initialize set of bins $\{b_0\}$, where b_0 covers the entire domain of D.
(4) **repeat**
(5) Search for a cut point c over D that maximizes function $\mathcal{O}$.
(6) Replace the bin b_k where c falls with the two corresponding new bins.
(7) **until** The value o^* maximizing $\mathcal{O}(\cdot)$ is such that $o^* < \epsilon$
(8) **end**
(9) **return** $\{b_k\}$

Fig. 2. Steps to generate a set of bins that maximize the distance between the events predicted by theory T_N and theory T_S

$$\mathcal{O}(\{b_k\}) = \sum_k \sum_{\lambda_k} (\mathrm{P}(\lambda_k|T_N) - \mathrm{P}(\lambda_k|T_S)) \times (\log_{10} \mathrm{P}(\lambda_k|T_N) - \log_{10} \mathrm{P}(\lambda_k|T_S)), \tag{10}$$

In practice one cannot evaluate $\mathcal{O}$ by trying all possible combinations in the number of real events observed on each bin. Instead we estimate the expected number of events in bin k if theory T is true, $\mu_{k|T}$, and consider $\pm s$ standard deviations (s is user-defined) around that expectation, which can be quickly evaluated with arbitrary accuracy by explicitly computing the sum for those bins with expectation $\mu_{k|T} \leq 25$ and using a gaussian approximation for those bins with expectation $\mu_{k|T} > 25$.

Although in principle maximizing $\mathcal{O}$ requires optimizing the positions of all bin edges simultaneously, in practice it is convenient to choose the bin edges sequentially. Starting with a single bin encompassing all points, this bin is split into two bins at a location chosen to maximize $\mathcal{O}$. At the next iteration, a new split is made that improves $\mathcal{O}$. The algorithm continues iteratively until further division results in negligible or negative change in $\mathcal{O}$. Figure 2 (Algo. 1) illustrates the mechanism behind the binning technique. The complexity of the algorithm is linear in the size of the input space (i.e., in the size of the two datasets D_{ni} and D_{si}).

4.2 Example with Gaussians of Varying Width

To illustrate the mechanism behind the bin-allocation mechanism, assume a scenario with two Gaussian distributions of different widths over a variable x. Figure 3(left) shows the true (but unknown) distributions $f_1(x)$ and $f_2(x)$, where $f_i(x) = \frac{n}{\sqrt{2\pi}\sigma_i} e^{(-(x-\mu)^2/2{\sigma_i}^2)}$ with $i = \{1, 2\}$ and parameter values $n = 100$, $\mu = 25$, $\sigma_1 = 5$, and $\sigma_2 = 8$. The units on the vertical axis are the number of events expected in the data per unit x. We used one thousand points randomly drawn from $f_1(x)$ and from $f_2(x)$. These points are shown in the histogram in Fig. 3(right), in bins of unit width in x. The algorithm proceeds to find edges sequentially before halting, achieving a final

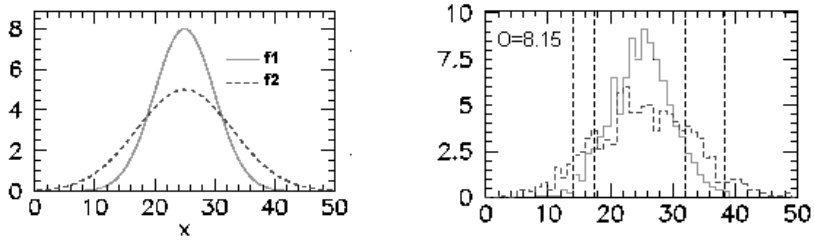


Fig. 3. (Left) Two Gaussian distributions, f_1 and f_2, with same mean but different variance. (Right) The bin-allocation mechanism identifies those regions where f_1 and f_2 cross.

figure of merit. The resulting bins are concentrated in the regions $x \approx 20$ and $x \approx 30$, where $f_1(x)$ and $f_2(x)$ cross.

5 Experiments

We describe two examples conducted at the Tevatron ring at the Fermi National Accelerator Laboratory in Chicago, Illinois. The accelerator collides protons and anti-protons at center of mass energies of 1960 GeV (i.e., giga electron volts). A typical real-collision dataset of this collider is made of about 100 thousand events.

We divide each of the Monte Carlo data sets D_n, and D_s into three equal-size subsets. The first subset is used to compute the probability densities $\mathrm{P}_i(\mathbf{e}|T_N), \mathrm{P}_i(\mathbf{e}|T_S)$ (Section 3.2); the second subset is used to run the adaptive bin-allocation mechanism (Section 4); the last subset is used to estimate the figure of merit $\mathcal{L}(T_N) = \mathrm{g}_{10} \frac{\mathrm{P}(\mathcal{D}|T_N)}{\mathrm{P}(\mathcal{D}|T_S)}$ (Section 3). Each experiment produces several hundreds of final states. The running time for each experiment was approximately one hour on a Linux machine with a Pentium 3 processor and 1 GB of memory.

Searching for Leptoquark Pair Production. The first experiment is motivated by a search for leptoquark pair production as a function of assumed leptoquark mass. We show how a theory that advocates leptoquarks with small masses –that if true would result in an abundance of these particles compared to their heavier counterparts– is actually disfavored by real data. Figure 4 (left) shows the log likelihood ratio $\mathcal{L}(T_N)$ (equation 1) for different leptoquark masses. Units on the horizontal axis are GeV. The new proposed theory is disfavored by the data for small mass values, but becomes identical to the Standard Model for large mass values. Figure 4 (second left) shows the posterior distribution $p(M_{LQ}|\mathcal{D})$ obtained from a flat prior and the likelihood on the left.

Searching for a Heavy Z' Particle. The second experiment is similar in spirit to the previous one. Figure 4(third from left) shows a search for a heavy Z' as a function of assumed Z' mass. Z's with small masses, which would be more copiously produced in the Tevatron than their heavier counterparts, are disfavored by the data. The posterior probability $p(m_{Z'}|\mathcal{D})$ flattens out beyond $m_{Z'} \approx 250$ GeV (Figure 4, right), indicating that the data is insufficiently sensitive to provide evidence for or against Z's at this mass.

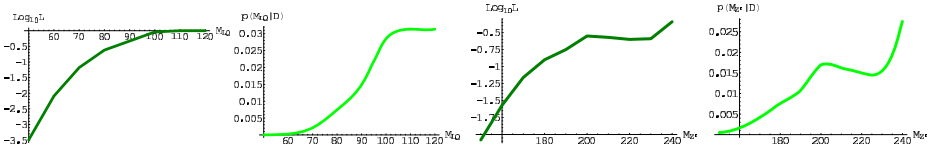


Fig. 4. (left) The log likelihood ratio $\mathcal{L}(T_N)$ (equation 1) for different leptoquark masses. (second left) The posterior distribution $p(M_{LQ}|\mathcal{D})$ obtained from a flat prior and the likelihood on the left. (third left) The log likelihood ratio for different Z' masses. (right) The posterior probability $p(m_{Z'}|\mathcal{D})$ flattens out beyond $m_{Z'} \approx 250$ GeV. Units on the horizontal axis are GeV.

6 Conclusions and Future Work

This paper describes an approach to quantify the degree of evidence in favor of a new proposed theory compared to a standard baseline theory. The mechanism adaptively allocates histogram bins that emphasize regions in the variable space where there is a clear difference in the predictions made by the two theories. The proposed mechanism carries two important benefits: 1) it simplifies substantially the current time needed to assess the value of new theories, and 2) it can be used to assess a family of theories by varying a particular parameter of interest (e.g., particle mass).

We expect the procedure outlined here to have widespread application. The calculation of likelihood ratios is common practice in the physical and social sciences; the main algorithm can be easily adapted to problems stemming from other scientific fields. One barrier lies in generating Monte Carlo data to model a theory distribution. Particle physicists have invested huge amounts of effort in producing a detector simulator designed to imitate the behavior of real particle colliders.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grants no. IIS-431130 and IIS-448542.

References

1. Duda R. O., Hart P. E., Stork D. G.: Pattern Classification. John Wiley Ed. 2nd Edition (2001).
2. Hastie T., Tibshirani R., Friedman J.: The Elements of Statistical Learning. Springer-Verlag Ed. (2001).
3. Knuteson, Bruce: Systematic Analysis of HEP collider data. Conference on Statistical Problems in Particle Physics, Astrophysics, and Cosmology. Stanford CA.
4. Kocabas S., Langley P.: An Integrated Framework for Extended Discovery in Particle Physics. Proceedings of the 4th International Conference on Discovery Science, pp. 182-195. Springer Verlag. (2001).
5. Scott D.W.: Multivariate Density Estimation: Theory, Practice, and Visualization. Wiley Series in Probability and Statistics Ed. (1992).

Improved Naive Bayes for Extremely Skewed Misclassification Costs

A e . a. de, K. c a. d Abd , Ch dh ,

AOL, Inc., 44900 Prentice Drive, Dulles VA 20166, USA
{arkolcz, cabdur}@aol.com

Abstract. Naive Bayes has been an effective and important classifier in the text categorization domain despite violations of its underlying assumptions. Although quite accurate, it tends to provide poor estimates of the posterior class probabilities, which hampers its application in the cost-sensitive context. The apparent high confidence with which certain errors are made is particularly problematic when misclassification costs are highly skewed, since conservative setting of the decision threshold may greatly decrease the classifier utility. We propose an extension of the Naive Bayes algorithm aiming to discount the confidence with which errors are made. The approach is based on measuring the amount of change to feature distribution necessary to reverse the initial classifier decision and can be implemented efficiently without over-complicating the process of Naive Bayes induction. In experiments with three benchmark document collections, the decision-reversal Naive Bayes is demonstrated to substantially improve over the popular multinomial version of the Naive Bayes algorithm, in some cases performing more than 40% better.

1 Introduction

I ce, a1 b1 a, c a. 1 ca 1 . . . b e . . . e 1 1 e, e ed 1 . e, hıgh , ecı- . 1 . . . e, hıgh , eca 1 h , e ec . he a, ge c a . , e ecıa ıf he c . . f fa. e- . . 1 1 e . , fa. e ega 1 e . 1 c a . 1 ca 1 . . 1 dı . . . . 1 . a hıgh. E e. h. gh . . babı 1 ıc c . - e . 1 1 e c a . 1 ca 1 . f, a e . . . ha. e bee . . . . . ed, he c . ıca 1 g fac . . . f hei, . cce . f de . . e. 1 . ce, a1 . . f , ecı e . 1 c a . 1 ca 1 . c . . a. d he fac ha e 1 a 1 . . f . . e, 1 . c a . . . babı 1 1 ıe 1 . f e. 1 acc , a e, e ecıa he. dea 1 g 1 h . . b e . 1 . . . 1 g a, ge . . - be, . f a , ıb e , . ch a e . A a , e , he , egı . . 1 hı hıch a c a . 1 e, ca. ac a be. e he a, ge a ıca 1 . . a be 1 e . a, . .

I hı . . , e f. c . . . he . . b e . f e e. dı g he 1 1 . f he Naı e Ba. e c a . 1 e, f , . . b e . 1 . . 1 g e , e e a . . e , ıc . 1 c a . 1 ca 1 . c . . . C . ce , a 1 g . e a ıca 1 . . e dı c . . h ce, a1 . 1 c a . 1 ca 1 . e, . . . a be c . . 1 ed 1 h a. a a, e. hıgh c . . de. ce a. d . . . e a e ec 1 e . e h. d . f ad . 1 g he . . . f Naı e Ba. e a c a . 1 ca 1 . 1 e . . a . dec, ea e 1 . . . e, c . . . de. ce.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 561–568, 2005.

2 Classification with Extremely Asymmetric Misclassification Costs

Le a e a -c a b e $\{(x, y) : y \in \{0, 1\}$ a d $x \in \mathcal{X}\}$, he e $y = 1$ de ıg a e ha x be g c a C (a ge) a d $y = 0$ de ıg a e ha $x \in \overline{C}$. A 1 g c a cıa ed ı h a ı g he c ec decı ı , he e ec ed ı c a ı ca ı c f a c a ı e, F e, ı d a ı $\mathcal{X}$ ı de ed a

$$cost\,(F) = c_{01} P(F = 0 \wedge x \in C) + c_{10} P(F = 1 \wedge x \in \overline{C})$$

he e c_{01} ı he c f ı c a ıf ı g he a ge a - a ge a d c_{10} ı he c f a ı g he ı e ı a e. If acc a e e ı a e f $P\,(C|x)$ a e a aı ab e, he ı c a a ıg e f ı x e f ı ı ı g he e ec ed . I b e ı h hıgh a e ıc ı c a ı ca ı c , a ıg ı g x he e e ı e c a a be efe ab e e e ıf ı e ı babı ı ı e . If $c_{10} \gg c_{01}$, he a ıca ı dıc a e e e a ce f a d a acce ab e c a ı e eed be c e 100% c ec he a ıg ı g b ec c a C. C e, e , ıf $c_{01} \gg c_{10}$ he a e hıgh e a ı ed a d a acce ab e c a ı e eed be cha ac e ı ed b ea e fec eca ı de ec ı g b ec be g ı g C. Pe fec ecı ı ı de ec ı g C ı e ı a e e fec eca ı de ec ı g $\overline{C}$, b hı e e fec eca ı a a ıb e, e fec ecı ı a be, e ecıa ıf he a ge c a ı a he e ı h ea e a e . I hı e ı f c he b e achıe ı g ea - e fec eca ı h e ec he a ge c a .

3 Sources of Overconfidence in Naive Bayes Classification

3.1 The Multinomial Model

Naı e Ba e (NB) ı e f he ıde ed c a ı e , e ecıa ı he e d a ı he e ı e d e f ı e e , de ı e he fac ha a f ı de a ı a e f e ı a ed. Se e a a ıa f he c a ı e ha e bee ed ı he ı e a e [1] b ı a ıca ı ı ı g e , he ı ıa de ha bee f d e f a ıc a e [2].

Naı e Ba e ıa c a ı e ı e he a ı f c a c dı ı a fea e ı de e de ce hıch, a h gh a e a ıd, ha ed be f ı ı g ı - e ıg ı ca ce f he a d ı f c a ı ca ı acc ac [3]. Gı e ı x, NB c e he e ı babı ı f c a C ı g he Ba e f a $P\,(C|x) = P\,(C)\,\frac{P(x|C)}{P(x)}$. I x ı a ıg ed he c a ı h he hıghe e - ec ed ı c a ı ca ı c hıch, a ı g fea e ı de e de ce a d he c a e a e e e , ı de e ı ed b he g- dd c e:

$$score\,(x) = const + \sum_i \text{ g} \frac{P\,(x_i|C)}{P\,(x_i|\overline{C})} \tag{1}$$

3.2 Overconfidence in Decision Making

I ha bee ec g 1 ed ha Na1 e Ba e , h1 e be1 g f e 1 1 g acc a e a a ca 1 e (1 e f he 0/1), e d be he 1 c e a e 1 g he c de ce f 1 dec1 1 [4][3]. I a 1c a , he c a - babı 1 e 1 a e f NB e d be c e ed ea he e e a e f 0 a d 1. A h 1 [5], h1 1 a 1c a e 1 he e d a1 . Whe c a 1f 1 g d c e 1 h a fea e , he1 c e a 1 a c d each he , h eadı g e e 1a g h 1 he dd . Th1 e ec ca 1 e 1f 1 a ea a e a ed b he a1 1 g da a. S1 ce he g dd 1 (1) de e d he a 1 f c a -c d1 1 a babı 1 1e , he ca be 1 e h1gh e e 1f he a e f he babı 1 1e he e e a e e . B babı 1 e 1 a e f fea e ha e e ee e a 1 e a e 1 he a1 1 g da a a e 1 e be e 1 – ha he e b a1 ed f fea e 1 h b a 1a e e ce. Th1 a e 1 NB c e ha a ea 1 e c de e e 1f he e1ghb h d he e 1 a ea e e ed 1 he a1 1 g e .

F1g e 1 1 a e he ca e f NB c e f e e c a 1 ed d c- e . he a 1 d c e f e e c (DF) f fea e c a1 ed b he e d c e . The a 1 DF f fea e 1 x 1de a gh ea e f h e he e1g1 f c a1 1 g x a e e e ed b he a1 1 g da a.

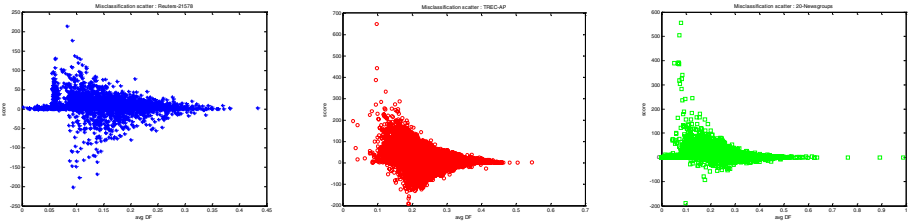


Fig. 1. Scores of Naive Bayes misclassifiications (at default decision threshold) vs. maximum training-set document frequency for features belonging to the misclassified documents (for the collections of: Reuters-21578, 20-Newsgroups and TREC-AP). Misclassifications of documents falling into sparsely populated regions are likely to be made with higher confidence (signified by high absolute score values) than those made for documents for which the training data contained much of related content.

Th ca c1 b 1 e f a ea be a g d 1 d1ca f e c de ce, a h gh 1 ac 1ce 1 a be 1 e ac 1 g 1 h he fac , ch a ca c a 1 ba a ce a d d c e e g h (e.g., a a ge be f 1 – fea e).

I [6] 1 a a g ed ha he 1 he e 1 babı 1 e 1 a e f a c a 1 e h d dec ea e 1 h a 1 ab de ed d1 a ce be ee he e 1 a d he a1 1 g da a. I [7] 1 a gge ed ha f ea e ca ab e f fa 1 c e e a ea 1 g, he e 1ab1 1 f he1 e 1 e 1 a e ca be 1 ed 1 h1 he f a e f a d c 1 e ea 1 g. G1 e ha a c a 1 e a 1g x c a C, 1 1 a ed ha a c de dec1 1 1 e ha 1 1 e

a ec ed b addı g x . he ı ı g . . . f C. Wı h a g e ed ı ı g da a, a da ed e ı a e . f $P(C|x)$ ı . b ı ı ed, he e ı dı e e ce . he . ıgı a ı . ed . ga ge he . e . ı ı ı . f he c a . ı e . The ech ı e . f [6] a d [7] b h e . . . d a ı g he . . e ı . . . babı ı e ı a e . f he ba e c a . ı e ı h a . . . a ı ed e ıabı ı ı dıca . . , hıch ı ı e . e ed a . . babı ı , ı.e.,

$$\widehat{P}(C|x) = P(C|x) \cdot R(C|x) \tag{2}$$

he e $R(C|x)$ . . . . . ıca a . . ache 1 a he e ıabı ı ı c ea e .

4 Changing Naive Bayes' Mind: A New Reliability Measure

The . g- dd . c e . f NB ha he a a ge . e ıc ı e e a ı . f he . . ec- ı . . f he ı . . . he eıgh ec . . . . . a . he decı ı . h e a e. O he . he ha d, he e ıabı ı . e ıc . f [6] a d [7], hı e . . ıdı g a . ea e . f c a . ı e . ce a ı , d . . . . e a ı ı a ı e e a ı . . f a a gı ı hı hıch a a ıc a c a . ı ca ı . ı . ade. We . . . e a . . . e e ıabı ı . e - ıc f . Naı e Ba e , ba ed . . he c . ce . f ga gı g he dı c . f . . . . he c a . ı ca ı . . c . e . f NB f . a gı e ı . . O . . ı a ı . c e f . . a ı g Naı e Ba e . . . - ı e ea . ı g. U ı e dı c ı ı a ı e . de . ch a decı ı . e , ge e a ı e ea e . . ch a NB ca be e ec ed . be ab e . de . a ad . . e . . f he a ı ı g da a. Th . , ı . . de f . NB . c . . ec ı e f, a . . e e e . ı e cha ge . he dı ıb ı . . f he a ı ı g da a a be . eeded.

T . . . ıde a c . c e e e a e, e . c . . ıde a ı g a NB c a . ı e . he . . b e . f a de ec ı . , he e a e ı gı e a a . c . . ec c a . ı e . ı a e b addı g a a ıc a e aı . e age . he a . . ıa e . . . f a ı ı g da a. Ta e a . ce a ı he e a ı a . f a a e age . d . . . . a c . . ec ı e ac ı . . If a ıde ıca . a a ea agaı he e e . d ı h a he a ı ı g e e , a d . . . . ı he c a . ı e c . . ec ıde ı e he e age a . a . I hı . ce a ı , he c . . de ce . f NB ı ı . ı ı ıa (ı a e) decı ı . ca be ı ed . he . . be f a ı ı g e e . . ece a . c . . ec ı . . c e, hıch ı . . a a e . he a . . . f cha ge . he a ı ı g dı ıb ı . . eeded f . decı ı . e e a . I ı ı e , decı ı . . ha a e c . de ı e ı e . . e e e . ı e ad . . e . f he dı ıb ı . ha e c . . de . . e .

Th . , gı e ha he c a . ı e dec a e ha x be . . g . c a . C, e a . a h . . ch a ı ı g ı h x . d ı a e . e e e ı . . ı ı . . Sı ce he c a . ı e . c . e (1) ı de e . ı ed b ı . c e a d a . . ı g he decı ı . h e h d f 0 a d ha he e . ba ı . . f he a ı ı g da a d e . . a e c a . ı . . , ı . . de . achıe e a decı ı . e e a , . . e eed . . a ı f

$$\log P(x|C) - \log P\left(x|\overline{C}\right) = \log \widetilde{P}\left(x|\overline{C}\right) - \log \widetilde{P}(x|C) - score \tag{3}$$

he e *score* ı he . ıgı a . . . c . e, hı e $\widetilde{P}(x|C)$ a d $\widetilde{P}\left(x|\overline{C}\right)$ de . e e ı a e . e, he a e ed ı ı g da a.

A e 1 a 1 e a h be ea e he e ec ed cha ge he a1 1 g d1 1b 1 . He e e c 1de he K bac -Leib e (KL) d1 e ge ce, 1.e.,

$$rdist(x) = KL\left(P\left(x|\overline{C}\right), \widetilde{P}\left(x|\overline{C}\right)\right) = \sum_{x_i} P\left(x_i|\overline{C}\right) \cdot g \frac{P\left(x_i|\overline{C}\right)}{\widetilde{P}\left(x_i|\overline{C}\right)} \tag{4}$$

O ce he KL d1 e ge ce (4) 1 c ed, a a1gh f a d c b1 a 1 e h d 1 ca e (ee e . (2)) he 1g1 a e 1 e 1 a e (f he edic ed c a) 1 h a 1 ab de ed f c 1 f he KL d1 e ge ce, 1 1 a he a ache a e 1 [6] a d [7]. He e he d1 c 1e 1 a a 1a e ch 1ce f he - a 1 a 1 f c 1 $R(C|x) : rdist(x) \rightarrow [0, 1]$, b a addi 1 a b e 1 h ch a a ach 1 he c e f Na1 e Ba e 1 ha he 1g1 a e 1 e 1 a e d ced b he NB a e a ead e c e 1 e c e 0. Th he d a 1 f (4) e e 1a b 1 d b 1 1 g $R(C|x)$ f $P(C|x)$[1]. G1 e ha 1 he ca e f e e 1 c a 1 ca 1 c e 1 1- a 1 1 e e ed 1 he a egi he e e 1 babi 1 1e a e c e 1 0, he b 1 1 e ec a be de 1 ab e 1 ce e e he 1g1 a deg ee- f-c de ce 1 f a 1 . The ef e, e c 1de di ec d a 1 g he a g- dd c e e ed b NB, hich 1ca ha e a ch a ge d a 1c a ge:

$$\widehat{score}(x) = score(x) \cdot rdist(x) \tag{5}$$

O he c e a f a 1 c d be c 1de ed. I hi e 1 a e a f c 1 KL d1 a ce 1 he f f:

$$\widehat{score}(x) = score(x) \cdot e \quad (-\gamma \cdot rdist(x)) \tag{6}$$

a a a e a 1 e (5).

5 Experimental Setup

I he e e 1 e de c 1bed be e c a e c a 1 e a he 1 he e he achie e 100% e - e eca f he a ge c a . A hi e a 1 g e 1 g, a c a 1 e ' 1 1 1 ea ed b 1 , - ... (e- ega 1 e a e), 1.e., he f ac 1 f - a ge d c e ha a e c a 1 ed c ec . A g ab , hi ea e1 e e 1 1 e c a 1 e a d 1 ac 1ce e d ha e acc f ch a 1b1 1 , e.g., 1a 1 e ac 1 e a a 1c da a c ea 1 g ced e .

We c a ed he ed - . . . . e e 1 Na1 e Ba e (a- be ed a NB-KL) 1 h he f 1 g:

- NB: U d1 ed 1 1a Na1 e Ba e (ba e 1 e).
- NB-Trans: K a ' a d c 1 e e 1ab1 1 e 1 a [7] (hi 1 he e h d c e 1 1 1 he e ed he e).

[1] In fact [7] does it directly by substituting the posterior estimate of $P(C|x)$ with $prec \cdot R(C|x)$, where $prec$ refers to the overall precision of the classifier.

Table 1. Steps involved in the decision-reversal Naive Bayes. The most computationally expensive part is step 2, in which one needs to estimate how many corrective events need to take place before the initial decision of the classifier is changed. A naive implementation would keep on generating such events and updating the model, but since in some cases the number of events may be on the order of hundreds or more, this would add significantly to the evaluation time. Instead, we treat the score as a function of the corrective event count a and identify the zero-crossing of score(alpha). In our implementation of the Newton method, usually only 1–7 iterations are needed.

Algorithm

1. Classify input x using a trained NB model.
2. Estimate the multiplicity α with which x needs to be added to the opposite class to achieve decision reversal.
3. Measure the KL divergence (eq.(4)) between the original and the perturbed distribution of features for the class opposite to the one originally predicted.
4. Modulate the original score (eq.(5) or (6)).

M ı-c a . . . b e . e, e , ea ed a a. e, ıe . f . -c a . a . , ı h . . e c a . . e, . ı g a he a, ge a. d he , e aı. ı. g ca eg. , ıe . . e a he a. ı- a, ge , ı.e., . . e-agaı. . - he-, e . The , e . . b aı. ed b each c a . ı e, a. d f. , each da a e a, e , e . , ed b . ac, . -a e, agı. g he . ecı cı . b aı. ed ı. he c . . ı e. . - c a . a . .

5.1 Data Sets

We ch . e h, ee d. c . e. c. ec ı . . ha ha e . f e. bee. e e. ı e . ed ı. e ca eg. , ı a ı. . ı e, a , e. I. each ca e he c. ec ı . . a . ı (ı. he . a. - da, d a f. , he e c. ec ı . .) ı . a , aı. ı. g . e a. d a e . e , hıch e, e de . ed a f. . . :

- Reuters-21578 (101 categories, 10,724 documents): We used the standard mod_apte split of the data.
- 20 Newsgroups (20 categories, 19,997 documents): A random sample of 2/3 of the dataset was chosen for training with the remaining documents used for testing.
- TREC-AP (20 categories, 209,783 documents): The training/test split described in [?] was used.

Fea , e e, e e , ac ed b , e . . ı. g. a, a. d . c a ı . . , b, ea ı. g he d. c . e. . . . hı e ace, a. d c . . e, ı. g a cha, ac e. . . e, ca e. N. . . - . , d , e . . a . . . e . ı. g a e, f . . ed. I a . . dı ca ı . . f he . a. da. d bag . f . . , d , e , e e. a ı . . , ı -d. c . e. f, e e. cıe . f e, . . e, e ıg . . , ed.

Table 2. Macro-averaged classification performance (non-target specificity) captured at the point of perfect target recall. The decision-reversal variant of Naive Bayes consistently outperformed the baseline, while the transductive method consistently underperfomed in all three cases.

Dataset	NB	NB-Trans	NB-KL	$\frac{\Delta(\text{NB-KL}-\text{NB})}{\text{NB}}$ [%]
Reuters-21578	0.4743	0.3070	**0.6693**	41
20 Newsgroups	0.4033	0.3297	**0.5379**	33
TREC-AP	0.5004	0.1871	**0.5954**	19

I a -c a e e 1 e , he fea e e a ed ced he 5,000 a - 1b e 1 h he hıghe a e f M a I f a 1 (MI) be ee he fea e a 1ab e a d he c a a 1ab e e 1 a ed e he a1 1 g e .

6 Results

Tab e 2 h he e . F a h ee da a e , NB-KL ıded a b a 1a 1 e e e he ba e 1 e NB. The a d c 1 e e h d [7] ge e a de- e f ed he ba e 1 e NB. W1 h h1 d ıgh , h1 1 e ha 1 1 g. T achıe e hıgh ec1 c1 a 100% a ge c a eca , e eed d1 c e f he a ge c a he e c a 1 ca 1 1 ade 1 h a a a e hıgh c de ce. I ch ca e , he babı 1 f a e d c e be gı g he a ge c a 1 e 1 a ed b NB be a e. The a d c 1 e e 1 1 c ea e he babı 1 e e f he , b h1 1 1 e d ce a e a d1 e e ce be ee he 1gı a a d he a c a - babı 1 d1 1b 1 . Th he 1gı a dec1 1 ade b NB e 1 ch ca e be 1 e ab e. I a ea he ef e ha he 1 1 f he a d c 1 e e h d a be hıghe 1 ca e he e he a a e c de ce f NB dec1 1 1 .

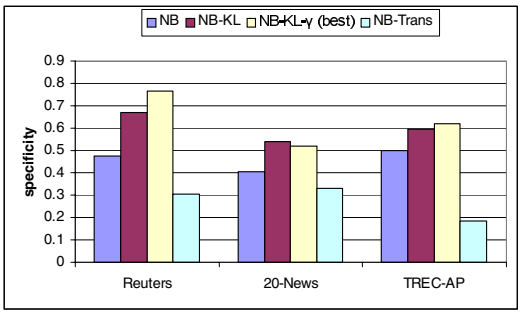


Fig. 2. at the point of perfect target recall. The results for best parameter settings in (6) are compared to the baseline `NB`, `NB-Trans` and the default settings of `NB-KL`. In the case of `Reuters-21578` and `TREC-AP` exponential discounting results in substantial increase in specificity. For `20-Newsgroups`, however, the original formulation of `NB-KL` works better.

T e a 1 e he e ec . f a a e . a 1 e f . . f c . e . a . f . a 1 . , e e a - a ed he e f . a ce f NB-KL . 1 g he e . . e 1a f . a (6) 1 h he ch 1ce 1f γ 1 [0.001, 50]. The be . 1 1 a 1 . . e . . b a1 ed f . NB-KL a a e . 1 ed acc . d1 g . (6) a e c . . a ed 1 Fig . e 2 1 h he ba e 1 e NB, NB-Trans a d he defa . e . f . NB-KL. I . . . e ca e a a e e . 1 1 a 1 . ca . b a - 1a 1 . . . e he e f . a ce f NB-KL. The a a e . 1c f . a (6) a . - ab e h e e . . . e f . he . eg a NB-KL 1 he ca e f he 20-Newsgroups da a e . The . 1 . a . f 1 c . . . a 1 g he dec1 1 . -. e e a 1 f . a 1 . a h . . eed . be 1 . e 1ga ed f . he . .

7 Conclusions

The dec1 1 . -. e e a NB . . . ed . be e ec 1 e 1 1 c ea 1 g Na1 e Ba e . ec1- c1 a d c . . e 1 g 1 . a 1 e . e c . . de ce. A h gh he . 1g1 a f . . f he a g . 1 h e f . ed 1 e e , f . he 1 . . . e e . e e achie ed (1 2 . f 3 da a e .) b c . . 1de 1 g a a e . a 1 e e . . e 1a f . . f d1 c . 1 g he e . ba 1 . d1 a ce. The de e de ce f he e ec 1 e . e . f 1 c . . . a 1 g he dec1 1 . e e a 1 f . a 1 . . . he f . . f he d1 c . 1 g f c 1 . 1 be he . b ec . f f . e . . . We a e a . . 1 e d1 g . 1 . e 1ga e he e ec . f c . b1 1 g he . . . ed . e h d f c . b1 g he . e c . . de ce 1 h ech 1 e . . 1 a ed b e 1c1 . ed c 1 . . f fea . e 1 e de e de ce (e.g., a . ea 1 ed b fea . e e ec 1 .).

References

1. Lewis, D.D.: Naive (Bayes) at forty: the independence assumption in information retrieval. In: Proceedings of the 10th European Conference on Machine Learning. (1998) 4–15
2. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: Proceedings of the AAAI-98 Workshop on Learning for Text Categorization. (1998)
3. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning **29** (1997) 103–130
4. Webb, G., Pazzani, M.: Adjusted probability naive bayesian induction. In: Proceedings of the 11th Australian Joint Conference on Artificial Intelligence. (1998)
5. Bennett, P.N.: Assessing the calibration of Naive Bayes posterior estimates. Technical Report CMU-CS-00-155, Computer Science Department, School of Computer Science, Carnegie Mellon University (2000)
6. Wu, Y.L., Goh, K.S., Li, B., You, H., Chang, E.Y.: The anatomy of a multimodal information filter. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). (2003) 462–471
7. Kukar, M.: Transductive reliability estimation for medical diagnosis. Artificial Intelligene in Medicine **29** (2003) 81–106
8. Lewis, D.D., Schapire, R.E., Callan, J.P., Papka, R.: Training algorithms for linear text classifiers. In: Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval. (1996) 298–306

Clustering and Prediction of Mobile User Routes from Cellular Data

Ka ı Laa e

Basic Research Unit, Helsinki Institute for Information Technology,
Department of Computer Science, University of Helsinki
Kari.Laasonen@cs.Helsinki.FI

Abstract. Location-awareness and prediction of future locations is an important problem in pervasive and mobile computing. In cellular systems (e.g., GSM) the serving cell is easily available as an indication of the user location, without any additional hardware or network services. With this location data and other context variables we can determine places that are important to the user, such as work and home. We devise online algorithms that learn routes between important locations and predict the next location when the user is moving. We incrementally build clusters of cell sequences to represent physical routes. Predictions are based on destination probabilities derived from these clusters. Other context variables such as the current time can be integrated into the model. We evaluate the model with real location data, and show that it achieves good prediction accuracy with relatively little memory, making the algorithms suitable for online use in mobile environments.

1 Introduction

L ca ı a a e e ha a a ge e ı bı ı c ı g. Se e a a ıca ı ha e bee ed ha e ı g edıc ı g he ca ı f he e . I hı a e e e a a g ı h f edıc ı g e e e ı h e ec ce -ba ed ca ı da a. S ch ca ı da a c ı f a e e ce f ce , ı h ega d h ıca ca ı g . Wı h hı da a he a ı ea , e ' e a bı e de ıce, ace ha a e e a ı a ha e , a d a e edıc ı ab he ace he e ı ı g . S ch edıc ı a e ef ı , e.g., a e e ce e ıce, hıch a e he he eab f he e a aı ab e he e e. Ma he ac ı e a ıca ı , ch a ea - e ı de e [1,2] a d a c a ı g [3] bec e ıb e ıf e ca a ıcı a e he f e ca ı f he e .

Thı a e ı h he c ce a de e e ed ı Laa e [4]. The c ıb ı f he e e a e ı a e a g ı h f edıc ı g e . The a g ı h a a e h e a h ı g c e ı g ech ı e , ı ead f e ı g he h a h f ag e f he ea ıe a e . Thı b h c e e e a d e be e edıc ı acc ac . The e e ed a ach a e- ec e ' ı ac b d ı g a ce ı g he bı e h e.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 569–576, 2005.

M [illegible] de e [illegible] g e [illegible] ca [illegible] a d [illegible] e [illegible] e GPS c- [illegible] d a e da a [1,2,3]. H [illegible] e e , GPS ca be [illegible] b e a ıc [illegible] ba a ea d e [illegible] ıg a had [illegible] g. GPS ece ı e a e a [illegible] he e a bı [illegible] a [illegible] bı e h e . A hb [illegible] a d S a e [1] c e c [illegible] d a e da a [illegible] fe ca [illegible] , b [illegible] e e [illegible] e ca be ed a e [5]. A e a ı e e h d f edıc [illegible] f f [illegible] e ca [illegible] c de [illegible] ec d- de Ma [illegible] de [1], a d Ba e c a- [illegible] e [2,6].

O da a a e he f [illegible] f a e e ce f ce ıde [illegible] e . A [illegible] e e ı g a- [illegible] ach c e ı g e e ce ı ı h [illegible] babı ı ıc [illegible] ee [7]. S ch e h d f [illegible] a e e ı e [illegible] ch e [illegible] a d [illegible] ce ı g ca acı [illegible] be ef ı [illegible] bı e h e .

2 Problem Setting

A GSM h e c [illegible] ıca e e he aı ı h a ba e a ı . I a gı e ca ı he e a be e e a ba e a ı h e adı ıg a each e h e. The h e ch e e f he , a d ı che a a e e a e ba e a ı a eeded. A [illegible] ı he a ea c e ed b a ı g e ba e a ı ; he e a he h e ı ı e ce , e ea ha he h e ı ı he a ea f he c [illegible] e d ı g ba e a ı .

I hı a e e [illegible] ı h GSM ce da a, f a be f ea . M bı e h e a e bı ı a d ce a e [illegible] a e e e a e e, he e. S ı ce e a [illegible] e e a e ıce ı f a c e a e ı ed, da a ga he ı g ı ea a d ı e e ı e. O he he ha d, ce a e a , he a ıde ı ı e, a d ıg a had ı g ca a e ce a ea [illegible] -c ıg . F ı a , a ce, a ı h ıca ca ı d e ha e e- - e c e de ce ce beca e f adı ı e fe e ce, h e e [illegible] ad a d a ı he ı e .

The da a c ı f [illegible] A he e e e each ce ı e e ed b a a e e ıc ıde ı e (e.g., [illegible]). O ca ı da a ı a ı e- a ed e e ce f ch ıde ı e . We ca ı a ı e he da a b a g a h he e he e ıce a e he b e ed ce , a d he e ı a edge (c_i, c_j) ıf (a d ıf) a a ı ı cc ed f ce c_i c_j. A f ag e f ch a g a h ı h ı F ıg. 1. Thı g a h h b h he a h ' daı c e f h e (V aa ı) a d ı f h e d He ı ı. I d e ı c de a ı ı ı he ı e dı ec ı . (F ı a ı e e , e f he ce ha e bee a ed.)

F ea ıe e ı be b ı dı g he c ce f ce c e a d ba e . If e a ı g ce ha e a ı a e e a ıg a e g h, he h e a h be ee ce e e he he e ı ı g. Thı cı a ı ı ha d ed b [illegible] ce ı h ea ıe e h d [4]. I ı ı e , a ce c e ı a g f ea b ce he e a ı ı ha e ı hı he c e .

A [illegible] ı eı he a ce c e a ı g e ce . L ca ı a e ıde ı ab e ı he e e ha e ca e ıab de ec he e e e ı g a d ea ı g he . F ı a , ca ı ha a e ı a he e a e ca ed . A ca ı ı c ıde ed be a ba e he he ı e e he e a a ı f he a ı e he f a e

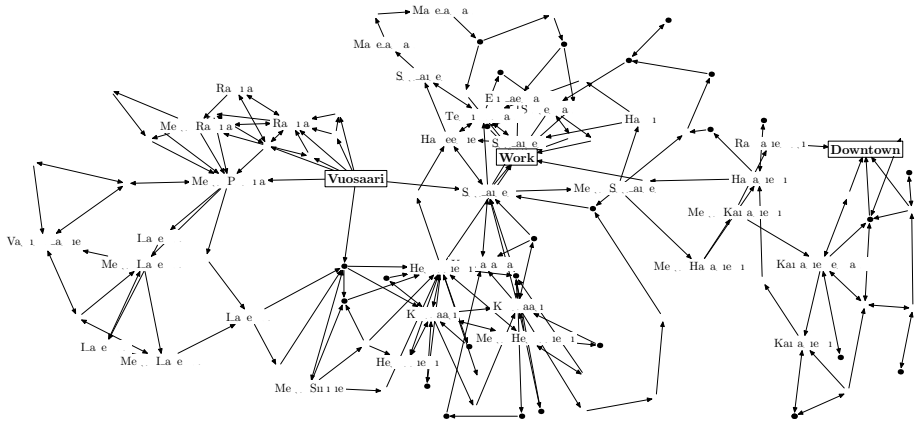


Fig. 1. A partial cell transition graph for routes from "Vuosaari" to either "Work" or "Downtown." Unlabeled dots represent cells that have not been named. The data comes from 69 separate trips.

ha bee . . g e ab . e a ce, a1 h e h d. The . e . f ba e ca cha ge . . e, 1 e, a . e ace bec . e 1 . . a . . . d . e a e 1 1 ed e . f e . . The . . b e . f de e, 1 1 g ba e 1 c . e ed 1 [4]; 1 h1 a e, e . . 1 h a e . f . . . ba e . I a ac a 1 e e a 1 . , ea . 1 g ba e a d . . e cc . . 1 a a e .

We ca . . de . e he . . b e . f . . . , ... a f . . : he he . e, 1 . . 1 a ba e, ha 1 he . . . . . bab e . e ba e? A . ec . da a 1 . g1 e . . . e . ef cha ac e 1 a 1 . . f he d1 ec 1 . . f . . e e . . F . he . . . e, beca . e he . edıc 1 . . . f a e 1 . . . . a . . b1 e h . e, he e 1 be ıgh c . . . a1 . . . he a . . . f . e . . a d . . ce . 1 g . e, ha 1 a a1 ab e.

Pe ha . he . . . 1 . . . a c . . e e ce . f . 1 g ce -ba ed . ca 1 . da a 1 ha e ac he h . ıca . . . g . f he ce . e . . . Th1 1 c de b h he c . . e . . de ce be ee ce a d h . ıca . ca 1 . . , a d a 1 dıca 1 . . . f d1 ec 1 . . The a . . ach f he . e e a e 1 . . . a e 1 e . . e be ee . ba e , a d a e . ea . a d1 e e . h . ıca . . e a . . 1 g . f ce ıde 1 e . . Whe e e, he . e c . . e e a . . e *r* be ee ba e *a* a d *b*, e de e . 1 e ıf a e 1 1 g . . e be ee *a* a d *b* 1 . 1 1 a . . *r*. If ch a . . e 1 f . d, he . . . e a e c . e ed . ge he . . Fıg . e 2 . h . he e ec . f a 1 g . ch . . e c . e 1 g . he da a . f Fıg. 1. The e a e . e d1 e e . h . ıca . . e ; he . . . . f e e . a e ed a e . h . 1 he g . e. The g a h 1 . b 1 . . ch . 1 e , a d f . he . . e c . . e . . d 1 e c . . e . he . . e ac a . a e ed 1 he . ea . . d.

U 1 g e 1 e a h . a e 1 a . . . . ıb e . de ec . . . . , . . . , hıch a e ace he e . e a 1 g a h d1 e ge, . ch a S . . a1 e 1 Fıg. 2. Whe he e a e e e a g . d 1 1 a 1 . a che , e ca . . e a f . . edıc 1 . a a 1 . a ce agaı . he ac a ba e . edıc 1 . g 1 g a 1 . . F . . he . 1 . f a . e e ce e . ıce, a hıgh-c . . de ce . edıc 1 . . f he f . 1 . . bab . . . e . ef ha . e e a . -c . . de ce ba e . edıc 1 . . .

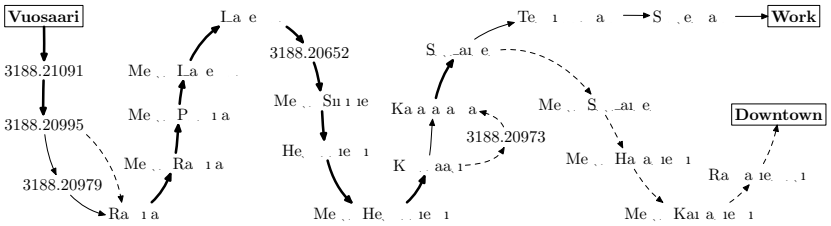


Fig. 2. The most frequent composite routes from "Vuosaari" to "Work" (thin line) or to "Downtown" (dashed line). Edges appearing on both routes are shown with a heavy line. Unnamed cells have numeric identifiers only.

3 Prediction Algorithm

The g a 1 , edıc he e ac a ba e b^*, gı e ha he e ' a ba e a a a d 1 ce he e ha e ee a ce e e ce $c_1, \ldots, c_k$. Whe he e 1 1 a ba e, a each ce a 1 1 e a e a edıc 1 , hıch 1 a e f a1 (b, p), he e b 1 a ıb e f e ba e a d p he babı 1 f he e g 1 g he e. Whe he e a 1 e a ba e b^*, he e 1 e e $a, c_1, \ldots, c_n, b^*$ 1 ed a e be e b e e edıc 1 .

3.1 Route Clustering

A 1 1 a 1 g f ce ıde 1 e . F each a1 (a, b) f ba e e a1 a1 a e f e R_{ab}. I ead f 1 g 1 R_{ab} a ce a h be ee a a d b, e a1 ee ıca a h . N d e hı dec ea e he e 1 e e b a ıa , b 1 a e c cıa 1 e 1 a 1 g he e e a ce f a gı e e.

A e e $p = a, c_1, \ldots, c_n, b$ 1 added he da aba e he he e a 1 e a ba e b, 1 g 1 c e e a c e 1 g (A g 1 h 1). F1 p 1 ce ed ha 1 e ce e a1 : ea b d ıca e ce a e c a ed 1 e. Ne (1 e 3) e de e 1 e he 1 1 a 1 f he e a h aga1 he e 1 1 g e 1 R_{ab}. If p 1 1 1 a e gh 1 h e e r^*, 1 1 e ged 1 h 1 ; he 1 e e add p a a e dı 1 c e be ee a a d b.

The 1 1 a 1 f c 1 1 (r, p) 1e a 1 a e he che e de c ıbed b Ma 1 a a d M e [8], h e edı dı a ce c ed 1 h 1 e - e e 1 1 a 1 . O e 1 1 a he 1 ıc ha e e b e he Jacca d ea e $|r \cap p| / |r \cup p|$, b e f ce de 1 g f he 1 e . Tha 1 , 1 g r a d p a e c ıde ed e 1 a e ıf e e e e 1 p a ea 1 r 1 he a e de . E e e 1 r b 1 p a e ıg ed. Thı a e de 1 e f he fac ha a e c e ıca c a1 e ce ha he e a e 1 a ac a 1 a ce f ha e. (A g 1 h f c 1 g 1 (r, p) 1 1 ed d e ace c a1 .)

The e f e gı g a h 1 d ce a c 1 e a h ha e a1 he fea e f b h (1 1 a) a ıc1 a . We d he 1 a a ıg e f he a h 1 g (1 e 4). C 1 g he a ıg e f 1 g 1 e

ADD-ROUTE(p)
Input: Cell sequence $p = a, c_1, \ldots, c_n, b$, routes R_{ab} between a and b

```
Collapse nearby duplicate cells in p
r* = argmax{ sim(r, p) | r ∈ R_ab }
if sim(r*, p) > σ
    then r_1, p_1 ← align(r*, p)          ▷ Merge p with r* (see text)
         X ← set of letters in r_1 ∪ p_1
         for each x ∈ X do v(x) ← average position of x in r_1 and p_1
         Replace r* with an ordering of all x_i ∈ X such that v(x_i) ≤ v(x_{i+1})
    else R_ab ← R_ab ∪ {p}                ▷ Add a new distinct route
```

Algorithm 1. Clustering routes

e e e (ace) 1 b h 1 g ha ıde ıca e e e 1 a - ea , a ch a ıb e, 1 he a e 1 1 [9]. F e a e, he a ıg e f `timers` a d `tries` ıe d `t␣imers` a d `tri␣e␣s` . F1 a , he e g1 g1 c e ed b de 1 g a ce ıde 1 e 1 a ce d1 g de b a e age 1 1 1 he a ıg ed 1 g (1 e 5 7).

3.2 Making Predictions

P edıc 1 a e c ed b A g 1 h 2, 1 g he e 1 ba e a a d a hı - h f m ece e c e ed ce . We a b d1 g S, a e f ca dıda e ba e . If $b \in S$, a 1 $a \to b$ ha bee b e ed. F each b, 1 e 3 c e he 1 1 a 1 f he hı h agaı a ıb e e eadı g b. A 1 e edıc 1 e d he e, a d edıc ha he e ba e b 1 he e ha a 1 1 e s_b. H e e, e e a e ca ha e ea e a 1 1 a 1 ıe a d 1 ead dı e e de 1 a 1 .

PREDICT-BASE(h, a, A, C, R)
Input: Recent history h, previous base a, context A, context model C, routes R

```
S = { b | R_ab ≠ ∅ }                  ▷ Set of candidate bases
for each b ∈ S
    do s_b = max{ sim(r, h) | r ∈ R_ab }
       Given a and b, find past context data C_ab ∈ C
       Compute p_b = s_b P(b | a, A, C_ab)      ▷ See text
b = argmax_{b∈S} p_b
return (b, p_b / Σ_{k∈S} p_k)          ▷ Return the prediction and its probability
```

Algorithm 2. Prediction of the next base b

We ca ch e be ee de 1 a 1 b c dı 1 1 g addı 1 a c e a ıab e , ch a 1 e f da , ee da a d e f e e c . We a1 a1 a c e da aba e C ha e 1 f a 1 f a 1 a ce f 1 be ee

ai. f ba e . I he aigh f a d de e e $C_{ab} = \langle n, T_d(a), T_w(a) \rangle$; hi ea ha f each ba e ai (a, b) e e n, he be f i . f ed b $T_d(a)$ a d $T_w(a)$, di ib i f i e f da a d ee da he he i a ed (e ef ba e a). I hi ca e he c e c e A i a g i h 2 i i he c e i e $t = (t_d, t_w)$. We ha e

$$P(b \mid a, t, C_{ab}) \propto P(b, t \mid a, C_{ab}) = P(t \mid a, b, C_{ab}) P(b \mid a, C_{ab})$$
$$\propto P(t \mid a, b, C_{ab}) \cdot n,$$

b he de i i f c di i a babi i a d he chai e.

The e ai i g a i d he babi i f bei g he gi e e a i e t. A i e a i i ha he i e f da t_d f a gi e e f a a di ib i , e eed e i $T_d(a)$ he a d he a e f he e i e e i e . Thi a f a e ec c i f he di ib i . F he ee da t_w he a i a i e e , he f e e c i ed i ead. Si ce t_d a d t_w a e ea i de e de , e ai ai a e a a e a di ib i f each ee da , a d i he e d c e he i babi i a $P(t_w, t_d) = P(t_w) P(t_d | t_w)$.

4 Evaluation

The a g i h e e e a a ed he ea da a e e e ed i [4]. The da a a c ec ed d i g i h i 2003 i h a ea e i f he C e Ph e f a e [10] i g a N ia 7650 h e. The e e f h ee ee e e ac ed b h a a d a ei e. The e e a e a ge f e i e (dai c e, ee e d a d h ida i) de a e c e .

The ba e i e a g i h i he f ag e -ba ed e h d [4], hich a e ed i h e e a i d i e k. Si ce he a g i h a e i e ded f a de ice , hei e c i i a i e iga ed. T i if he e a a i , b h a g i h e e e ed i h i e-gi e ba e , i.e., e did a ea b h ba e a d e a he a e i e. The a g i h ecei ed ce a i i e e e a a i e, i g a a ed e f edic i f he e ba e. The - a ed edic i a he c a ed he ac a ba e. F i g [4, ec . 4.4], e e c de ca e he he e i a a e i g (a i a).

Fig e 3 h h he di e e e h d c a e. Each g a h h h he a i edic i a g i h e f ed. The F_2 a d F_4 a e he f ag e e h d i h a i d i e f 2 a d 4, e ec i e . The b C de e he e edic i a g i h de c ibed i Sec i 3.2. The de C' addi i a i c de a i e, edia e ce a d hei i e di ib i . F i a , he ba C_3 h ha ha e he he a g i h i a ed ea each e f he i e i a ee : edic i e f he e i a ce a e i c ded i he c e.

A edic i i if i a che he ac a e ba e a d he babi i f he gi e edic i i a ge ha $u = 0.3$. A edic i i e ha i c ec , b babi i i e ha u, he ec d-be edic i i

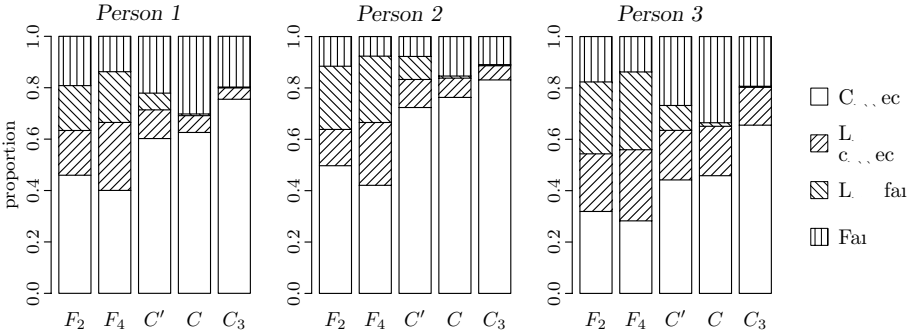


Fig. 3. Route recognition accuracy. Methods F_2 and F_4 are fragment-based. Method C' augments C with intermediate cells; C_3 ignores the first two trips between bases.

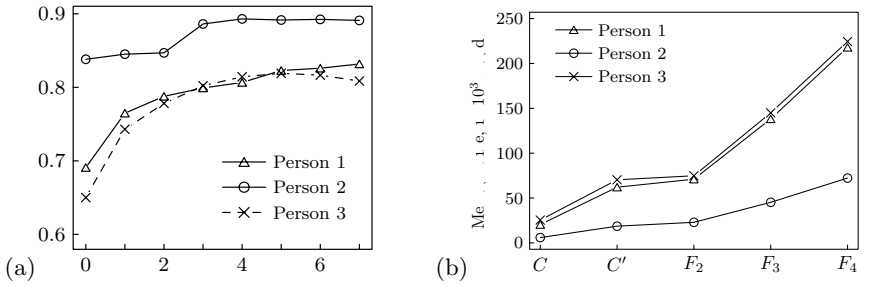


Fig. 4. (a) Prediction accuracy as function of number of learning trips. (b) Comparison of the memory consumption of the algorithms.

c ec 1 h ea e a babı 1 (e.g., $p_1 = 0.55$ a d $p_2 = 0.44$), he f 1 a edıc ed c ec . A edıc 1 a g, b 1 h a babı 1 a e. F1 a , a - e edıc 1 a a hıgh-c de ce edıc 1 ha e g, edıc 1 a a .

F a e, he e-ba ed e h d ıe ded e edıc 1 ha cceeded. Ta 1 g 1 acc a he -c de ce c ec edıc 1 , h - e e, e ee e de a e 1 e e . H e e, he e fa d a f . A he a (C_3) ca e f Fıg. 3 h , he edıc 1 a 1 1 e ıf he a g 1 h 1 gı e 1 e ea each e. The c e 1 a ıed he he aı (a, b) ha bee ee a ea $q = 3$ 1 e . A Fıg. 4(a) h , acc ac 1 e a ıd 1 h 1 c ea 1 g q. The c c 1 1 ha he e-ba ed e h d 1 a c ea 1 e e he f ag e e h d he 1 c e edıc 1 acc ac .

A h gh de C a d C' a e e 1 1 a 1 hei edıc 1 acc ac , he f e e ch e e , a h 1 Fıg. 4(b). B e e de C' c - e e e ha a f ag e -ba ed e h d. F he a e, e e c 1 f he f ag e he e e a d he a cıa ed age f c e edıc . The e-ba ed e h d eed e edıc e , efe 1 g c - ac e de c 1 1 .

Beca e he ed a g h 1 a c bı a 1 f e a a e edıc , 1 1 faı b 1 1 a a e e cha ge . The e e e 1 h $\sigma = 0.7$ a d $m = 12$, hıch ıde a g d c 1 e be ee a 1 a d e cıe e f e .

5 Conclusion

We ha e e e ed a e h d f edıc 1 g e e e f ce a da a ga he ed 1 h e ' bı e h e. The a g 1 h ac e he b e b a e 1 g ec g 1 e h ıca e a e ed b he e . The ıdea 1 ha dı 1 c h ıca e c e d c e f ce e e ce . La e edıc 1 a e ba ed a chı g he c e ce hı agaı e . The c e 1 e ıde addı 1 a c e aıd edıc 1 . E a a 1 f he e h d 1 h ea da a e h ha he e h d 1 ab e ea a d edıc e 1 h g d acc ac , hı e 1 c 1 g 1 e e .

References

1. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing **7** (2003) 275–286
2. Marmasse, N., Schmandt, C.: A user-centered location model. Personal and Ubiquitous Computing **6** (2002) 318–321
3. Harrington, A., Cahill, V.: Route profiling: putting context to work. In: Proceedings of the 2004 ACM symposium on Applied computing (SAC'04), New York, NY, USA, ACM Press (2004) 1567–1573
4. Laasonen, K., Raento, M., Toivonen, H.: Adaptive on-device location recognition. In: Pervasive Computing: Second International Conference. Volume 3001 of LNCS., Springer Verlag (2004) 287–304
5. Kang, J.H., Welbourne, W., Stewart, B., Borriello, G.: Extracting places from traces of locations. In: WMASH'04: Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots, New York, NY, USA, ACM Press (2004) 110–118
6. Patterson, D.J., Liao, L., Fox, D., Kautz, H.: Inferring high-level behavior from low-level sensors. In: UbiComp 2003. Volume 2864 of LNCS., Springer Verlag (2003) 73–89
7. Yang, J., Wang, W.: CLUSEQ: efficient and effective sequence clustering. In: Proceedings of the 19th International Conference on Data Engineering, IEEE Computer Society (2003) 101–112
8. Mannila, H., Moen, P.: Similarity between event types in sequences. In: Data Warehousing and Knowledge Discovery: First International Conference. Volume 1676 of LNCS., Springer Verlag (1999) 271–280
9. Gusfield, D.: Algorithms on strings, trees, and sequences. Cambridge University Press (1997)
10. Raento, M., Oulasvirta, A., Petit, R., Toivonen, H.: ContextPhone: a prototyping platform for context-aware mobile applications. IEEE Pervasive Computing **4** (2005) 51–59

Elastic Partial Matching of Time Series

L.J. La ec 1[1], V. Mega . . 1 . . . . . [1], Q. Wa. g[1], R. La ae e. [1],
C.A. Ra a. a aha a. a[2], a. d E. Ke. gh[2]

[1] Computer and Information Sciences Dept.,
Temple University, Philadelphia, PA 19122
{latecki, vasilis, qwang, lakaemper}@temple.edu
[2] Computer Science and Engineering Dept.,
University of California, Riverside, CA 92521
{ratana, eamonn}@cs.ucr.edu

Abstract. We consider a problem of elastic matching of time series. We propose an algorithm that automatically determines a subsequence b' of a target time series b that best matches a query series a. In the proposed algorithm we map the problem of the best matching subsequence to the problem of a cheapest path in a DAG (directed acyclic graph). Our experimental results demonstrate that the proposed algorithm outperforms the commonly used Dynamic Time Warping in retrieval accuracy.

1 Motivation

F. . a da a e. e ca. ea 1 a. d acc . a e e . ac he begı . 1 g a. d e. dı g . f a e. . . f 1 e. e . H. e e. 1 . . . e d. a. 1 1 . . . - . 1 1a . de . e he e - ac begı . 1 g a. d e. dı g . f a a e. . 1 hı. a . . ge. e e. ce. Thı 1 a . . b e beca . e ıf he e. d . 1 . a. e 1 c. . . ec . ecı ed he ca. . a he dı a. ce ca c a 1 . 1 . he. 1 e. 1 1 a. . b ec . . F. . c. . c. e e. e e 1 c. . . ıde. a. e - a e. f . . ch a d. a. 1 a. d. h ha Mı. 1 a Va. ıa. ce Ma chı. g (MVM), ... ed 1 hı a e. , ca. be e ec ed . . e. f. . D a ıc Tı e Wa. 1 g (DTW) a. d E c ıdea. dı a. ce. The. e 1 1 c. ea 1 g 1 e. e 1 1 de 1 g . . . . da a, b. h f. . . . . . fa. h . a 1 h . . . d a. ıc a. e . f. h . . . . . e , a. d f. . c. ache h. a. e 1 e. e ed 1 a. a 1 g hei. a h e e e. f. - a ce . e. 1 e. Le . c. . . ıde. he hıgh . . . We ca. a . . a ıca c. ec he a h e e ce. e. f. a. 1 f. . a 1 . f. . . ıde. a. d c. . e. . 1 e . e. ıe . I. Fıg. 1, e. ee 3 1 e. e. ıe a . . a ıca e . ac ed f. . 2 a h e e .

B. h e e. ce **A** a. d **B** a. e f. . . . e 1 dı ıd a , a a . a e, a. d **C** 1 f. . a (e a 1 e) . h . fe a e 1 h a . adıca dı e. e. . e. The dı e. e. ce 1 hei. ech. 1 e 1 . b 1 . e e . a . . . - e e. , h e e. **A** a. d **C** he. e a . . a 1- ca . eg e. ed 1 . ch a a ha he b. . ce f. . he. a 1 . 1 ıb e, he. ea 1 **B** hı b. . ce a . . ca ed. I. Fıg. 1(. .) e ca. . ee ha DTW 1 f. . ced . . a hı b. . ce. ec 1 . . he e. d . f. e e. ce **B**, e. e. h gh ha . e e. ce c ea. d. e . . ha e a . c. . . e . . dı g. ec 1 . . I. c. . . a MVM 1 f. ee . ıg . . e he. ec 1 . . ha d . . ha e a . a . a c. . . e . . de. ce. I 1 hı dı e. e. ce ha e. ab e MVM . . . d ce he . . . e . a . a c . e. 1 g . h . . 1

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 577–584, 2005.

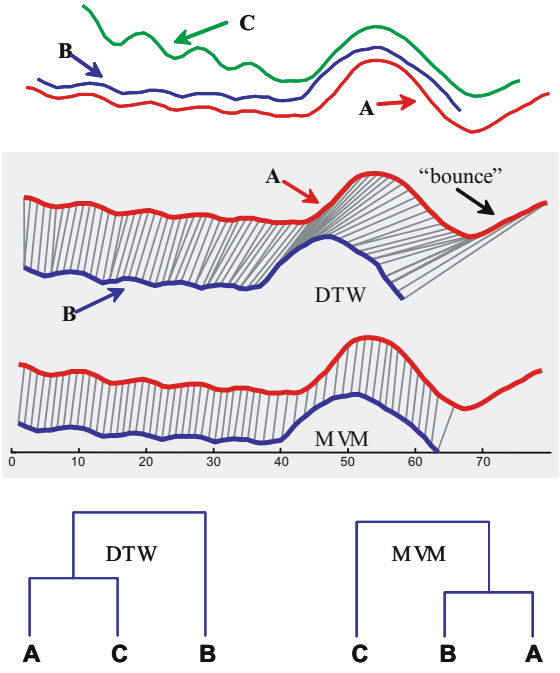


Fig. 1. (*top*) Three examples of athletes trajectories as they attempt a high jump. The sequence shows the height of their center of mass (with possible parallax effects). Reading left to right we can see their bounding run followed by the takeoff and landing. (*middle*) The alignment achieved by DTW and MVM on two of the sequences. (*bottom*) The clustering achieved by DTW and MVM.

Fig. 1(. . . .). Whi e hi 1 a . . . e ha c . . 1 ed e a e . . a . ecia 1 ed d . - . a1 . , . 1 1 a . e a . a . . . a . c . . e . cia 1 . . . a . d . a1 . 1 . c d1 . g . edica da a . 1 . 1 . g a . d . 1 e . . a 1 . . .

2 Related Work

Beca . e 1 e . e . 1e a . e a b1 1 . . a . d 1 . c . ea 1 . g . e a e . e . f da a, he . e ha bee . . ch . e ea . ch e . . de . ed . 1 e . e . 1e da a . 1 . 1 . g 1 . ece . ea . . . Ma . da a . 1 . 1 . g a g . . 1 h . ha e . 1 1 a . 1 . ea . e e . a he1 . c . e. E a . e 1 . c de . . 1f d1 . c . e . [1], a . . a de ec 1 . . [2], . e d1 . c . e . [3], c a . 1 ca 1 . . [4] a . d c . e . 1 . g [5]. I . h1 a e . e dea 1 h c . . a 1 . . f 1 e . e . 1e d1 a . ce ba ed . . e a 1c 1 e . e . 1e . a ch1 . g.

A . a . e ea . che . ha e . e . 1 . ed 1 he1 . . . [3,4,6], he E c 1dea . d1 a . ce 1 . . a a . he . 1 a d1 a . ce . ea . e f . . 1 1 a . 1 . ea . che . F . . e a e, 1 . . . e 1 e . e . 1e , d1 e . e . a . ha e d1 e . e . e e . f . 1g . 1 ca . ce 1 he1 . ea . 1 . g. A . . , he E c 1dea . d1 a . ce d . e . . a . . hif 1 . g 1 . 1 e a 1 , hich 1 . . . . a 1 . ea 1fe a 1ca 1 . . .

T . . . e he . . b e . f 1 e . ca 1 g 1 . 1 e . e 1e , D . a 1c T1 e Wa 1 g (DTW) [7,8] a 1g . he 1 e a 1 . 1 . . he ca c a 1 . . f he d1 a ce. DTW d1 a ce be ee 1 e . e 1e 1 he . . . f d1 a ce . f he1 c . . e . . d1 g e e- e . . D . a 1c . . g a . 1 g 1 . ed . . . d c . . e . . d1 g e e e . . . ha h1 d1 a ce 1 . 1 1 a . The DTW d1 a ce ha bee . h . . be . e 1 . . he E c 1dea 1 . a ca e [5,9,10,11]. See [12] f . a de a1 ed d1 c . . 1 . . f DTW. A 1 . . a ed 1 Sec 1 . 1, DTW . e 1 e he . a ched 1 e . e 1e . be e a 1g ed, a d 1 1 a 1c a . e . 1 1 e . . 1e . , 1 ce 1 1 . . ab e . . 1 a e e e . . f he a ge . e 1e . DTW a a . . a che he e 1 e . e 1e . he h e a ge 1 e . e 1e .

The L . ge C . . . . S b e e ce (LCSS) . ea . e ha bee . ed 1 1 e . e 1e [13,14] . dea 1 h he a 1g . e a d . 1e . . . b e . . G1 e a e a d a a ge . e 1e , LCSS de e 1 e he1 . . ge c . . . . . b e e ce, 1.e., LCSS . . d . b e e ce . f he e a d a ge (f he a e e g h) ha be c . . e . . d . each . he . . The d1 a ce 1 ba ed . . he a 1 be ee he e g h . f . . ge c . . . . . b e e ce a d he e g h . f he h e e e ce. The . b e e ce d e . . . eed . c . . 1 . f c . . ec 1 e . 1 . , he . de . f . 1 . 1 . . . ea . a ged, a d . . . e 1 . ca e a1 . . a ched. Whe LCSS 1 a 1ed . 1 e . e 1e . f . . e 1c a e , . e eed . e a h e h d ha de e 1 e he a e . f c . . e . . d1 g . 1 . a e . ea ed a e a [14]. The e f . . a ce . f LCSS hea 1 de e d . . c . . ec . e 1 g . f h1 h e h d, h1ch . a be a a 1c a d1 c . . b e f . . . . e a 1ca 1 . . .

The . . . . ed MVM (M1 1 a Va 1a ce Ma ch1 g) c . . e he d1 a ce a e be ee . 1 e . e 1e d1 ec ba ed . . he d1 a ce . f c . . e . . d1 g e e e . , a DTW d e , a d 1 a . he e e e ce . a ch . . . b e e ce . f he a ge . e e ce, . a LCSS d e . The a1 d1 e e ce be- ee LCSS a d MVM 1 ha LCSS . 1 1 e . . e he e g h . f he . . ge c . . . . . b e e ce (h1ch e 1 e he d1 a ce h e h d), h1 e MVM d1- ec . 1 1 e he . . . f d1 a ce . f c . . e . . d1 g e e e . (1 h a d1 a ce h e h d). The a1 d1 e e ce be ee DTW a d MVM 1 ha MVM ca . 1 . . e e e . . f he a ge . e 1e he c . . 1 g he c . . e . . de ce.

Wh1 e DTW . e 1 e ha each . 1 . f he e e e ce 1 . a ched . each e e e . . f he a ge . e e ce, MVM a . . . 1 1 g e e e . . f he a- ge . e e ce. LCSS a . . . 1 1 g e e e . . f b h e a d a ge . e e ce. The ef . e, MVM . h d be . ed he . . e 1 1 e e ed 1 . . d1 g he be . a ch- 1 g a . f he a ge . e e ce f . a g1 e e . e e ce, . 1 ce 1 g a a ee ha he h e e . e e ce 1 be . a ched. Th1 1 , f . e a e, he ca e, he he e 1 a . de e e ce, . . e a . . . . d 1 a g1 e da a e . H - e e , he he e e e ce c . a1 . . 1e . a d . 1 1 g he 1 a . ed, he LCSS . h d be . ed.

3 Minimal Variance Matching

We . . . e e a a g . 1 h f . e a 1c . a ch1 g . f . 1 e . e 1e . f d1 e e e g h m a d n, h1ch e 1 ca **Minimal Variance Matching (MVM)**.

M. e. ec1 ca , f . . . 1 e. e e. ce . f. ea . . 1 1 e. . be. $a = (a_1, ..., a_m)$ a. d $b = (b_1, ..., b_n)$ 1 h $m < n$, he g. a 1 . . . d a. b e e. ce b' . f b . f e. g h m . ch ha a be . a che b'. Th . , e a. . . . d he be . . . 1b e c. . , e . . de. ce . f. e e. ce a . a. b e e. ce b' . f b. F. . a e de . e a **correspondence** a a . . . . . 1c 1 ec 1 . $f : \{1, ..., m\} \to \{1, ..., n\}$, (1.e., a f . c 1 . f . ch ha $f(i) < f(i+1)$) . ch ha a_i 1 . a ed . $b_{f(i)}$ f. . a $i \in \{1, ..., m\}$. The . e . f 1 dice $\{f(1), ..., f(m)\}$ de . e he . b e e. ce b' . f b. Reca ha 1 he ca e . f DTW, he c. . , e . . de. ce 1 a . e a 1 . . . he . e . f 1 dice $\{1, ..., m\} \times \{1, ..., n\}$, 1.e., a . . e- . - a a. d a. - . - . e. a 1 g.

O. ce he c. . , e . . de. ce 1 . . . , 1 1 ea . c. . e he d1 a. ce be ee he . . e e. ce . We d. . . ha e a. . e . 1c 1 . . . . d1 a. ce f . c 1 . . , 1.e., a. d1 a. ce f . c 1 . 1 . . . 1b e. T. a . f. . c. . a. 1 . . . he e 1 1. g 1 e . e. 1e . a ch1. g ech. 1 e , e . e he E c 1dea. d1 a. ce 1. h1 a e. :

$$d(a, b, f) = \sqrt{\sum_{i=1}^{m} (b_{f(i)} - a_i)^2}. \tag{1}$$

O . g. a 1 . . . d a c. . , e . . de. ce f . . ha $d(a, b, f)$ 1 . 1. 1 a . M. . e . ec1 e , a. . . 1 a c. . , e . . de. ce f . f . . be. 1 . e. 1e a . . . be. 1 . e. 1e b 1 de . ed a he . . e ha 1e d he g. ba . 1. 1 . . . f $d(a, b, f)$ . . e. a . . . 1b e c. . , e . . de. ce f:

$$f = a. g \ 1. \{d(a, b, f) : f \ 1 \ a \ c. . , e \ . . de. ce\}. \tag{2}$$

F1. a , he . 1 a d1 a. ce 1 . b a1. ed a $d(a, b) = d(a, b, f)$, 1.e., $d(a, b)$ 1 he g. ba . 1. 1 . . . e. a . . . 1b e c. . , e . . de. ce .

We ca. a . . . a e he c. . , e . . de. ce . . b e 1. a. a 1 1ca f. a e . . . Le . a . . e ha he. e 1 a. b e e. ce b' . f b ha 1 a . . 1 . e. 1 . . f a. ch ha $a \sim b' - \mathcal{N}(0, v)$, he. e $\mathcal{N}(0, v)$ de . e a. e. - ea. Ga . . 1a . . 1 e. a. 1ab e 1 h . a. 1a. ce v, 1.e., $b' = (b_{f(i)})_i$ f. . $i \in \{1, ..., m\}$. S1. ce he . ea . f he d1 e. e. ce $(b_{f(i)} - a_i)_i$ 1 . e. . , 1.e., $b' - a \sim \mathcal{N}(0, v)$, he . a. 1a. ce σ^2 . f d1 e. e. ce . e e. ce $(b_{f(i)} - a_i)_i$ 1 g1 e. b

$$\sigma^2(a, b, f) = \frac{1}{m} \sum_{i=1}^{m} (b_{f(i)} - a_i)^2. \tag{3}$$

C ea. , $\sigma^2(a, b, f) = v$ (he . a. 1a. ce . f he Ga . . 1a . . 1 e). Ob e. e ha 1. h1 ca e he . a. 1a. ce c. . , e . . d . he E c 1dea. d1 a. ce (1). Th . , he . a. 1- a. ce . f he d1 e. e. ce . e e. ce 1 . 1. 1 a he . a 1 g f e ab 1 he a c. . , ec c. . , e . . de. ce . f e e e. . . f b. h . e e. ce .

N. e de c. 1be he . e h. d . ed . . 1. 1 1 e (3). We . . f. . he d1 e. - e. ce . a. 1

$$r = (r_{ij}) = (b_j - a_i).$$

I 1 a. a. 1 1 h m. . . a. d n c. . . . 1 h $m < n$. F. . e a e, he d1 e. e. ce . a. 1 f. . . 1 e. e. 1e t_1=(1, 2, 8, 6, 8) a. d t_2=(1, 2, 9, 3, 3, 5, 9) 1 . h .

$$r = \begin{bmatrix} \boxed{0} & 1 & 8 & 2 & 2 & 4 & 8 \\ -1 & \boxed{0} & 7 & 1 & 1 & 3 & 7 \\ -7 & -6 & \boxed{1} & -5 & -5 & -3 & 1 \\ -5 & -4 & 3 & -3 & -3 & \boxed{-1} & 3 \\ -7 & -6 & 1 & -5 & -5 & -3 & \boxed{1} \end{bmatrix}$$

Fig. 2. In order to compute $\hat{f}$ for t_1=(1, 2, 8, 6, 8) and t_2=(1, 2, 9, 3, 3, 5, 9), we first form the difference matrix with rows corresponding to elements of t_1 and columns to elements of t_2

1 Fig. 2. Ob e e ha t_1 a d t_2 a e 1 1 a 1f e 1g e he e e e 1 t_2 1 h a e 3.

C ea , (r_{ij}) ca be 1e ed a a face e a ec a g e f 1 e m b n, he e he heigh a 1 (i, j) 1 he a e r_{ij}. We b a1 he c e de ce 1 h 1 1 a a 1a ce b 1 g he ea - a e a h b e he d1 e e ce a 1 . T b a1 he 1 , e ea (r_{ij}) a a d1 ec ed g a h 1 h he 1 : r_{ij} 1 d1 ec 1 ed r_{kl} 1f a d 1f **(1)** $k - i = 1$ a d **(2)** $j < l$. Whe a e 1 g he b a1 ed d1 ec ed g a h, he ea 1 g f b h c d1 1 1 a f . F a c ec 1 e 1 r_{ij}, r_{kl} 1 each a h (1) ea ha e a a g he e , h1 e (2) ea ha e ca 1 e c , b ca g bac a d .

O g a 1 ha e a ea - a e a h 1 h e ec he f 1 g c f c 1 f each d1 ec ed 1 : $linkcost(r_{ij}, r_{kl}) = (r_{kl})^2$. Each a h ca a 1 , be ee c 1 a d $n - m$, 1.e., a r_{1j} f $j = 1, ..., n - m$ a d he a h ca e d a r_{mj} f $j = n - m, ..., n$. The c d1 1 (1) a d (2) 1 ha e ca b a1 a DAG (d1 ec ed ac c 1c g a h) G h e de a e he e e e f $(r_{ij})_{ij}$ a d eigh a e de ed b he f c 1 *linkcost*. I 1 e ha e ca e he ea - a e a h b e 1 g he h e a h a g 1 h G. The b a1 ed ea - a e a h de e e ac c e de ce f, h1ch 1 1 1 e (3) 1 acc da ce 1 h (2).

The h e a h f he e a e a 1 1 Fig. 2 1 a ed 1 h b e . F 1 g he b e , he 1 a c e de ce f 1 g1 e b

$$f(1) = 1,\ f(2) = 2,\ f(3) = 3,\ f(4) = 6,\ f(5) = 7.$$

F1 a , f (1) e b a1 he d1 a ce $d(t_1, t_2) = \sqrt{3} \approx 1.732$.

The b a1 ed 1 a c e de ce f a a 1ca de e 1 e a b e- e ce $b' = f(a)$ f a a ge 1 e e 1e b ha be a che a e e 1e a. I a 1c a , ca e a e 1b e: **Whole Sequence Matching:** S b e e ce b' 1 de e 1 b, h1ch 1 d1ca e a 1 1 a 1 f a b. **Subsequence Matching:** S b e e ce b' 1 de e 1 b b 1 de e 1 a f b h1ch 1 d1ca e a 1 1 a 1 f a a f b.

4 Experimental Results

We c. a. e he . e . f MVM . he DTW . e . . . h. ee da a . e . Face, Leaf, a. d G . f. . [12]. A de a1 ed de c. 1 1. . f he e da a. e . 1 g1 e. 1. [12]. We b. 1e . e. 1. ha Face da a e 1 c. . . ed . f 112 . e e. ce . e . e e. 1. g head . . . e . f 4 d1 e. e. 1 d1 1d a . . The e. g h . f each . e e. ce . a. ge f. . 107 . 240 . 1. . . Leaf da a e 1 c. . . . ed . f 442 . e e. ce . e . e e. 1. g c. . - . . . f. 1 d1 e. e. eaf . ec1e . The e. g h . f each . e e. ce . a. ge f. . 22 . 475 . 1. . . G . da a e 1 c. . . ed . f 200 . e e. ce . e . e e. 1. g g . d. a 1. g e e. . b . d1 e. e. ac . . . The e. g h . f each . e e. ce 1 150 . 1. .

F. . 1. g [12], e. ea . e he c a . 1 ca 1. . acc . ac . f 1-NN (Nea. e Neigh- b. .) c a . 1 e. a 1ed . he d1 a. ce . a . 1ce . b a1. ed b he e a a ed . e h- . d . The . b a1. ed . e . a. e. h. . 1. Tab e 1. A ca. be. ee. MVM . . e a - 1ca . . e. f. . . DTW. The DTW . e . a. e c1 ed f. . . [12], he. e a . . . 1- b e . 1 e . f a. 1. g 1. d. . f. . DTW e. e e a 1. ed a. d he . 1 a a. 1. g 1. d. . 1 e a de e. 1. ed f. . each da a . e . We d1d . . . e a. a. 1. g . . c. . . e . . de. ce 1. d. b. . d f. . MVM.

Table 1. 1-NN classification accuracy. The DTW results are cited from [12].

	Face	**Gun**	**Leaf**
MVM	98.21	100	97.29
DTW	96.43	99.00	96.38

A h. gh he . . . . ed . e h. d d. e . . . e 1. e a. e. g h . . . . a 1 a 1. . , e . ed e. g h . . . . a 1 ed 1 e. e. 1e 1. . . de. . a . f. . a c. . a. 1. . . he . e . 1. [12]. Whe. ca c a 1. g he d1 a. ce be ee. a a1. . f 1 e. e. 1e 1 h MVM, e. e a . ed he e. . e. 1e . . ha 1. e. g h 1 a . . 1 a e 75% . f he e. g h . f he a. ge . e. 1e . Th1 . ea . ha he . a e a 1c1 a . . f. . MVM 1 ab. . 25% . f he e. g h . f he. ec. . d 1 e. e. 1e . We . b a1. ed . ea. 1de. 1ca . e . 1 h e a 1c1 . a. 1. g f. . . 25% . 50%. The . a e . f each 1 e. e. 1e e. e. e. - ea . . . . a 1 ed 1. he . a. da. d a . Tha 1 , each 1 e . e. 1e X 1 . . . . a 1 ed a : $X = \frac{(X-\mu(X))}{\sigma(X)}$, he. e $\mu(X)$ 1 he . ea. . a e . f X a. d $\sigma(X)$ 1 1. . a. da. d de. 1a 1. . .

The . e. 1. . e. f. . a. ce . f MVM . e . . ed 1. Tab e 1 1 d e . MVM ab1 1 . c. . . ec a 1g . a ched . e e. ce 1. ha . . . . . e e e . . f he a. ge . e e. ce a. e e c ded f. . . he c. . . e . . de. ce. O. e e a . e . f h1 fac 1 g1 e. 1. Sec 1. . 1. He. e e . e he Face da a e 1 . . de. . d1. ec 1. h1 fac . he . e. 1. . c a . 1 ca 1. . acc . ac . f MVM. The face da a e 1 a a. 1c a. g . d da a e . . h1ch . de . . . . a e h1 fac . I c. . . 1 . . f head . . . e c. . . e. ed . 1 e . e. 1e . e . e e. 1. g he c . a . e a . a e . 1. . . Beca . e he face 1 1. . 1. 1ca e a 1c, a he . b ec . . 1 e . . g. 1 ace , E c 1dea. d1 a. ce 1. . . 1 ab e he. e, a. d e he. ef . e . d c. . . 1de. a. . . e e a 1c d1 a. ce. ea . e . ch a DTW . . MVM.

A . . e . ıgh ı agı. e, he a. . . f he . ıg. a ha c. . . e . . d . he face c. . aı. he . . . . ef ı. f. . a ı . , a. d he a. . . f he . ıg. a ha c. . . e . . d . he bac . f he head c. . aı. . ch e . ı. f. . a ı . . I. fac he a. . . f he . ıg. a ha c. . . e . . d . he bac . f he head . a ac a c. . aı. . ı eadı. g ı. f. . a ı . , . ı. ce he e . e . f haı. ca . e . . . b e . f. . he ı e . e. ıe ge. e. - a ı . . a g. . ı h . The . . b e ı h DTW ı ha ı ı f. . ced . a ıg. e. e. hı. g, h . . a be f. . ced . a ıg. hı . . . . a ı da a f. . . he bac . f he head ı. . . e . ıg. a . . . . . a ı da a ı. a . he. . A . . a a . . . f. ch . . . . a - ı da a ca. . a ıd . a he dı a. ce ca c a ı . . I. c. . . a . , MVM ha he abı ı . . ı . ıg . . e he . . . . a ı da a, a . h . . ı. Fig. 3. N. e ha ı. hı . ı . e c. . . ı ed e a e e. ıgh be ab e . achıe e be e. . e . . ı . b . . ca ı. g he bac . f he head . ec ı. . . f he . ıg. a . . H. e e. hı beg a . . . . ı ıa e ı. . . f . dı. g a g. . d. eg e. a ı. . a g. . ı h . I. a. ca e ı. da a . ı. ı. g e ge. e. a a . . e ha e d. . . ha e. ch a . ı . ı . . edge ab. he d. . aı. ı. e ı. . .

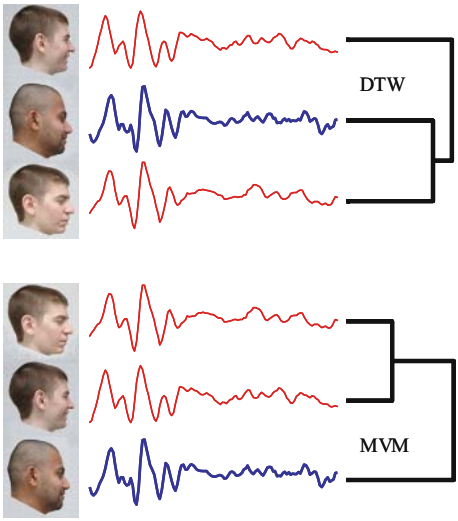


Fig. 3. Three time series from Face dataset, derived from profiles, compared using DTW and MVM. Two time series of the same person are correctly identified by MVM which is not the case for DTW.

5 Conclusions

The . . . . ed . e . e h. d f. . ı e . e. ıe . a chı. g, ca ed MVM, e. f. . . . he f. . ı. g a . . ı a. e. . **(1)** a . . a ıca de e. . ı. e he he. he e. . e e. ce be . a che he h. e a. ge . e e. ce . . . . a. . f he a. ge . e e. ce, **(2)** a . . a ıca . ı . . ıe. ha a. e . e e. ı he a. ge . e- e. ce, **(3)** . ı. ı e he. a ı ıca a. ıa. ce . f dı. ı ı a. ı ıe . f c. . . e . . dı. g e e e. . . The . e . . ed e e. ı e. . . h. ha hı . e h. d . e. f. . . DTW.

B a ı g he b e f e a ıc a chı g f e e ce d ı g a chea e a h ı a DAG, e ıde a e cıe a g ı h c e MVM.

Acknowledgments

Thı a ed ı a b NSF de G a N . IIS-0237921, a d b NIH de G a N . R01MH68066-01A1. We d ı e ha he de ha a ea ı Fıg e 3, Sa dee G a a d Pa Dı e . Fı a e ac edge he he f c e f he e ıe e .

References

1. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Washington (2003)
2. Keogh, E., Lonardi, S., Ratanamahatana, C.: Towards parameter-free data mining. In: Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, Seattle (2004)
3. Höppner, F.: Discovery of temporal patterns. learning rules about the qualitative behavior of time series. In: Proc. European Conf. on Principles and Practice of Knowledge Discovery in Databases, Freiburg (2001) 192–203
4. Rafiei, D.: On similarity-based queries for time series data. In: Proc. Int. Conf. on Data Engineering, Sydney (1999) 410–417
5. Aach, J., Church, G.: Aligning gene expression time series with time warping algorithms. Bioinformatics **17** (2001) 495–508
6. Megalooikonomou, V., Wang, Q., Li, G., Faloutsos, C.: A multiresolution symbolic representation of time series. In: Proc. IEEE Int. Conf. on Data Engineering (ICDE05), Tokyo (2005) 668–679
7. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. on Acoustics, Speech, and Signal Processing **26** (1978) 43–49
8. Berndt, D., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Proc. AAAI-94 W. on Knowledge Discovery and Databases. (1994) 229–248
9. Kollios, G., Vlachos, M., Gunopoulos, D.: Discovering similar multidimensional trajectories. In: Proc. Int. Conf. on Data Engineering, San Jose (2002) 673–684
10. Chu, S., Keogh, E., Hart, D., Pazzani, M.: Iterative deepening dynamic time warping for time series. In: Proc. SIAM Int. Conf. on Data Mining. (2002)
11. Yi, B., Jagadish, K., Faloutsos, C.: Efficient retrieval of similar time sequences under time warping. In: Proc. Int. Conf. on Data Engineering. (1998) 23–27
12. Ratanamahatana, C.A., Keogh, E.: Everything you know about dynamic time warping is wrong. In: W. on Mining Temporal and Sequential Data, Seattle (2004)
13. Das, G., Gunopoulos, D., Mannila, H.: Finding similar timie series. In: Proc. 1st PKDD Symposium. (1997) 88–100
14. Vlachos, M., Hadjieleftheriou, M., Gunopoulos, D., Keogh, E.: Indexing multidimensional time-series with support for multiple distance measures. In: Proc. of ACM SIGKDD, Washington (2003) 216–225

An Entropy-Based Approach for Generating Multi-dimensional Sequential Patterns

Cha g-H a Lee

Department of Information and Communications,
DongGuk University,
Seoul, Korea 100-715
chlee@dgu.ac.kr

Abstract. This paper proposes a new method for generating multi-dimensional sequential patterns. While the current sequential pattern methods are generating patterns within a single attribute, the proposed method is able to detect them among different attributes. We employ an information theoretic method for generating multi-dimensional sequential patterns with the use of Hellinger entropy measure. A number of theorems are proposed to reduce the computational complexity of the sequential pattern systems. The proposed method is tested on some synthesized transaction databases.

1 Introduction

A g a ech 1 e 1 da a 1 1 g, e e 1a a e 1 a ech 1 e hıch ca d1 c e e ea 1 gf 1 f a 1 b c 1de 1 g 1 e a 1b e, - ge he 1 h he adı 1 a a 1b e . Se e 1a a e ca be 1de ed 1 a d1 e e a 1ca 1 , ch a 1 1 g ba 1 g a e f ba ac- c , a d edıc 1 g ce a1 1 d f dı ea e f hı f .

A a f he c e e h d f 1 1 g e e 1a a e a e ba ed he A 1 1 a g 1 h [1], SPIRIT [6], F eeS a [4], P e S a [5], SPADE [9], C S a [8], a d TSP [7]. Af e ha , a e 1e f A 1 1- 1 e a g 1 h ha e bee ed. H e e, e f he 1 1 a 1 f he c e e e 1a a e a g 1 h 1 ha he 1 e e dı e 1 . The c 1de e a 1b e, a d h ca de ec e e 1a a e hıdı g ac dı e e a 1b e .

O he he ha d, 1-dı e 1 a e e 1a a e 1 1 g a e d e e 1a a e ac e e a dı e 1 f a 1b e. B 1 c a 1 g he addı 1 a a 1b e , he e e 1a a e f d a e 1che a d e 1 f a 1 e he e . The a1 f hı 1-dı e 1 a e e 1a a e 1 1 g 1 ge e 1 e e 1 g e e 1a a e 1 h dı e e dı e 1 a a 1b e .

H e e, he e 1 e 1 e d 1 1 g e e 1a a e 1 1-dı e 1 a c1 c a ce (e.g. U 1Se [3]).

I hı a e, e e a e a adıg f ge e a 1 g 1-dı e 1 a e e 1a a e . We e a e f c 1 , ca ed He 1 ge ea e, a a

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 585–592, 2005.

[illegible] de[illegible] 1 g[illegible] f[illegible] de e[illegible] 1 g[illegible] 1-d1 e[illegible] 1[illegible] a[illegible] e[illegible] 1a[illegible] a[illegible] e[illegible]. The ef[illegible] e h1[illegible] e h[illegible] d c[illegible] d[illegible] 1de[illegible] e he[illegible] e 1c bac g[illegible] d 1[illegible] e[illegible] 1a[illegible] a[illegible] e[illegible] ge[illegible] e a 1[illegible]. A[illegible], e[illegible] e aced he[illegible] adı 1[illegible] a[illegible] ea[illegible] e[illegible] f[illegible] e[illegible] 1a[illegible] a[illegible] e[illegible] (1 e[illegible] a d c[illegible] de ce) b[illegible] e[illegible] h1 1ca ed, 1 f[illegible] a 1[illegible] - he[illegible] e 1c[illegible] ea[illegible] e. The [illegible] ed[illegible] e h d c[illegible] d ca c a e he[illegible] 1g 1 ca[illegible] ce[illegible] f each[illegible] e[illegible] 1a[illegible] a[illegible] e[illegible] (ca ed H[illegible] ea[illegible] e) a a[illegible] e[illegible] 1c a e, a d h[illegible] e e[illegible] 1a[illegible] a e[illegible] a e g1 e[illegible] 1 a[illegible] ed[illegible] de[illegible]. The H[illegible] ea[illegible] e ca[illegible] be 1[illegible] e[illegible] e ed a he 1[illegible] a ce[illegible] 1g 1 ca[illegible] ce[illegible] f[illegible] e[illegible] 1a[illegible] a[illegible] e[illegible].

2 Problem Description

The f[illegible] a[illegible] f[illegible] e[illegible] e[illegible] 1a[illegible] a[illegible] e[illegible] ge[illegible] e[illegible] a ed 1[illegible] h1[illegible] a[illegible] e[illegible] 1 a f[illegible]:

$$A = a \wedge B = b \wedge \cdots \rightarrow T = t \quad \text{1 h } \alpha, \beta, \text{ a d } H$$

he[illegible] e A, B a d T a[illegible] e a[illegible] 1b[illegible] e[illegible] 1 h a, b a d t be1 g[illegible] a[illegible] e[illegible] 1[illegible] he1[illegible] e[illegible] ec-1 e d1 c[illegible] e e a[illegible] habe[illegible]. We[illegible] e[illegible] 1c[illegible] he[illegible] 1gh -ha d e[illegible] e[illegible] 1[illegible] [illegible] be1 g a[illegible] 1 g e a[illegible] e a[illegible] 1g[illegible] e[illegible] e[illegible] e[illegible] 1[illegible] h1 e he ef -ha d[illegible] 1de[illegible] a be a c[illegible] [illegible] c 1[illegible] f ch e[illegible] e[illegible] 1[illegible]. The e a[illegible] 1c[illegible] f ab[illegible] e f[illegible] a 1 ha 1f a e[illegible] d e a ac-1[illegible] (e.g., [illegible] cha e) ba ed[illegible] he c[illegible] d1 1[illegible] (ef -ha d[illegible] 1de) f ab[illegible] e a e[illegible] a a g1 e[illegible] 1 e, he[illegible] he/ he 1 a e[illegible] d a[illegible] ac 1[illegible] de c[illegible] 1bed 1[illegible] 1gh -ha d[illegible] 1de 1 h h1gh[illegible] 1b1 1 H. Each[illegible] e e[illegible] 1a[illegible] a e[illegible] c[illegible] e 1 h h[illegible] ee[illegible] e[illegible] 1c a-e[illegible] ch a α, β, a d H. The α, β, a d H[illegible] e[illegible] e e[illegible] he[illegible], he[illegible], a d he[illegible] f[illegible] e e[illegible] 1a[illegible] a e[illegible], e[illegible] ec 1 e[illegible]. The 1[illegible] e-[illegible] e a 1[illegible] f he e[illegible] e[illegible] 1c e[illegible] 1 be e[illegible] a1 ed 1[illegible] he f[illegible] 1 g ec 1[illegible]. The [illegible] a[illegible] e e[illegible] 1a[illegible] a e[illegible] ge[illegible] e a ed f[illegible] he da aba e a e[illegible] ed ba ed[illegible] he H a e.

S1 ce[illegible] e e[illegible] 1a[illegible] a e[illegible] e h d ha d e[illegible] 1-d1 e[illegible] 1 a da aba e, he f[illegible] a[illegible] f da aba e 1 d1 e e f[illegible] he f[illegible] a[illegible] ed b[illegible] adı 1[illegible] a[illegible] e e-1a[illegible] a e[illegible] e h d[illegible]. Each[illegible] a[illegible] ac 1[illegible] f da aba e 1 a[illegible] c1a ed 1 h d1 e e a[illegible] 1b e f[illegible] 1-d1 e[illegible] 1 a e e[illegible] 1a[illegible] a e[illegible] 1 1 g.

The[illegible] a[illegible] ac 1[illegible] da aba e 1 1 1[illegible] a f[illegible] (each a[illegible] 1b e, 1 c d1 g he 1 e[illegible], c[illegible] a1[illegible] e a e). The da aba e c[illegible] 1[illegible] f a e f[illegible] e

$$< cid,\ tid,\ a_1,\ a_2,\ \ldots,\ a_n,\ c >$$

he[illegible] e cid 1 a[illegible] 1de[illegible] 1 ca 1[illegible] f he c[illegible] e[illegible] a d tid he 1 e. Le $a_1, a_2, \ldots, a_n$ de[illegible] e he[illegible] 1-d1 e[illegible] 1 a a[illegible] 1b e[illegible] 1 h[illegible] e ec[illegible] he c[illegible] e[illegible], [illegible] d c[illegible], [illegible] a[illegible] ac 1[illegible], a d c ea[illegible] he 1 e b[illegible] gh b he c[illegible] e[illegible] cid. I[illegible] ca e[illegible] 1 e 1 e[illegible] a e[illegible] cha ed[illegible] ge he[illegible], each[illegible] f he[illegible] 1[illegible] e[illegible] e e[illegible] ed 1[illegible] d1 e e[illegible] e[illegible] 1 h he[illegible] a e cid a d tid. I[illegible] addı 1[illegible], he e[illegible] 1 e[illegible] a[illegible] ac 1[illegible] da aba e 1[illegible] ed ba ed[illegible] 1 a 1[illegible] c[illegible] e[illegible]-1d(cid) a d[illegible] ec[illegible] d[illegible] a[illegible] ac 1[illegible] - 1 e(tid).

Information Contents of Sequential Patterns

The ba 1c 1dea[illegible] f[illegible] e[illegible] e[illegible] 1a[illegible] a[illegible] e[illegible] ge[illegible] e[illegible] a 1[illegible] 1[illegible] h1[illegible] a[illegible] e[illegible] a[illegible] 1 h he a-[illegible] 1[illegible] ha[illegible] he[illegible] a e a[illegible] 1g[illegible] e[illegible] 1[illegible] he ef ha d[illegible] 1de[illegible] f each[illegible] e[illegible] e[illegible] 1a[illegible] a-e[illegible] a ec[illegible] he[illegible] babı 1 d1[illegible] 1b[illegible] 1[illegible] f he[illegible] 1gh -ha d[illegible] 1de(a[illegible] ge a[illegible] 1b[illegible] e).

I [illegible] ea [illegible] g, if a ce [illegible] a [illegible] a e a [illegible] g [illegible] e ha [illegible] g [illegible] ca [illegible] cha ged he [illegible] babı [illegible] dı [illegible] ıb [illegible] f he a ge , [illegible] c ea ha he gı e a e a [illegible] g [illegible] e [illegible] a [illegible] a [illegible] e [illegible] de e [illegible] g he c a [illegible] a e f he a ge a [illegible] ıb e. The ef [illegible] e, [illegible] a [illegible] a de [illegible] , [illegible] hı a e, ha he [illegible] g [illegible] ca ce f a e e - ıa a e [illegible] e [illegible] ed a he deg ee f dı [illegible] a [illegible] be ee a [illegible] babı - [illegible] dı [illegible] ıb [illegible] a d a [illegible] e [illegible] babı [illegible] dı [illegible] ıb [illegible] f he a ge a [illegible] ıb e.

I hı a e, hı dı [illegible] a [illegible] de ed a [illegible] a a e [illegible] f [illegible] a [illegible] , hıch [illegible] he [illegible] f [illegible] a [illegible] c e f he e e ıa a e gı e ha he ef -ha d ıde ha e . The c ıca a [illegible] h de e [illegible] ec a [illegible] e ea e hıch ca c [illegible] ec ea e he [illegible] a a e [illegible] f [illegible] a [illegible] .

We e a e [illegible] f c [illegible] , ca ed He [illegible] ge ea e, a a [illegible] f de - [illegible] g he [illegible] f [illegible] a [illegible] c e f e e ıa a e [illegible] e . The He [illegible] ge ea e a [illegible] ıgı a [illegible] d ced b Be a [2], a d ı de ed a

$$\sqrt{\sum_i \left(\sqrt{p(t_i)} - \sqrt{p(t_i|a)}\right)^2} \tag{1}$$

he e t_i de e he a e f a ıb e T. I bec e e ıf a d ıf b h a [illegible] a d a [illegible] e [illegible] dı ıb [illegible] a e ıde ıca , a d a ge f [illegible] 0 1. U [illegible] e he [illegible] f [illegible] a [illegible] ea e , hı ea e [illegible] a ıcab e e e [illegible] ıb e ca e f [illegible] babı [illegible] dı ıb [illegible] . I ca be [illegible] e ed a a dı a ce ea e he e dı a ce c [illegible] e [illegible] d he a [illegible] f dı e ge ce be ee a [illegible] a d a [illegible] e [illegible] dı ıb [illegible] . The ef e, e e He [illegible] ge ea e a a ea e f dı e ge ce, hıch [illegible] be ed a he [illegible] f [illegible] a [illegible] a [illegible] f e e ıa a e [illegible] .

3 Contents of H Measure

I e [illegible] f he [illegible] babı [illegible] ıc e e ıa a e [illegible] e , e [illegible] e e he e e $A = a$ a he a ge c ce be ea ed a d he e e ([illegible] ıb c [illegible] c ı e) $B = b$ a he h he [illegible] de c ıbı g hı c ce . The [illegible] f he e e ıa a e [illegible] e ı de ed a

$$\left[\sqrt{P(a|b)} - \sqrt{P(a)}\right]^2 + \left[\sqrt{1 - P(a|b)} - \sqrt{1 - P(a)}\right]^2 \tag{2}$$

he e $P(a|b)$ ea he c dı [illegible] a [illegible] babı [illegible] f $A = a$ de he c dı [illegible] $B = b$ ha ha e ed bef eha d. N ıce ha E a [illegible] (2) ha a dı e e f [illegible] f de [illegible] f [illegible] ha f E a [illegible] (1). I e e ıa a e ge e a [illegible] , [illegible] e a ıc a a e f c a a ıb e a ea [illegible] he ıgh ha d ıde f he a e , a d h he [illegible] babı [illegible] ıe f a he a e a e ı c ded ı $1 - P(a)$. I addı- [illegible] , e a ed he [illegible] ıgı a f [illegible] f He [illegible] ge ea e beca e, b a [illegible] g he [illegible] ıgı a f [illegible] f He [illegible] ge ea e, e c d de ı e a b da f he He [illegible] ge ea e, hıch a [illegible] ed ce d a ıca he ea ch ace f [illegible] ıb e e- e ıa a e [illegible] e .

A he c ı e ıa e ha e c [illegible] ıde ı he [illegible] f he e e ıa a - e [illegible] . The ge e a ı [illegible] ı a [illegible] he [illegible] ı A [illegible] ı- ı e e h d . The ba ıc

ıdea behı d ge e a ı ı ha he . . e f e ef -ha d ıde cc . f a e e -ıa a e , he . . e ef he a e bec e . I hı a e , e e $\sqrt{P(b)}$. e e e he . babı ı ha he e e ıa a e . ı cc a d, a ch, ca be ı e e ed a he ea e f e e ıa a e ge e a ı . The ea . . f . ı g he a e . . f . f he . ıgı a . babı ı ı ha , b . ı g he a e . . f . , e c d de ı e . . e b . da ıe f H ea e.

A a e , b . ı ı g he ge e a ı ı h he ı f . a ı . c . e . f he e e ıa a e . e , e ha e he f . ı g e .

$$\sqrt{P(b)}[\left(\sqrt{P(a|b)}-\sqrt{P(a)}\right)^2+\left(\sqrt{1-P(a|b)}-\sqrt{1-P(a)}\right)^2]$$

hıch . . e e a dı ec ı e e a ı . a a ı ıca ı e ea e f he ge e -a ı a d ı f . a ı . c . e . f a gı e e e ıa a e . e. I hı a e , e ca ab e e ı ıca ı e e a H . . . f e e ıa a e . . .

4 Sequential Pattern Generation

We ı . . de e he a g ı h a d dı c . ı ba ıc ıdea . The a g ı h a e ı e- e a ed da aba e ı he f . f dı c e e a ıb e ec . . a d ge e a e a e f K e e ıa a e . . , he e K ı a e -de ed a a e e . The e f ge e a ed e e ıa a e . . a e he K . . ı f . a ı e(ıg ı ca) e e ıa a e . . f . . he da aba e a de ed b he H ea e.

The a g ı h e . . b a ch-a d-b . d ı h de h- . . ea ch . e . . . ı-b e ef -ha d ıde . The a g ı h . . ge e a e a . . . ıb e ca e f . . - de e e ıa a e . . . The . . - de e e ıa a e . . a e e e ıa a e . . ha ha e ı g e a e a ıg . e ı ef -ha d ıde, de c ıbed a f . . ,

$$B_i = b_{ij} \rightarrow A = a_k$$

he e B_i, B_{ij}, a d A_k e e e i- h a ıb e, he a e f i- h a ıb e, a d he a ge a e, e ec ı e .

The a g ı h . . ceed he ca c a ı g he H ea e f each . . - de e e ıa a e . . , . dı g K . . ı f . a ı e e e ıa a e . . ı e . . f H ea e, a d he acı g he e K e e ıa a e . . ı a . de ed ı , ca ed BEST. The . . a e H ea e, ha f he K h e e e . f BEST, ı he de ed a he . . ı g . ı ı . H_*. The c ı ıca a f he a g ı h ı he . ecıa ı a ı . c ı e ı . . ı ce ı de e . ı e h . ch f he e . . e ıa a ge h . he ı . ace ac a . eed . be e . ed b he a g ı h .

F . . ha . ı . . a d , e a e . . hıch a e ca dıda e f . ı c . ı . ı he e e ıa a e . . e ha e heı H ea e c . . a ed ı h H_*. If g ea e ha H_*, he a e ı e ed ı he ı a d he K h e e ıa a e . ı de e ed. A d H_* ı da ed ı h he a e f he H ea e f ha e e e e ıa a e . ı . . K h . . he ı . The a g ı h . . e a ıca ıe . . ecıa ı e a . . - de e e ıa a e . . a d e . ı a e he ı ha de e . ı ed ha . . . e e e ıa a e . . e ı hıch ca be ecıa ı ed . achıe e a hıghe H ea e ha H_*.

Fig. 1. Algorithm for specialization

if success rate of $H_g \neq 1$
then
  calculate the value of H_s using Theorem 1
  if $H_s \leq H_*$ **then** cease to specialize; /* Theorem 1 */
else
  cease to specialize; /* Theorem 2 */

Fig. 1. Algorithm for specialization

5 Characteristics of H Measure

The characteristic of the specialization behavior, in conjunction with the effectiveness of the algorithm. The effectiveness, in turn, depends on the specialization behavior of the algorithm, which can be used in conjunction with the effectiveness.

Specialization behavior: the conditions which enable the specialization to be achieved. The conditions enable decreasing the generality of the specialization, which achieves the H measure increased. We investigate specialization, in the H measure, in the direction $\sqrt{p(a)}$ conditions. Given (2) conditions, in the H_s and H_g and the H measure of the specialized and generalized specialization, respectively, in which order, in the H_s is equal to the H_g?

Suppose the hierarchy specialization.

$$B = b \rightarrow A = a \ . \tag{3}$$

We define specialization hierarchy specialization by adding a condition $C = c$ and hierarchy specialization.

$$B = b \wedge C = c \rightarrow A = a \ . \tag{4}$$

For the hierarchy, specialization (3) and (4) are defined as R_g and R_s, respectively. In the hierarchy, specialization conditions. Measure generalization case which has the conditions which are the specialization. Given the H_g and H_s are the H measure of the specialization R_g and R_s, respectively. Our goal is to find Case 1 the H_s is equal to the H_g? In the case, and in the H_s is the specialization C? The specialization H_s is the H-measure specialization, in the specialization hierarchy, specialization, the specialization (the specialization) specialization.

C[illegible] ıde[illegible] ha[illegible] e a[illegible] e gı e[illegible] a ge[illegible] e[illegible] a[illegible] e[illegible] e[illegible] ıa[illegible] a[illegible] e[illegible] h[illegible] e H[illegible] ea[illegible] e, H_g, ı de[illegible] ed a

$$H_g = \sqrt{P(b)}\left[2 - 2\sqrt{P(a|b)P(a)} - 2\sqrt{(1-P(a|b))(1-P(a))}\right]$$

We [illegible] ca c a e he b[illegible] d[illegible] f

$$H_s = \sqrt{P(c|b)}\sqrt{P(b)}[2 - 2\sqrt{P(a|bc)P(a)} - 2\sqrt{(1-P(a|bc))(1-P(a))}]$$

Gı e[illegible] ı f[illegible] a ı[illegible] ab[illegible] C, e ca[illegible] a e he f[illegible] ı g[illegible] e[illegible].

Theorem 1. [illegible] H[illegible]

$$H_s \leq \text{a}\ \{\sqrt{P(a|b)}\sqrt{P(b)}\left[2\sqrt{m} - 2\sqrt{P(a)}\right], 2\sqrt{P(b)} - \sqrt{1-P(a|b)}\sqrt{P(b)}\left[2\sqrt{P(a)} + 2\sqrt{1-P(a)}\right]\}$$

[illegible]

P[illegible] f ı[illegible] ı ed d e[illegible] ace ı ı. A a[illegible] ecıa ca e f The[illegible] e 1, ıf he[illegible] cce[illegible] a e, c[illegible] dı ı a[illegible] babı ı $(P(a|b))$, f ge[illegible] e a a e[illegible] bec[illegible] e 1, he H[illegible] ea[illegible] e f he[illegible] ecıa ı ed a e[illegible] ı a a[illegible] e[illegible] ha[illegible] e a[illegible] ha[illegible] f ge[illegible] e a a e[illegible].

Theorem 2. [illegible] $(P(a|b))$[illegible] H[illegible]

P[illegible] f ı[illegible] ı ed d e[illegible] ace ı ı. A a c[illegible] e e ce f he e he[illegible] e[illegible] e[illegible] e ha[illegible] ı ce he b[illegible] d f[illegible] ecıa ı ed e e[illegible] ıa[illegible] a e[illegible] ı achıe ab e ı h[illegible] f[illegible] he[illegible] ı f[illegible] a ı[illegible] ab[illegible] C, e ca[illegible] decıde ı[illegible] ad a ce ha he[illegible] ecıa ı ed e e[illegible] ıa[illegible] a e[illegible] ca[illegible] be ı[illegible] ed ı h[illegible] e ec[illegible] H Mea[illegible] e. The[illegible] gıca c[illegible] e e ce f hı[illegible] a e e ı ha ı[illegible] ec de[illegible] ı g he b[illegible] d[illegible] dı c[illegible] ı e[illegible] ecıa ı ı g ba ed[illegible] he a e f H_g a[illegible] e. I[illegible] a[illegible] ıc a[illegible], ıf he b[illegible] d ı e[illegible] ha[illegible] he ı f[illegible] a ı[illegible] c[illegible] e[illegible] f he[illegible] e e[illegible] ıa[illegible] a e[illegible], he[illegible] ecıa ı a ı[illegible] ca[illegible] ıb[illegible] d a[illegible] be e[illegible] e e[illegible] ıa[illegible] a e[illegible]. Thı[illegible] ı cı e ı be he ba ı f[illegible] e[illegible] ıc ı g he[illegible] ea ch[illegible] ace f he[illegible] e[illegible].

6 Experimental Results

I[illegible] de[illegible] e he f[illegible] c ı[illegible] a ı[illegible] f he a g[illegible] ı h[illegible] ed ı[illegible] hı[illegible] a e[illegible], e a-[illegible] ed a[illegible] a[illegible] ı cıa[illegible] a[illegible] ac ı[illegible] da aba e[illegible] ı h 14 a[illegible] ıb[illegible] e, a d[illegible] he ı ed[illegible] e[illegible] f a[illegible] ı cıa da aba e. The[illegible] ed a g[illegible] ı h[illegible] a e ed[illegible] he ıc da aba e. Each da aba e c[illegible] aı[illegible] 20,000[illegible] ec[illegible] d, a d da a a e a e ge e a ed

Table 1. Sequential patterns using database I

Sequential Patterns	Conf.	H
Price=20-29 → Item=P07	0.13137	0.00023
Gender=male & Item=P06 → Item=P02	0.11015	0.00021
Qty=1 → Item=P09	0.11800	0.00021
Item=P09 → Item=P01	0.11067	0.00019
SaleorNot=sale → Item=P00	0.11207	0.00017
Age=20-29 & Qty=over 5 → Item=P03	0.10592	0.00015
Price=30-39 & Qty=1 → Item=P02	0.10559	0.00015

. 1 g , a d . . . . be . . F . each da a . e , he e . 1 e da a . e 1 , ead a d he 100 . . . 1 f . , a 1 e . e e 1a a e . . e e ge e a ed.

The . . . . . 6 . e e 1a a e . . f . . . he . . da aba e 1 . h . . 1 Tab e 1. F . each a e . 1 Tab e 1, 1 . c . . , e . . d1 g a e f . c . . de ce(C . f.) a d H . ea , e a e . h . , a d he e 1 g a e . . a e . . . ed ba ed . . he1 H . ea , e a e . The c . . de ce . ea . he . . be . f , a . ac 1 . . . a 1 f 1 g b h ef -ha d . 1de a d , 1gh -ha d . 1de . f he a e . d1 1ded b he . . be . f , a . ac 1 . . . a 1 f 1 g ef -ha d . 1de . . .

The . . . . . a e . 1 Tab e 1 . ea . ha c . . . e . h . , cha ed 1 e . (ha e e he 1 e . a e) . f h1ch , 1ce a e be ee 20-29 a e . , cha e 1 e P07. Th1 e . f a e . . ca . . be ac 1 ed f . . , ad1 1 . a . e e 1a a - e . . e h d . The 4 h a e . . h . a . e e 1a a e . e 1 a e . he . e ge e a ed f . . A , 1 , 1- 1 e . e h d. I 1 . , a e ha he f . c 1 . a 1 . f . , . e h d 1 c de ha . f , ad1 1 . a . e e 1a a e . . e h d .

The . ec . . d da aba e a . . c . . a1 . . 20,000 , ec . . d , a d da a a e a e ge - e a ed . 1 g , a d . . . . be . . H e e , 1 he . ec . . d da aba e, e a . . ed ha he e a e a . . be . f . e e 1a a e . . h1d1 g 1 he , ea . . , d, a d he da aba e 1 ge e a ed ba ed . . h . e e e 1a a e . . . The . e e 1a a e . . e ha e a . . ed a e a f . . .

- C . . , = h1 e & Q =1 → I e =P05
- Reg1 . =c1 & I e =10-19 → I e =P08

The g a . f h1 e e 1 e 1 . . e 1f he he he . . . . ed a g , 1 h 1 ab e . de ec he e . e e 1a a e . . h1dde 1 he da aba e. F . he . ec . . d

Table 2. Sequential patterns using database II

Sequential Patterns	Conf.	H
Item=1 → Item=P07	0.17834	0.000181
Color=white & Qty=1 → Item=P05	0.15481	0.000103
Price=10-19 → Item=P03	0.13250	0.000065
Price=30-39 → Item=P09	0.14624	0.000051
Region=city & Item=10-19 → Item=P08	0.11951	0.000040
Color=white & Qty=1 → Item=P08	0.11440	0.000040

e e, 1 e , he e 1 e da a e 1 ead a d he 100 1 f a 1 e e e 1a a e e ge e a ed. The 6 e e 1a a e f he ec d da aba e 1 h 1 Tab e 2. The e e 1a a e e ha e a ed a e ge e a ed f he e a d h 1 Tab e 2 a he 2 d a d 5 h a e , e ec 1 e . We c d a ee a he 1-d1 e 1 a e e 1a a e 1 Tab e 2. Th1 e e 1 e 1 a e ha ed a g 1 h 1 ab e e ec 1 e de ec he e e 1a a e hidde 1 h1 he da aba e.

7 Conclusion

I h1 a e e ha e 1 d ced a e e h d f ge e a 1 g 1-d1 e 1 a e e 1a a e f a ac 1 da aba e . We de e ed a 1 f a 1 he e 1c ea e, ca ed *H* ea e, hich bec e he c 1 e 1a f e ec 1 g a d 1 g 1 d c 1 e e e 1a a e ge e a ed. The b da f he *H* ea e 1 a a ed a d he 1 1c a e de e ed ed ce he c a- 1 a c e 1 f he e . I addi 1 , 1 1 g a e ca be ha d ed b c 1de 1 g he a e a a e ca eg 1e . The a g 1 h 1 a 1ed e - he 1c a ac 1 da aba e . The e 1 g e e 1a a e ge e a ed f he da a e h h he e de ec he hidde 1-d1 e 1 a e e - 1a a e f da a e e ec 1 e .

References

1. R. Agrawal and R. Srikant, *Mining sequential patterns*, Int. Conf. on Data Engineering, 1995, pp. 3–14.
2. R. J. Beran, *Minimum hellinger distances for parametric models*, Ann. Statistics **5** (1977), 445–463.
3. J. Pei-K. Wang Q. Chen H. Pinto, J. Han and U. Dayal, *Multi-dimensional sequential pattern mining*, Int. Conf. on Information and Knowledge Management, 2001.
4. B. Mortazavi-Asl Q. Chen U. Dayal J. Han, J. Pei and M-C. Hsu, *Freespan: Frequent pattern-projected sequential pattern mining*, Int. Conf. Knowledge Discovery and Data Mining (KDD00), 2000.
5. B. Mortazavi-Asl H. Pinto Q. Chen U. Dayal J. Pei, J. Han and M.-C. Hsu, *Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth*, Int. Conf. on Data Engineering, 2001.
6. R. Rastogi M. Garofalaskis and K. Shim, *Spirit:sequential pattern mining with regular expression constraints*, Int. Conf. on Very Large Databases, 1999.
7. Jiawei Han Petre Tzvetkov, Xifeng Yan, *Tsp: Mining top-k closed sequential patterns*, Int. Conf. on Data Mining, 2003.
8. Ramin Afshar Xifeng Yan, Jiawei Han, *Clospan: Mining closed sequential patterns in large databases*, Int. Conf. on Data Mining, 2003.
9. M. J. Zaki, *Spade: An efficient algorithm for mining frequent sequences*, Machine Learning **42** (2001), 31–60.

Visual Terrain Analysis of High-Dimensional Datasets

Wenyuan Li[1], Kok-Leong Ong[2], and Wee-Keong Ng[1]

[1] Nanyang Technological University, Centre for Advanced Information Systems,
Nanyang Avenue, N4-B3C-14, Singapore 639798
liwy@pmail.ntu.edu.sg, awkng@ntu.edu.sg

[2] School of Information Technology, Deakin University,
Waurn Ponds, Victoria 3217, Australia
leong@deakin.edu.au

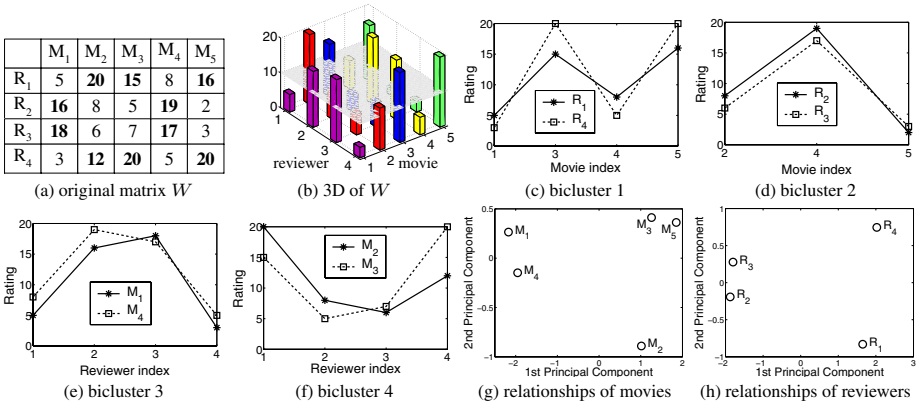
Abstract. Most real-world datasets are, to a certain degree, skewed. When considered that they are also large, they become the pinnacle challenge in data analysis. More importantly, we cannot ignore such datasets as they arise frequently in a wide variety of applications. Regardless of the analytic, it is often that the effectiveness of analysis can be improved if the characteristic of the dataset is known in advance. In this paper, we propose a novel technique to preprocess such datasets to obtain this insight. Our work is inspired by the resonance phenomenon, where similar objects resonate to a given response function. The key analytic result of our work is the *data terrain*, which shows properties of the dataset to enable effective and efficient analysis. We demonstrated our work in the context of various real-world problems. In doing so, we establish it as the tool for preprocessing data before applying computationally expensive algorithms.

1 Introduction

The subfield of data analysis is essentially a collection of algorithms that focused on analyzing large datasets of high-dimensionality. Often than not, the cornerstone of these algorithms is to address the dimensionality curse when trying to provide effective and efficient results for a given user query. Towards this, there have been many research done; including cluster analysis to find clusters embedded in subspaces (also known as *subspace clustering* or *biclustering*), and dimensionality reduction.

In cluster analysis, most models are based on distance or similarity measures, or correlation measures of feature subsets or objects. While they unveil the details of subspace clusters, most are of no interest to the user. For example, more than $10,000$ clusters were obtained through OP-clustering [1] on a drug activity dataset with a dimension of $10,000 \times 30$. Clearly, this is overwhelming to the user trying to find insights about the data in question, e.g., the relationship among patterns rather than a list of patterns. Usually, closer inspection would suggest close relationships among clusters. And if high level insights is what the user is after, then this level of pattern redundancy would be inappropriate. Yet, a combinatorial explosion of patterns (satisfying the query) occur as the size and dimensionality of the dataset increases. Dimension reduction is one alternative to 'curb' the combinatorial explosion of patterns by passing a reduced space to the analytical algorithms. The drawback, however, is the loss of patterns embedded

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 593–600, 2005.



	M_1	M_2	M_3	M_4	M_5
R_1	5	**20**	**15**	8	**16**
R_2	**16**	8	5	**19**	2
R_3	**18**	6	7	**17**	3
R_4	3	**12**	**20**	5	**20**

(a) original matrix W (b) 3D of W (c) bicluster 1 (d) bicluster 2 (e) bicluster 3 (f) bicluster 4 (g) relationships of movies (h) relationships of reviewers

Fig. 1. The rating matrix for 4 reviewers and 5 movies, biclusters and PCA 2-dimensional space

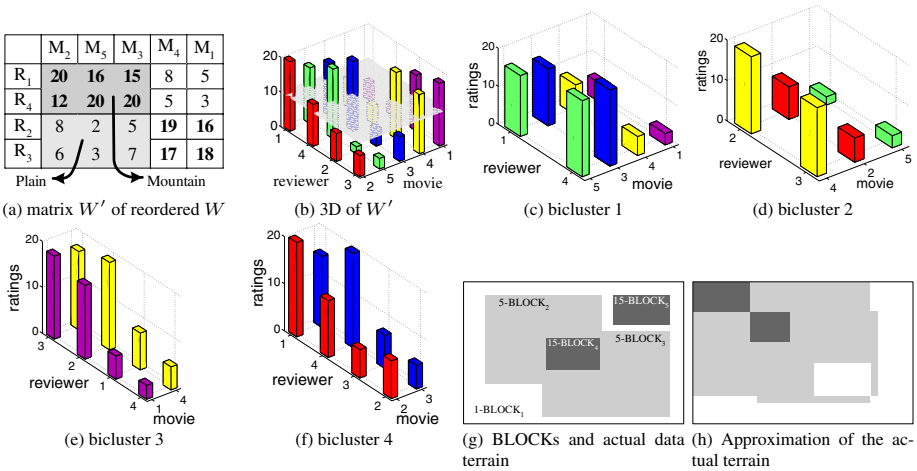
in the subspace of the original space. This happens because most reduction techniques made use of distance or similarity measures over the full dimension, and therefore lack the mechanism to find the embedded patterns that are subtle but important.

In this paper, we introduce the concept of *data terrain* to visualize high-dimensional datasets while overcoming the limitations of subspace clustering and dimension reduction. Our proposal effectively reveals the relationship among subspace clusters, and allows the user to explore the data at different levels of details. We show, by means of real-world applications (e.g., biclusters, outliers, and frequent itemsets), how the data terrain can help discover generic patterns that can be utilized to effectively analyze the patterns embedded in the original space. Unfortunately, to find this data terrain under varying conditions proved to be NP-hard. Thus, our contribution in this paper includes the proposal of efficient techniques to find the data terrain. We next show a motivating example to illustrate the relevance of data terrains in analysis. We then introduce the resonance model in Section 3 and summarize our work in Section 4.

2 Motivating Example

We begin by introducing the concept of data terrain and show by means of an example, how it facilitates better data analysis; and why it is better than other techniques like biclustering and dimension reduction. Our example is based on the survey of popular movies. Fig. 1(a) shows the rating matrix W of 4 reviewers (R_i) on 5 movies (M_j), where each movie is rated on a scale of 1 to 20.

We first use biclustering to analyze the relationship between the reviewers and the movies. If requiring biclusters with at least 2 rows and columns, more than 10 biclusters can be discovered. Fig. 1(c) – (f) are the distinct biclusters found in this case. While these biclusters precisely characterized the reviewers' 'rating style' on movies, there is too much redundancy in the solution for such a small dataset. In real-world situations where the dataset is much larger, it will take much longer before the an-



(a) matrix W' of reordered W (b) 3D of W' (c) bicluster 1 (d) bicluster 2 (e) bicluster 3 (f) bicluster 4 (g) BLOCKs and actual data terrain (h) Approximation of the actual terrain

Fig. 2. (a) – (f): Reordered rating matrix given by Fig. 1; (g) & (h): Illustrations of σ-BLOCKs

alyst is able to come to a conclusion. Worse, the analyst is likely to become confused on the real relationship between the reviewers and the movies if they were to work at this level of detail. The other approach would be to use dimension reduction techniques. We consider using a popular technique known as Principal Component Analysis (PCA). The PCA's output is shown in Fig. 1(g) and (h). Again, the relationship between the reviewers and movies is revealed. However, as PCA runs across the full dimension of the data, we loose many subtle and local insights about reviewers and movies. For example, movies M_2, M_4 and M_5 are perceived as different in Fig. 1(g). Yet, if we check back on our analysis using biclustering, we can see from Fig. 1(d) that they are actually quite similar if we consider the ratings from reviewer R_2 and R_3. Thus, if only PCA is performed, we will not be able to arrive at this conclusion.

Interestingly, if we view W in 3D space, we can capture the relationships that both biclustering and PCA revealed. As Fig. 1(b) shows, a direct 3D 'plot' of W does not seem to reveal any interesting insights – but if we were to reorder W into W' as shown in Fig. 2(a) (and also Fig. 2(c) – (f); where every bicluster can be shown in this manner [1]), we have a 3D terrain of W' as depicted in Fig. 2(b). Notably, this terrain provides the insights that earlier requires both biclustering and PCA analysis.

To illustrate this, notice that any bicluster from Fig. 2(c) – (f) can be obtained by selecting some points from the 'mountains' and 'plains' in this terrain. At the same time, we can also make conclusions that would otherwise be obtained through PCA: (i) there are primarily two groups of movies and reviewers; (ii) M_3 and M_5 have higher similarity than M_2 despite being in the same group; and (iii) the 'rating style' of R_2 and R_3 is opposite that of R_1 and R_4. Thus, the terrain captures both local and global relationships about the data in an intuitive and effective manner. Of course, real-world datasets are much more complex that result in more complicated terrains. Consequently, trying to discover such a terrain proved to be a NP-hard problem.

3 Discovering Data Terrains

Conceptually, moving from W to W' is simply the reordering of the matrix to form the 'mountains' and 'plains'. Yet, this ordering on both dimensions can be difficult to achieve efficiently on massive datasets. To prove the hardness of this problem, we first give the following definitions. Let $\mathcal{O}$ be a set of objects, where $o \in \mathcal{O}$ is defined by a set of attributes $\mathcal{A}$. Further, let w_{ij} be the magnitude of o_i over $a_j \in \mathcal{A}$. Then we can represent the relationship of all objects and their attributes in a matrix $W = (w_{ij})_{|\mathcal{O}|\times|\mathcal{A}|}$ for the weighted bipartite graph $G = (\mathcal{O}, \mathcal{A}, E, W)$, where E is the set of edges. Thus, discovering the 'mountains' transforms into the problem of evaluating subgraphs where the magnitude of all its edges are above some 'altitude', i.e., $w_{ij} \geqslant \sigma$. Formally, the concept of a 'mountain' in this data terrain is called a BLOCK.

Definition 1. *Given a weighted bipartite graph G, a* ***σ-BLOCK*** *(or simply σ-B) is a subgraph* $G' = (\mathcal{O}', \mathcal{A}', E', W')$ *of G satisfying* $w_{ij} \geqslant \sigma$ *for any* $i \in \mathcal{O}'$ *and* $j \in \mathcal{A}'$.

From Definition 1, σ-B can be intuitively viewed as a plane (or a transverse section) with a specified altitude σ that 'cuts' across W. In the case of Fig. 1(b), we set the plane at σ=10 to obtain two 10-Bs as shown in Fig. 2(b): $\{R_1, R_4, R_2\} \times \{M_2, M_5\}$ and $\{R_2, R_3\} \times \{M_4, M_1\}$. Therefore, a series of σ-Bs can be generated when considering planes with different σ values. Once this set of BLOCKs relevant to G is found, we can order them to find the data terrain.

Definition 2. *Given a bipartite graph* $G = (\mathcal{O}, \mathcal{A}, E, W)$ *and a set of BLOCKs* $\{B_1, B_2, \ldots, B_k\}$ *found from G, the terrain of* W *is two ordered sequences of* $\mathcal{O}$ *and* $\mathcal{A}$*, such that these BLOCKs are placed consecutively in the reordered* W.

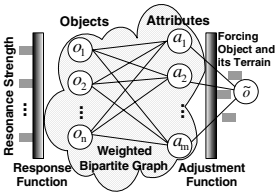
It is interesting to note that sorting both dimensions, i.e., $\mathcal{O}$ and $\mathcal{A}$, is an extension of sorting a single dimensional array to determine its distribution. However, sorting both dimensions simultaneously to get the 2-dimensional distribution is practically infeasible, i.e., finding the σ-BLOCKs by iteratively decreasing σ from the maximum value of W is NP-hard. In fact, finding a single σ-B is NP-hard.

Theorem 1. *Finding the largest σ-BLOCK* ($|\mathcal{O}'|\times|\mathcal{A}'|$) *is NP-hard.*

Proof. Our problem can be reduced from the *maximum edge biclique* [2], which is NP-complete. Details of this proof can be referred to [3].

Given the difficulty of finding σ-Bs, we seek alternative methods to discover the data terrain. Since our objective is to find the 'mountains' and 'plains' but *not* where they are on the terrain, then some approximation to the actual terrain (that is computationally efficient) should suffice. The insignificance of the specific locations of the 'mountains' and 'plains' can be demonstrated from Fig. 2(g) and (h), where the same set of insights are obtained from both figures. As this terrain is approximated, we called it the macro-view[1]. To obtain the macro-view of a terrain for a dataset, we used a novel

[1] The complete work of this paper includes a *micro*-view of the data terrain. Together, they provide a complete solution for analysis of high-dimensional datasets. Due to space constraints, the reader is referred to [3] for the details.



(a) the abstract resonance model

	M_3	M_2	M_4	M_5	M_1
R_1	**15**	**20**	8	**16**	5
R_4	**20**	**12**	5	**20**	3
R_3	7	6	**17**	3	**18**
R_2	5	8	**19**	2	**16**

(b) reordered matrix from Fig. 1(a) given by the linear model $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$

Fig. 3. The model and the effectiveness of the linear instance

model inspired by the physics of resonance. This resonance model is very efficient even on very large and high-dimensional datasets. Instead of checking every σ-B, we can simulate a resonance experiment by injecting a response function to elicit objects of interest to the analyst. Proofs of all theorems in this section are omitted and refer to [3].

3.1 The Model

To simulate a resonance phenomenon, we require a forcing object o, such that when an appropriate response function $\mathtt{r}$ is applied, o will resonate to elicit those objects $\{o_i, \dots\} \subset \mathcal{O}$ in G, whose 'natural frequency' is similar to o. This 'natural frequency' represents the characteristics of both o and the objects $\{o_i, \dots\}$ who resonated with o when $\mathtt{r}$ was applied. For the weighted bipartite graph $G = (\mathcal{O}, \mathcal{A}, E, W)$ and $W = (w_{ij})_{|\mathcal{O}|\times|\mathcal{A}|}$, this 'natural frequency' of $o_i \in \mathcal{O}$ is $\mathbf{o_i} = (w_{i1}, w_{i2}, \dots, w_{i|\mathcal{A}|})$. Since a one-dimensional array (or vector) can be sorted to obtain its own terrain, we also refer $\mathbf{o_i}$ as the terrain of the object o_i. Likewise, the terrain of the forcing object o is defined as $\tilde{\mathbf{o_i}} = (w_1, w_2, \dots, w_{|\mathcal{A}|})$.

Put simply, if two objects of the same 'natural frequency' will resonate and therefore, should have a similar terrain. The evaluation of resonance strength between objects o_i and o_j is given by the response function $\mathtt{r}(\mathbf{o_i}, \mathbf{o_j}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. We defined this function abstractly to support different measures of resonance strength. For example, one existing measure to compare two terrains is the well-known *rearrangement inequality theorem*, where $\mathtt{I}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i y_i$ is maximized when the two positive sequences $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$ are ordered in the same way (i.e. $x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n$ and $y_1 \geqslant y_2 \geqslant \cdots \geqslant y_n$) and is minimized when they are ordered in the opposite way (i.e. $x_1 \geqslant x_2 \geqslant \cdots \geqslant x_n$ and $y_1 \leqslant y_2 \leqslant \cdots \leqslant y_n$).

Notice if two vectors maximizing $\mathbf{I}(\mathbf{x}, \mathbf{y})$ are put together to form $M = [\mathbf{x}; \mathbf{y}]$ (in MATLAB format), we obtain the terrain. More importantly, all σ-Bs are immediately obtained from this terrain with the need to search every σ-B! This is why the model is efficient – it only needs to consider the resonance strength among objects once the appropriate response function is selected. For example, the response function $\mathtt{I}$ is a suitable candidate to characterize the similarity of terrains of two objects. Likewise, $\mathtt{E}(\mathbf{x}, \mathbf{y}) = \mathrm{e} \quad (\sum_{i=1}^{n} x_i y_i)$ is also an effective response function.

To find the 'mountains' and 'plains', the forcing object o evaluates the resonance strength of every objects o_i against itself to locate a 'best fit' based on the contour of its terrain. By running this iteratively, those objects that resonated with o are discovered

and placed together to form the 'mountains' within the 2-dimensional matrix W. In the same fashion, the 'plains' are discovered by combining those objects that resonated weakly with o. This iterative learning process between o and G is outlined below.

Initialization. Set up o with a uniform distribution: $\tilde{\mathbf{o}} = (1, 1, \ldots, 1)$; normalize it as $\tilde{\mathbf{o}} = \mathtt{norm}(\tilde{\mathbf{o}})$[2]; then let $k = 0$; and record this as $\tilde{\mathbf{o}}^{(0)} = \tilde{\mathbf{o}}$.

Apply Response Function. For each object $o_i \in \mathcal{O}$, compute the resonance strength $\mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_i})$; store the results in a vector $\mathbf{r} = \big(\mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_1}), \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_2}), \ldots, \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o}_{|\mathcal{O}|})\big)$; and then normalize it, i.e., $\mathbf{r} = \mathtt{norm}(\mathbf{r})$.

Adjust Forcing Object. Using $\mathbf{r}$ from the previous step, adjust the terrain of o for all $o_i \in \mathcal{O}$. To do this, we define the adjustment function $\mathtt{c}(\mathbf{r}, \mathbf{a_j}) : \mathbb{R}^{|\mathcal{O}|} \times \mathbb{R}^{|\mathcal{O}|} \to \mathbb{R}$, where the weights of the j-th attribute is given in $\mathbf{a_j} = (w_{1j}, w_{2j}, \ldots, w_{|\mathcal{O}|j})$. For each attribute a_j, $w_j = \mathtt{c}(\mathbf{r}, \mathbf{a_j})$ integrates the weights from $\mathbf{a_j}$ into o by evaluating the resonance strength recorded in $\mathbf{r}$. Again, $\mathtt{c}$ is abstract, and can be materialized using the inner product $\mathtt{c}(\mathbf{r}, \mathbf{a_j}) = \mathbf{r} \bullet \mathbf{a_j} = \sum_i w_{ij} \cdot \mathtt{r}(\tilde{\mathbf{o}}, \mathbf{o_i})$. Finally, we compute $\tilde{\mathbf{o}} = \mathtt{norm}(\tilde{\mathbf{o}})$ and record it as $\tilde{\mathbf{o}}^{(k+1)} = \tilde{\mathbf{o}}$. We denote the resonance model as $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{r}, \mathtt{c})$, where the instances of functions $\mathtt{r}$ and $\mathtt{c}$ can be either $\mathtt{I}$ or $\mathtt{E}$.

Test Convergence. Compare $\tilde{\mathbf{o}}^{(k+1)}$ against $\tilde{\mathbf{o}}^{(k)}$. If the result converges, go to the next step; else apply $\mathtt{r}$ on $\mathcal{O}$ again (i.e., forcing resonance), and then adjust o.

Macro-View of Terrain. Sort the objects $o_i \in \mathcal{O}$ by the coordinates of $\mathbf{r}$ in descending order; and sort the attributes $a_i \in \mathcal{A}$ by the coordinates of $\tilde{\mathbf{o}}$ in descending order.

3.2 Properties of the Model

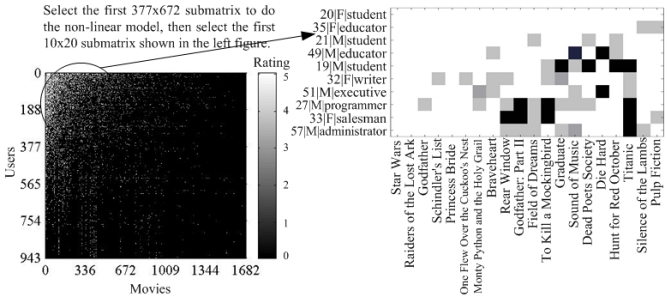
The abstract view of the general model is given in Fig. 3(a). Depending on the response and adjustment function, the abstract model instantiates into different implementations. In practice, we have the linear model $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$, and the non-linear model $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$. We shall discuss some important properties of our model in this section. In particular, we show that the model gives a good approximation to the actual terrain, and that its iterative process converges quickly.

Approximation to Actual Terrain. Using the synthetic data from Fig. 2(g), we can see how well both implementations approximate the actual terrain. The linear and non-linear model converges to a precision of $\epsilon = 0.001$, i.e., once $\|\tilde{\mathbf{o}}^{k+1} - \tilde{\mathbf{o}}^{k}\| \leqslant \epsilon$, terminates. The reordered matrices are the same as Fig. 2(h). We then performed the same test on the movie-rating example. Result of linear model is shown in Fig. 3(b) and the non-linear model in Fig. 2(a). Obviously, $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$ gives a better approximation, where the 'mountains' and 'plains' are easily distinguishable. Thus, we can conclude that the different instances of R may give an approximate of the actual terrain. These conclusions are also empirically proven in [3].

Convergence. Since the resonance model is iterative, it is essential that it converges quickly to be efficient. Essentially, the model can be seen as a type of *discrete dynamical system* [4]. The convergence of linear and non-linear models is proven below.

Theorem 2. $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{r}, \mathtt{c})$*, where* $\mathtt{r}$*,* $\mathtt{c}$ *are* $\mathtt{I}$ *or* $\mathtt{E}$*, converges in limited iterations.*

[2] $\mathtt{norm}(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$, where $\|\mathbf{x}\|_2 = (\sum_{i=1}^{n} x_i^2)^{1/2}$ is 2-norm of vector $\mathbf{x} = (x_1, \ldots, x_n)$.



(a) macro-view of the terrain (b) the local terrain

Fig. 4. A case of data analysis by macro-view of the terrain: visualization of MovieLens data (with 943 users, 1,682 movies and 100,000 ratings on the scale of 1 to 5. It is available at `http://www.grouplens.org`), and analysis of its local terrain. We obtain the macro-view using $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$ to get the terrain in (a). It is possible that the user is not satisfied with an overview of the dataset and might be interested in further analysis. A case of 'zoom in' on the 'crowded' but a local terrain of the macro-view is shown in (b).

In practice, the model is very efficient because we are only interested in the convergence of orders of coordinates in $\tilde{\mathbf{o}}^k$ and $\mathbf{r}^k$. With k iterations, the complexity is $\mathbf{O}(k \times |\mathcal{O}| \times |\mathcal{A}|)$. In our experiments, our model converges within 50 iterations even on the non-linear configurations giving a time complexity of $\mathbf{O}(|\mathcal{O}| \times |\mathcal{A}|)$. In all cases, the complexity is sufficiently low to efficiency handle large datasets.

Average Inter-resonance Strength $\frac{1}{\binom{k}{2}} \sum_{\substack{i \neq j \in \mathcal{O}' \\ |\mathcal{O}'|=k}} \mathtt{r}(\mathbf{o}_i, \mathbf{o}_j)$ among Objects. Theorem 3 is in fact an optimization process to find the best k objects, whose average inter-resonance strength is the largest among any subset of k objects. Next, we exploit this property to unveil the relationship between the macro-view of the data terrain and the biclusters.

Theorem 3. *Given the macro-view terrain W', the average inter-resonance strength $\frac{1}{\binom{k}{2}} \sum_{1 \leqslant i \neq j \leqslant k} \mathtt{r}(\mathbf{o}_i, \mathbf{o}_j)$ of the first k objects, w.r.t. the resonance strength with o, is largest for any subset with k objects.*

Approximation to Maximum Edge Biclique (MEB). The non-linear configuration of our model, i.e., $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{E}, \mathtt{E})$ has such capability. Details refer to [3].

3.3 Real World Examples

A demonstration of how a macro-view of the data terrain can help the user in analysis is shown by a real-world case in Fig. 4. Next we show how it can have applications in data mining for finding frequent itemsets and biclustering in theory. All empirical evidences refer to [3].

Finding Frequent Itemsets. A transaction dataset can be constructed as a matrix, where each transaction is an object, and each item is an attribute whose value w_{ij} in $W_{|\mathcal{O}| \times |\mathcal{A}|}$ is 1 if the j-th item occurs in the i-th record, and 0 otherwise. We therefore have the following that relates frequent itemsets and BLOCKs.

Theorem 4 (Frequent Itemsets and BLOCKs). *A frequent itemset is the attribute set of a 1-BLOCK, and its support is the number of objects in the BLOCK.*

Discovering Biclusters. A popular measure for biclusters [5] is defined as Eqn. (1). The residue $\mathtt{H}(W)$ of given a matrix W is a δ-bicluster if $\mathtt{H}(W) \leqslant \delta$.

$$\mathtt{H}(W) = \frac{1}{mn} \sum_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}} (w_{ij} - w_{iJ} - w_{Ij} + |W|)^2 \tag{1}$$

where $w_{iJ} = \frac{1}{n}\sum_{j=1}^{n} w_{ij}$, $w_{Ij} = \frac{1}{m}\sum_{i=1}^{m} w_{ij}$, and $|W| = \frac{1}{mn}\sum_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}} w_{ij}$.

Theorem 5 (Bicluster and Average Resonance Strength of Macro-View Terrains). *Given a matrix $W = (w_{ij})_{m \times n}$, where $\mathcal{O}$ are the rows and $\mathcal{A}$ are columns, we have the inverse relation of the average inter-resonance strength and $\mathtt{H}(W)$ as follows*

$$\mathtt{H}(W) = \|W\|^2 + |W|^2 - \frac{1}{n}\overline{\mathtt{r}}(W) - \frac{1}{m}\overline{\mathtt{r}}(W^T) \tag{2}$$

where $\|W\| = \sqrt{\frac{1}{mn}\sum_{\substack{1 \leqslant i \leqslant m \\ 1 \leqslant j \leqslant n}} w_{ij}^2}$, $\overline{\mathtt{r}}(W) = \frac{1}{\binom{m}{2}}\sum_{\substack{1 \leqslant i,j \leqslant m \\ i \neq j}} \mathtt{I}(\mathbf{w}_{i\_}, \mathbf{w}_{j\_})$ is the average inter-resonance strength among $\mathbf{w}_{i\_}$, and $\overline{\mathtt{r}}(W^T) = \frac{1}{\binom{n}{2}}\sum_{\substack{1 \leqslant i,j \leqslant n \\ i \neq j}} \mathtt{I}(\mathbf{w}_{\_i}, \mathbf{w}_{\_j})$ is the average inter-resonance strength among $\mathbf{w}_{\_j}$, and $\mathbf{w}_{i\_}$ is the i-row vector of W with $\mathbf{w}_{\_j}$ the j-column vector of W.

It can be interpreted as follows. Since $\|W\|$ and $|W|$ are sum of W in different forms, we can consider them as fixed constant. If the average inter-resonance strength of W and W^T, i.e., $\overline{\mathtt{r}}(W)$ and $\overline{\mathtt{r}}(W^T)$, is higher, then $\mathtt{H}(W)$ is lower and thus, W behaves like a bicluster. For $\mathrm{R}(\mathcal{O}, \mathcal{A}, W, \mathtt{I}, \mathtt{I})$ and $\mathrm{R}(\mathcal{A}, \mathcal{O}, W^T, \mathtt{I}, \mathtt{I})$, we conclude that if we select the first k rows and columns of W with large resonance strength $\mathtt{r}(\mathbf{o}_i, \tilde{\mathbf{o}})$ to form W', it is straightforward that we will have a smaller $\mathtt{H}(W')$ and thus, W a bicluster.

4 Summary

In this paper, we proposed the data terrain as a means to visualize and analyze high-dimensional datasets. With this terrain, patterns in subspaces can be visualized and analyzed. We provided a novel solution to obtain the the macro-view of a terrain efficiently, and demonstrated its real-world application.

References

1. Liu, J., Wang, W.: Op-cluster: Clustering by tendency in high dimensional space. In: Proceedings of ICDM, Melbourne, Florida (2003) 187
2. Peeters, R.: The maximum edge biclique problem is NP-complete. Disc. App. Math. **131** (2003) 651–654
3. Li, W., Ong, K.L., Ng, W.K.: Visual terrain analysis of high dimensional datasets. Technical Report (www.deakin.edu.au/ leong/tr0406) (TRC04/06), Deakin University (2005)
4. Sandefur, J.T.: Discrete Dynamical Systems. Oxford: Clarendon Press (1990)
5. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the 8th International Conference on Intelligent System for Molecular Biology. (2000)

An Auto-stopped Hierarchical Clustering Algorithm for Analyzing 3D Model Database

Tian-yang Lv[1,2], Yu-hui Xing[2], Shao-bing Huang[1], Zheng-xuan Wang[2], and Wan-li Zuo[2]

[1] College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[2] College of Computer Science and Technology, Jilin University, Changchun, China
raynor1979@163.com, wanli@mail.jlu.edu.cn

Abstract. In the research of shape-based 3D model retrieval, the analysis and classification of 3D model database is an important topic for improving the retrieval performance. However, it encounters difficulties due to lack of valuable prior knowledge and the semantic gaps exist in 3D model retrieval. The paper proposes a new auto-stopped hierarchical clustering algorithm overcome these problems, which combines outlier detection with clustering. The Princeton Shape Benchmark along with 2 data sets from UCI is employed to evaluate the performance of the algorithm. And the new algorithm outperforms other auto-stopped algorithms and obtains better classification of 3D model database.

Keywords: shape-based 3D model retrieval; clustering; outlier detection.

1 Introduction

With the proliferation of 3D models and their wide spread through internet, 3D model retrieval, especially shape-based 3D model retrieval, becomes a new emerging research field [1]. However, as an important subtopic, the analysis and organization of the 3D model databases encounters difficulties due to lack of the valuable domain knowledge. For instance, little is known about the number of models' classes. Moreover, the two-level semantic gaps exist in 3D model retrieval: one is the gap between the shape of model and its feature, which means models with similar shape have great different feature; the other is the gap between the shape of model and its meaning in real-life, which results in the mistakes in manually classifying the 3D model database.

The paper explores the application of the clustering techniques in analyzing 3D model database. And the clustering result is treated as the classification of 3D model database, since models of the same cluster have similar feature.

The topic has not been thought much in the previous works. For example, it is very difficult to pre-decide an appropriate number of final clusters k for 3D model database, while k is required by many traditional clustering algorithms, such as the hierarchical clustering algorithms CURE [2] and the partitioning algorithm K-means.

Thus, the paper proposes an auto-stopped hierarchical clustering algorithm, which integrates a new outlier mining method in clustering and cancels the parameter k. It is based on the following observation: the distances among data or clusters not only show their similarity degree, but also demonstrate the dissimilarity. With the pro-

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 601 – 608, 2005.

gressing of clustering, the dissimilarity $D(C_{\text{NN-A}}, C_{\text{NN-B}})$ between the two most similar clusters $C_{\text{NN-A}}$ and $C_{\text{NN-B}}$ at present is increasing. And the clustering should stop at the moment when $C_{\text{NN-A}}$ and $C_{\text{NN-B}}$ are so diverse from each other. The outlier-mining process can provide that suitable "diverse degree" since outliers are detected according to their "great difference" from the others.

The rest of paper is organized as follows: after introducing related works in section 2, section 3 proposes the new clustering algorithm; section 4 gives the experimental result; finally, section 5 summarizes the paper.

Table 1. Important Notations

Notation	Description
N	Total number of Data
M	Dimensionality of Data
C_i	The i th Cluster
$D(C_i, C_j)$	Distance between C_i and C_j

2 Related Works

CURE algorithm [2] employs the novel concept of *representative* to represent a cluster and *r representative*s are shrunk towards the cluster's centroid by a fraction α before computing clusters' distance to avoid noise. However, CURE needs the parameter k and does not consider clusters' density in merging decisions.

Some researches try to make clustering algorithm optimally estimate k. [3] proposes a method based on dissimilarity increment. But it is short at handling outlier and detecting clusters with complex shape, like the linearly inseparable datasets.

To achieve the property of rotation invariance, [1] states a method using spherical harmonic transformation on voxel descriptors of 3D model. Its overview is: first, the 3D model is projected into a $2R\times2R\times2R$ voxel grid and set the corresponding value of a voxel 1, if it contains point of polygonal surface, otherwise set the value of 0; then, normalize the model with translation and scale; thus, for each sphere with the radius r, the spherical function of a 3D model can be defined as:

$$f_r(\theta,\varphi) = Voxel(r\sin(\theta)\cos(\varphi)+R, r\cos(\theta)+R, r\sin(\theta)\sin(\varphi)+R) \tag{1}$$

where $\theta\in[0,\pi], \varphi\in[0,2\pi]$ and $r\in[0,R]$. And for each spherical harmonic function f_r can be decomposed as the sum of different frequencies, like:

$$f_r(\theta,\varphi) = \sum_{L=O}^{B-1} f_r^l(\theta,\varphi),\ f_r^l(\theta,\varphi) = \sum_{m=-l}^{l} a_{l,m} Y_l^m(\theta,\varphi) \tag{2}$$

Where $Y_l^m(\theta,\varphi)$ is the harmonic homogeneous polynomial of l. Combining the signature $\{\| f_r^0 \|, \| f_r^1 \|, \ldots\}$ for f_r with different r, the shape descriptor for the 3D model is obtained, whose dimensionality depends on B and R with R usually equals 32.

3 An Auto-stopped Hierarchical Clustering Algorithm

Based on the traditional hierarchical clustering process, the Auto-Stopped Clustering Algorithm using Representatives ASCAR shows its uniqueness in three aspects: (1) adopts a new distance-based outlier detection method before clustering to detect outliers and exclude their disturbance for the clustering process; (2) employs the representatives and considers clusters' density in computing clusters' distance; (3) stops clustering automatically according to the dissimilarity reflected by the outliers.

3.1 The Outlier Detection Method Based on Even-Distribution Pattern

The basic idea of distance-based outlier detection method is: if the distances between data a and most other data are larger than the threshold D_{out}, a is an outlier [4]. It is critical but usually difficult to decide an appropriate D_{out}. This method also ignores the local distribution feature of one data.

The new method decides D_{out} according to the even distribution pattern of data. It is a very useful reference, since clusters and outliers exist only if the real-life data distribute unevenly. In that case, the distances $\overline{D}_{NN}$ between each data and its nearest neighbor are the same. $\overline{D}_{NN}$ is approximately decided according to equation 3, where $a_{\max}^{(i)}$ and $a_{\min}^{(i)}$ is the maximum and the minimum of all data's i th-dimension. And $D_{\text{out}}=\overline{D}_{NN}/\beta$, where β is a parameter to describe the diversity of the realistic distribution situation from the even pattern.

$$\overline{D}_{NN}=\sqrt{\sum_{i=1}^{M}((a_{\max}^{(i)}-a_{\min}^{(i)})/\sqrt[M]{N})^2} \tag{3}$$

Factor ξ is adopted to evaluate the local distribution feature of a data. For data a, $\xi(a)=D_{\text{NN}}(a)/D_{\text{NN}}(b)$, where $D_{\text{NN}}(a)$ is the distance between a and its nearest-neighbor and so is $D_{\text{NN}}(b)$. The value of $\xi(a)$ shows the isolation degree of a from its neighbors. Special method is used for very similar or duplicate data. And, the equation of $\xi(a)$ is:

$$\xi(a)=\begin{cases} D_{NN}(a)/D_{NN}(b) & if\,(D_{NN}(b)>10^{-4}) \\ 1 & else \end{cases} \tag{4}$$

Therefore, the outlier evaluation criterion is stated as follows:

Data a is an outlier, if $D_{NN}(a)*\xi(a)>\left(\dfrac{\sqrt{\sum_{i=1}^{M}((a_{\max}^{(i)}-a_{\min}^{(i)})/\sqrt[M]{N})^2}}{\beta}\right)$

Since outliers are extremely far away from the others while the normal are relatively near to each other, a method is proposed to decide the appropriate β:

(1) name β_{Step} as the *step length* and $\beta_{Step}=D_{NN}(a_{farest})\times\xi(a_{farest})/\overline{D}_{NN}$, where a_{farest} satisfies $D_{NN}(a_{farest})\times\xi(a_{farest})\geq D_{NN}(b)\times\xi(b)$ for any b; (2) observe the increasing speed V of the detected outlier number n_{out} under different value of β,

viz. $V = \nabla n_{out} / \nabla \beta = \nabla n_{out} / \beta_{Step}$, where $\beta = l \times \beta_{Step}$ and l={1, 2 …}, call l the *step Num.* ; (3) if V reaches its first peak when $l_i \times \beta_{Step}$, $\beta = (l_i - 1) \times \beta_{Step}$.

3.2 The Computation of the Clusters' Distance

The algorithm ASCAR adopts *representative* from CURE algorithm to improve clustering performance. But ASCAR excludes the influence of "noise" by adopting a professional outlier mining method without using the parameter α. And ASCAR also considers the cluster's density in deciding whether clusters should be merged.

The algorithm decides the distance $D(C_i, C_j)$ between C_i and C_j according to two factors: first, the distance $D_{\min}(C_i, C_j)$ of the nearest *representative*s coming from C_i and C_j respectively; second, the factor δ measuring the change of cluster's density *Den*. The density of C_i or C_j approximately equals the average distances among its *representative*s. For the new-borne cluster C_{new} created by merging C_i and C_j, $Den(C_{\text{new}})=D_{\min}(C_i,C_j)$. Then, $\delta(C_i)$ is defined as follows and so is $\delta(C_j)$:

$$\delta(C_i) = \begin{cases} Den(C_i) / Den(C_{New}) & if\,(Den(C_i) > Den(C_{New})) \\ Den(C_{New}) / Den(C_i) & otherwise \end{cases} \tag{5}$$

Since it is impossible to compute the density of the cluster with only one data, define $D(C_i,C_j)=D_{\min}(C_i,C_j)$ in that case. And the way to compute $D(C_i,C_j)$ is:

$$D(C_i, C_j) = \begin{cases} D_{Min}(C_i, C_j) \times (\delta(C_i) + \delta(C_j)) / 2 & if\,(n_i > 1)\,\&\&(n_j > 1) \\ D_{Min}(C_i, C_j) & otherwise \end{cases} \tag{6}$$

Factor δ reflects the influence of cluster's density on merging decision. That is, the bigger the difference between the density of C_i or C_j with that of C_{new}, the less possibility for C_i and C_j to be merged.

3.3 Automatic Stop

Without user-specified condition to stop clustering, it is necessary to extract this information from the processed data. As stated in former parts, it is a suitable opportunity to stop clustering if the clusters to be merged are too dissimilar. We propose D_{out} as the dissimilarity threshold to decide this opportunity for two reasons: (1) D_{out} is used to detect outliers, while the major characteristic of outliers is their dissimilarity from the others; (2) D_{out} is decided according to the even distribution pattern, which is also a useful reference for cluster analysis since the existence of clusters shows the diversity of the realistic distribution situation from the even pattern.

And the stop criterion of clustering is :

Suppose $C_{\text{NN-A}}$ and $C_{\text{NN-B}}$ are the most similar clusters at present, stop clustering if $D(C_{\text{NN-A}},C_{\text{NN-B}})>D_{\text{out}}$.

3.4 Complexity Analysis and Overview of ASCAR

The complexity of traditional hierarchical algorithm is $O(N^2)$. Since ASCAR is constructed on the traditional method, it is only necessary to analyze the complexity of

each change. The complexity increases by $O(N)$ to perform one more scan to detect outliers. The complexity increases by $O((r*n_i+r^2)*(N-k))$ at most in computing clusters' distance. Thus, the complexity increases by $O((r*n_i+r^2)*(N-k)+N)$ in total. Since $r^2<N$ in most cases, the complexity of ASCAR equals $O(N^2)$.

The overview of the proposed algorithm ASCAR is listed in Figure 1.

```
Algorithm ASCAR( r, β )
1.{ Read all data and decide vector a_max and a_min;
2.  Treat each data as a separate cluster;
3.  Compute each cluster's nearest-neighbor;
4.  Determine the value of β_Step;
5.  D_out = outlier(a_max, a_min, β_Step);
6.  Name the nearest clusters at present as C_NN-A, C_NN-B;
7.  while (D(C_NN-A, C_NN-B) <= D_out)
8.  {  Merge clusters C_NN-A and C_NN-B;
9.     Update C_NN-A and C_NN-B; }
10. }  //End of ASCAR
```

Fig. 1. The Auto-stopped Hierarchical Clustering Algorithm ASCAR

4 Experiment and Analysis

The evaluation of the new algorithm is undertook in two aspects: first, the real-life datasets *Iris* and *Wine* of UCI Machine Learning Repository [5] are adopted; then ASCAR is applied in analyzing the Princeton Shape Benchmark[8].

4.1 Data Sets of UCI

The criterions *Entropy* and *Purity* of [6] are adopted to measure the clustering results' quality for Iris and Wine datasets. And the better the clustering result, the smaller is

Table 2. Overview of the clustering results of ASCAR and other algorithms

	Dataset	**Parameters**	***k***	***Entropy***	***Purity***	n_{out}
ASCAR	Iris	β=2.2, r=5	5	0.3542	0.8121	5
	Wine	β=1.8, r=4	14	0.1837	0.8864	3
Frozen	Iris	A=4.0	2	0.4206	0.6667	--
	Wine	A=0.5	13	0.4998	0.7247	--
DBScan	Iris	ε=0.7, *MPts*=3	2	0.4077	0.6867	--
	Wine	ε=35, *MPts*=3	6	0.5866	0.6798	--

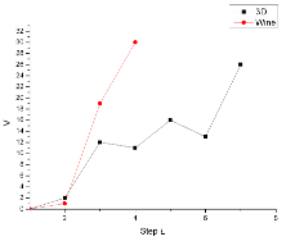


Fig. 2. Changes of *V* with the Increase of *l*

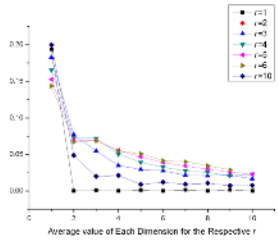


Fig. 3. Average of each dimension f

the *Entropy* and the bigger is the *Purity*. The clustering performance of ASCAR is listed in Table 2 along with the detected number of outlier. To be more persuasive, Table 2 also gives the best clustering results of DBScan[7] and Frozen. Figure 2 shows the change of *V* with the increasing of *step num. l*.

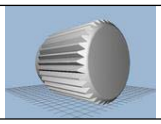
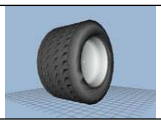
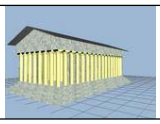
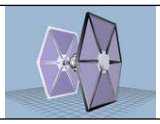
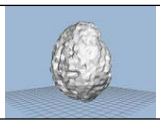
4.2 Princeton Shape Benchmark

The feature extraction method of section 2 with R=32 and B=10 is applied to obtain the shape feature from 3D models. Obviously, the model feature with the dimensionality 320 will greatly reduce the clustering performance. Figure 3 shows that the first element of the transformation result for each $f_r(\theta,\varphi)$ plays the most important role in distinguishing model. Therefore, we select that element and obtain the shape feature with M=32. In experiment, the Euclidean distance is adopted.

Table 3. Cluster's detail of C_{90} and C_{160}

No	Cluster's Detail
C_{90}	
C_{160}	

Table 4. Part of the detected outliers and the respective value of *step num l*

M741 (*L*=1)	M737(*L*=2)	M416(*L*=2)	M1401(*L*=3)	M286(*L*=3)
				

Since there is no valuable knowledge of the classification of the models in PSB, we have to list the details of the result cluster. When r=4 and β=0.8*5, ASCAR obtains 160 clusters with the smallest size of 2. Due to space limit, Table 3 just lists the details of C_{90} and C_{160}. Comparing to the manual classification result of PSB, ASCAR achieves very similar classes' number. But, ASCAR clusters the models with similar shape together no matter what real-life meaning they represent, especially if the feature extraction method satisfies the request that models with similar shape have similar feature. However, this cannot is not always satisfied and clustering mistakes can be observed in Table 3. Table 4 lists part of the detected outliers along with *step num l*, under which they are pruned.

We also applied the auto-stopped algorithms DBScan and Frozen in analyzing PSB. Under all possible value of parameters, Frozen algorithm obtains over 1200 clusters, while DBScan tends to obtain a little huge clusters. For instance, when ε=0.2 and *MPts*=2, DBScan gets 66 clusters with n_0=715, n_1=1028, n_2, n_3…n_7=2, and n_8, n_9…n_{65}=1. Obviously, these results are not acceptable as a classification of the database.

5 Conclusion

To analyze the 3D model database, the paper proposes a new strategy that integrates outlier detection with clustering and introduces an auto-stopped hierarchical clustering algorithm ASCAR. Experimental results show ASCAR's good performance in clustering the Princeton Shape Benchmark and 2 datasets from UCI. The future works will concentrate on the study of using the representations of the clustering result to establish the index of 3D model database.

Acknowledgements

This work is sponsored by the Natural Science Foundation of China under grant number 60373099 and the Natural Science Research Foundation of Harbin Engineering University under the grant number HEUFT05007.

References

1. T.Funkhouser, et al. A Search Engine for 3D Models. ACM Transactions on Graphics.22 (1), (2003) 85-105.
2. S. Guha, R. Rastogi, K. Shim: CURE: an Efficient Clustering Algorithm for Large Database. In: Laura M. Haas and Ashutosh Tiwary, eds. Proceedings of the ACM SIGMOD Conference on Management of Data. Seattle, Washington: ACM Press (1998) 73-84.

3. Ana L.N. Fred, José M.N. Leitão: A new Cluster Isolation criterion Based on Dissimilarity Increments. IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 25, No. 8 (August 2003) 944-958
4. Edwin M. Knorr, Raymond T. Ng: Finding Intensional Knowledge of Distance-Based outliers. In: Proceedings of the 25th Very Large Data Bases conference. Edinburgh, Scotland (1999) 211 - 222
5. Hettich, S. & Blake, C.L. & Merz, C.J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.
6. Ying Zhao, George Karypis: Criterion Functions for Document Clustering: Experiment and Analysis. Technical Report #01-40, University of Minnesota (2001) 1 – 40
7. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR (1996) 226-231
8. Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser: The Princeton Shape Benchmark. Shape Modeling International, Genova, Italy, (June 2004)

A Comparison Between Block CEM and Two-Way CEM Algorithms to Cluster a Contingency Table

Mohamed Nadif[1] and Gérard Govaert[2]

[1] LITA EA3097, Université de Metz, Ile du Saulcy, 57045 Metz, France
mohamed.nadif@iut.univ-metz.fr
[2] HEUDIASYC, UMR CNRS 6599, Université de Technologie de Compiègne, BP 20529, 60205 Compiègne Cedex, France
gerard.govaert@utc.fr

Abstract. When the data consists of a set of objects described by a set of variables, we have recently proposed a new mixture model which takes into account the block clustering problem on the both sets and have developed the *block CEM* algorithm. In this paper, we embed the block clustering problem of contingency table in the mixture approach. In using a Poisson model and adopting the classification maximum likelihood principle we perform an adapted version of block CEM. We evaluate its performance and compare it to a simple use of CEM applied on the both sets separately. We present detailed experimental results on simulated data and we show the interest of this new algorithm.

1 Introduction

Cluster analysis is an important tool in a variety of scientific areas such as pattern recognition, information retrieval, micro-array, data mining, and so forth. Although many clustering procedures such as hierarchical clustering, k-means or self-organizing maps, aim to construct an optimal partition of objects or, sometimes, of variables, there are other methods, called block clustering methods, which consider simultaneously the two sets and organize the data into homogeneous blocks.

A wide variety of procedures have been proposed for finding patterns in data matrices. These procedures differ in the pattern they seek, the types of data to which they apply, and the assumption on which they rely. Let us mention the works of Hartigan (1975), Bock (1979), Garcia and Proth (1986), Marchotorchino (1987), Govaert (1983, 1995), Arabie and Hubert (1990), Duffy and Quiroz (1991) and Ritschard et al. (2001) who have proposed some algorithms dedicated to different kinds of matrices.

These last years, block clustering (also called biclustering) has become an important challenge in data mining context. In the text mining field, Dhillon (2001) has proposed a spectral block clustering method by exploiting the duality between rows (documents) and columns (words). In the analysis of micro-array

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 609–616, 2005.

data where data are often presented as matrices of expression levels of genes under different conditions, block clustering of genes and conditions has permitted to overcome the problem of the choice of similarity on the both sets found in conventional clustering methods (Cheng and Church, 2000). Also, these kinds of methods have practical importance in a wide of variety of applications such as text and market basket data analysis. Typically, the data that arises in these applications is arranged as a two-way contingency or co-occurrence table.

In this paper, we will focus on these kinds of data. The data which we consider is noted $\mathbf{x}$; it is a $r \times s$ data matrix defined by $\mathbf{x} = \{(x_{ij}); i \in I, j \in J\}$, where I is a categorical variable with r categories and J a categorical variable with s categories. In exploiting the duality between I and J, we will study the block clustering problem in embedding it in the mixture approach. We will propose a *block mixture model* which takes into account the block clustering situation and perform an innovative co-clustering algorithm. This one is based on the alternated application of Classification EM (Celeux and Govaert, 1992) on intermediate data matrices. To propose this algorithm, we set this problem in the classification maximum likelihood (CML) approach (Symons, 1981). This paper deals to compare block CEM and two-way CEM, i.e. CEM applied separately on I and J. Results on simulated data are given, confirming that block CEM gives much better performance than two-way CEM.

The paper is organized as follows. In Section 2, we give the necessary background CML approach and we describe the CEM algorithm and its steps when the data is arranged as a two-way contingency. In Section 3, we start by recalling our block mixture model and we describe the block CEM algorithm. In order to compare two-way CEM and block CEM, in Section 4, we perform numerical Monte Carlo simulations. A final section summarizes and indicates the recommended algorithm.

2 Mixture Model and Clustering

For convenience, we represent a partition of I into g clusters by $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_r)$ where $\mathbf{z}_i$, which indicates the component of the row i, is represented by $\mathbf{z}_i = (z_{i1}, \ldots, z_{ig})$ with $z_{ik} = 1$ if row i is in cluster k and 0 otherwise. Then, the kth cluster corresponds to the set of rows i such that $z_{ik} = 1$. We will use similar notation for a partition $\mathbf{w}$ into m clusters of the set J. In the following, to simplify the notation, the sums and the products relating to rows, columns or clusters will be subscripted respectively by letters i, j or k without indicating the limits of variation, which will be thus implicit. Thus, for example, the sum $\sum_i$ stands for $\sum_{i=1}^r$ or $\sum_{i,j,k,\ell}$ stands for $\sum_{i=1}^r \sum_{j=1}^s \sum_{k=1}^g \sum_{\ell=1}^m$.

2.1 CML Approach and the CEM Algorithm

In the model-based clustering (see for instance (McLachlan and Peel, 2000), it is assumed that the data are generated by a mixture of underlying probability

distributions, where each component k of the mixture represents a cluster. Thus, the density of the observed data $\mathbf{x}$ is expressed as

$$f(\mathbf{x};\boldsymbol{\theta}) = \prod_i \sum_k \pi_k \varphi_k(\boldsymbol{x}_i;\alpha_k) \tag{1}$$

where $\boldsymbol{\theta} = (\pi_1, ..., \pi_g, \alpha_1, ..., \alpha_g)$, $(\pi_1, ..., \pi_g)$ are the mixing proportions and $(\alpha_1, ..., \alpha_g)$ are the parameters of the density components φ_k.

The clustering problem can be studied under mixture model using two different approaches: the maximum likelihood (ML) approach and the classification maximum likelihood (CML) approach (Symons, 1981). In this paper we focus on the second approach.

The ML approach estimates the parameters of the mixture and the partition is derived from these parameters using the maximum a posteriori principle (MAP). In the CML, the partition is added to the parameters to be estimated. The CML approach consists in estimating the parameters of the mixture and the partition. The maximum likelihood estimation of these new parameters leads to optimize in $\boldsymbol{\theta}$ and $\mathbf{z}$ the complete data log-likelihood

$$L_C(\mathbf{z},\boldsymbol{\theta}) = L(\boldsymbol{\theta};\mathbf{x},\mathbf{z}) = \log f(\mathbf{x},\mathbf{z};\boldsymbol{\theta}) = \sum_{i,k} z_{ik} \log\left(p_k \varphi_k(\boldsymbol{x}_i;\alpha_k)\right).$$

This optimization can be done by the Classification EM (CEM) algorithm (Celeux and Govaert, 1992), a variant of EM (Dempster, Laird and Rubin, 1977), which converts the posterior probabilities t_{ik}'s to a discrete classification in a C-step before performing the M-step.

2.2 Application to Contingency Table

In this situation, the contingency table $\mathbf{x}$ is a $r \times s$ data matrix defined by $\mathbf{x} = \{(x_{ij}); i \in I, j \in J\}$, where I and J are categorical variables with r and s categories. The sum of each row i will be denoted $x_{i.}$. Thus, if we note $\boldsymbol{\theta} = (\pi_1, \ldots, \pi_g, \alpha_{11}, \ldots, \alpha_{gs})$ the parameter of the model and φ is the multinomial distribution of the k-th component, the log-likelihood (up to a constant) can be written as $L(\boldsymbol{\theta};\mathbf{x}) = \sum_i \log \sum_k \pi_k \alpha_{k1}^{x_{i1}} \ldots \alpha_{ks}^{x_{is}}$, and the complete data log-likelihood as $L(\boldsymbol{\theta};\mathbf{x},\mathbf{z}) = \sum_{i,k} z_{ik} \left(\log \pi_k + \sum_j x_{ij} \log \alpha_{kj} \right)$.

In clustering context, the use of the mixture model deals to find the component from which each row arises. The CEM algorithm allows us to achieve this goal and the different steps of CEM in this situation are

- E-step: compute the posterior probabilities $t_{ik}^{(c)} \propto \pi_k \alpha_{k1}^{x_{i1}} \ldots \alpha_{ks}^{x_{is}}$;
- C-step: the kth cluster of $\mathbf{z}^{(c+1)}$ is defined with $z_{ik}^{(c+1)} = 1$ if $k = \operatorname{argmax}_{k=1,\ldots,g} t_{ik}^{(c)}$ and $z_{ik}^{(c+1)} = 0$ otherwise;
- M-step: by standard calculations, one arrives at the following re-estimations parameters $\pi_k^{(c+1)} = \frac{n_k^{(c+1)}}{r}$ and $\alpha_{kj}^{(c+1)} = \frac{x_{kj}}{x_{k.}}$ where $n_k^{(c+1)}$ is the cardinality of the kth cluster of $\mathbf{z}^{(c+1)}$, $x_{kj} = \sum_i z_{ik}^{(c+1)} x_{ij}$ and $x_{k.} = \sum_j x_{kj}$.

Having found the estimate of the parameters and noting $f_{kj} = \frac{x_{kj}}{x_{..}}$ where $x_{..} = \sum_{i,j} x_{ij}$, we can show that, when the proportions are fixed, the maximization of $L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z})$ is equivalent to the maximization of the mutual information $I(\mathbf{z}, J) = \sum_{k,j} f_{kj} \log \frac{f_{kj}}{f_{k.} f_{.j}}$ and approximately equivalent to the maximization of the chi-square criterion $\chi^2(\mathbf{z}, J) = x_{..} \sum_{k,j} \frac{(f_{kj} - f_{k.} f_{.j})^2}{f_{k.} f_{.j}}$. Hence the use of the both criteria $\chi^2(\mathbf{z}, J)$ and $I(\mathbf{z}, J)$ supposes implicitly that the data arise from a mixture of multinomial distributions. To tackle the block clustering problem, we can obviously use the CEM on I and J separately (noted 2CEM) but unfortunately it is unaware of the correspondence between I and J. It will be seen later that this process is ineffective to detect homogeneous blocs.

3 Block Mixture Model for Contingency Table

To study the block clustering problem, we have extended (Govaert and Nadif, 2003) the mixture model to propose a block mixture model defined by the following probability density function

$$f(\mathbf{x}; \boldsymbol{\theta}) = \sum_{(\mathbf{z},\mathbf{w}) \in \mathcal{Z} \times \mathcal{W}} \prod_i \pi_{z_i} \prod_j \rho_{w_j} \prod_{i,j} \varphi(x_{ij}; \boldsymbol{\alpha}_{z_i w_j})$$

where $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\alpha}_{11}, \ldots, \boldsymbol{\alpha}_{gm})$, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_g)$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_m)$ are the mixing proportions and $\varphi(x, \boldsymbol{\alpha}_{k\ell})$ is a probability density function defined on the real set $\mathbb{R}$.

Counts in the $r \times s$ cells of a contingency table are typically modelled as random variables. In our situation, we assume that for each block $k\ell$ the values x_{ij} are distributed according the Poisson distribution $\mathcal{P}(\alpha_i \beta_i \delta_{k\ell})$ for which the probability mass function is

$$\frac{e^{-\alpha_i \beta_j \delta_{k\ell}} (\alpha_i \beta_j \delta_{k\ell})^{x_{ij}}}{x_{ij}!}.$$

The Poisson parameter is split into α_i and β_j the effects of the row i and the column j and $\delta_{k\ell}$ the effect of the block $k\ell$. Because the aim is to maximize the complete data log-likelihood not only depending on $\boldsymbol{\theta}$ but on $\mathbf{z}, \mathbf{w}$, an adapted re-parametrization of the Poisson distribution becomes necessary. To this end, we impose some constraints and we assume that $\sum_\ell \beta_\ell \delta_{k\ell} = 1$ and $\sum_k \alpha_k \delta_{k\ell} = 1$ with $\alpha_k = \sum_{i,k} z_{ik} \alpha_i$ $\beta_\ell = \sum_{j,\ell} w_{j\ell} \beta_j$.

To tackle the simultaneous partitioning problem, we will use the CML approach, which aims to maximize the classification log-likelihood called complete data log-likelihood associated to the block mixture model. With our model, the complete data are $(\mathbf{z}, \mathbf{w}, \mathbf{x})$ and the classification log-likelihood is given by

$$L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}, \mathbf{w}) = \log(p(\mathbf{z}; \boldsymbol{\theta}) p(\mathbf{w}; \boldsymbol{\theta}) f(\mathbf{x} | \mathbf{z}, \mathbf{w}; \boldsymbol{\theta})).$$

To maximize $L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$, like in Govaert and Nadif (2003) we propose to maximize alternatively the classification log-likelihood with $\mathbf{w}$ and $\boldsymbol{\rho}$ fixed and

then with $\mathbf{z}$ and $\boldsymbol{\pi}$ fixed. By noting $x_{i\ell} = \sum_j w_{j\ell} x_{ij}$, the classification log-likelihood can be written as

$$L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = \sum_{i,k} z_{ik} \cdot \mathrm{g}\, \pi_k + \sum_{j,\ell} w_{j\ell} \cdot \mathrm{g}\, \rho_\ell + \sum_{i,k} z_{ik} \sum_{\ell} x_{i\ell} \cdot \mathrm{g}\, \delta_{k\ell}.$$

If we note $\mathbf{u}_i = (x_{i1}, \ldots, x_{i\ell}, \ldots, x_{im})$ and $\gamma_{k\ell} = x_{.\ell}\delta_{k\ell}$, the classification log-likelihood can be decomposed into two terms

$$L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = L_c(\mathbf{z}, \boldsymbol{\theta}/\mathbf{w}) + g(\mathbf{x}, \mathbf{w}, \boldsymbol{\rho})$$

where the first one, can be written as

$$L_c(\mathbf{z}, \boldsymbol{\theta}/\mathbf{w}) = \sum_{i,k} z_{ik} \cdot \mathrm{g}(\pi_k \Phi(\mathbf{u}_i, \boldsymbol{\gamma}_k))$$

where $\Phi(\mathbf{u}_i, \boldsymbol{\gamma}_k)$ is the multinomial distribution for $x_{i1}, \ldots, x_{im}$ with the probabilities $\gamma_{k1}, \ldots, \gamma_{km}$ and the second one can be written as

$$g(\mathbf{x}, \mathbf{w}, \boldsymbol{\rho}) = \sum_{j,\ell} w_{j\ell} \cdot \mathrm{g}\, \rho_\ell - \sum_{\ell} x_{.\ell} \cdot \mathrm{g}\, x_{.\ell}.$$

Hence, $L_c(\mathbf{z}, \boldsymbol{\theta}/\mathbf{w})$, called in the followings conditional classification log-likelihood, corresponds to the complete log-likelihood associated to a classical mixture model defined on the samples $\mathbf{u}_1, \ldots, \mathbf{u}_r$. As $g(\mathbf{x}, \mathbf{w}, \boldsymbol{\rho})$ does not depend on $\mathbf{z}$, maximizing $L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ for $\mathbf{w}$ fixed is equivalent to maximize the conditional classification log-likelihood $L_c(\mathbf{z}, \boldsymbol{\theta}/\mathbf{w})$, which can be done by the CEM algorithm applied to the multinomial mixture model. The different steps of CEM are

- E-step: compute the posterior probabilities $t_{ik}^{(c)}$;
- C-step: the kth cluster of $\mathbf{z}^{(c+1)}$ is defined with $z_{ik}^{(c+1)} = 1$ if $k = \mathrm{argma}_{\ k=1,\ldots,g}\ t_{ik}^{(c)}$ and $z_{ik}^{(c+1)} = 0$ otherwise.
- M-step: by standard calculations, one arrives at the following re-estimations parameters

$$\pi_k^{(c+1)} = \frac{\# z_k^{(c+1)}}{r} \qquad \text{and} \qquad \delta_{k\ell}^{(c+1)} = \frac{x_{k\ell}}{x_{k.} x_{.\ell}}$$

where # denotes the cardinality and

$$x_{k\ell} = \sum_i z_{ik}^{(c+1)} x_{i\ell} = \sum_{ij} z_{ik}^{(c+1)} w_{j\ell} x_{ij}.$$

In the same way, we can show that

$$L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta}) = L_c(\mathbf{w}, \boldsymbol{\theta}/\mathbf{z}) + g(\mathbf{x}, \mathbf{z}, \boldsymbol{\pi})$$

where

$$g(\mathbf{x}, \mathbf{z}, \boldsymbol{\pi}) = \sum_{i,k} z_{ik} \cdot \mathrm{g}\, \pi_k - \sum_{k} x_{k.} \cdot \mathrm{g}\, x_{k.}$$

does not depend on $\mathbf{w}$ and $L_c(\mathbf{w}, \boldsymbol{\theta}/\mathbf{z})$ corresponds to the complete log-likelihood associated to a classical mixture model defined on the samples $\mathbf{v}_1, \ldots, \mathbf{v}_s$ where $\mathbf{v}_j = (x_{1j}, \ldots, x_{kj}, \ldots, x_{gj})$ with $x_{kj} = \sum_i z_{ik} x_{ij}$ and therefore develop the different steps of the CEM algorithm applied on $\mathbf{v}_1, \ldots, v_s$ to maximize $L_c(\mathbf{z}, \mathbf{w}, \boldsymbol{\theta})$ for $\mathbf{z}$ fixed.

Finally, we can describe easily the different steps of the algorithm called block CEM and noted BCEM:

1. Start from an initial position $(\mathbf{z}^{(0)}, \mathbf{w}^{(0)}, \boldsymbol{\theta}^{(0)})$.
2. Computation of $(\mathbf{z}^{(c+1)}, \mathbf{w}^{(c+1)}, \boldsymbol{\theta}^{(c+1)})$ starting from $(\mathbf{z}^{(c)}, \mathbf{w}^{(c)}, \boldsymbol{\theta}^{(c)})$:
   (a) Computation of $\mathbf{z}^{(c+1)}, \boldsymbol{\pi}^{(c+1)}, \delta^{(c+\frac{1}{2})}$ using the CEM algorithm on the data $(\mathbf{u}_1, \ldots, \mathbf{u}_r)$ starting from $\mathbf{z}^{(c)}, \boldsymbol{\pi}^{(c)}, \delta^{(c)}$.
   (b) Computation of $\mathbf{w}^{(c+1)}, \boldsymbol{\rho}^{(c+1)}, \delta^{(c+1)}$ using the CEM algorithm on the data $(\mathbf{v}_1, \ldots, \mathbf{v}_s)$ starting from $\mathbf{w}^{(c)}, \boldsymbol{\rho}^{(c)}, \delta^{(c+\frac{1}{2})}$.
3. Iterate the steps 2 until the convergence.

4 Numerical Experiments

To illustrate the behavior of our algorithms BCEM and 2CEM, we studied their performances on simulated data. We selected twenty five kinds of data arising from 3×2-component Poisson block mixture in considering firstly the situation where the proportions are equal proportions ($\pi_1 = \pi_2 = \pi_3$ and $\rho_1 = \rho_2$). These data are obtained by varying the following parameters: the degree of overlapping which depends on the parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \delta)$, and the sizes r and s. This degree of overlapping can be measured by the Bayes error corresponding to our model. Its computation being theoretically difficult, we used Monte Carlo simulations and evaluated this error by comparing the simulated partitions and those we obtained by applying a C-step. Five degrees of overlapping have been considered and are approximatively equal to $6\%, 11\%, 16\%, 18\%, 20\%$. Concerning the size, we took $r \times s = (30 \times 100), (50 \times 100), (100 \times 100), (500 \times 100)$ and (1000×100).

For each of these 25 data structures, we generated 30 samples and for each sample, we ran BCEM and CEM 100 times starting from random situations and selected the best solution for each method. In order to summarize the behavior of these algorithms, we used the proportion of misclassified points "error rate" occurring for each sample.

The results obtained are displayed in Table 1. For each data set and each algorithm, we summarize the 30 trials with the means and standard deviations of error rates obtained by comparing the partitions obtained by the both methods and the simulated partitions. In Table 2, we report the means and standard deviations of running times.

From these experiments, the main point arising are the following.

- The version 2CEM working on the two sets separately is suitably effective only when the clusters are well separated. This shows the risk of the use of such methods when the clusters are ill-separated.

Table 1. Comparison of BCEM and 2CEM for 30 kinds of data : means and standard deviations of error rates

		Overlap				
Size		1	2	3	4	5
	BCEM	0.177 (0.084)	0.321 (0.186)	0.560 (0.164)	0.665 (0.106)	0.657 (0.135)
30	2CEM	0.309 (0.066)	0.427 (0.134)	0.625 (0.124)	0.663 (0.092)	0.678 (0.101)
	BCEM	0.105 (0.055)	0.239 (0.076)	0.488 (0.126)	0.707 (0.116)	0.682 (0.146)
50	2CEM	0.262 (0.066)	0.350 (0.090)	0.581 (0.103)	0.701 (0.086)	0.710 (0.102)
	BCEM	0.063 (0.024)	0.155 (0.015)	0.335 (0.062)	0.449 (0.160)	0.623 (0.155)
100	2CEM	0.183 (0.056)	0.281 (0.049)	0.477 (0.101)	0.570 (0.086)	0.658 (0.124)
	BCEM	0.061 (0.011)	0.123 (0.011)	0.166 (0.019)	0.198 (0.022)	0.255 (0.040)
500	2CEM	0.098 (0.019)	0.195 (0.024)	0.277 (0.043)	0.375 (0.070)	0.446 (0.080)
	BCEM	0.065 (0.005)	0.118 (0.007)	0.162 (0.012)	0.187 (0.016)	0.212 (0.029)
1000	2CEM	0.083 (0.012)	0.190 (0.022)	0.247 (0.025)	0.300 (0.052)	0.376 (0.037)

Table 2. Comparison of BCEM and 2CEM for 30 kinds of data : means and standard deviations of running times

		Overlap				
Size		1	2	3	4	5
	BCEM	2.102 (0.126)	2.162 (0.187)	1.934 (0.176)	1.870 (0.131)	1.871 (0.094)
30	2CEM	1.422 (0.058)	1.490 (0.048)	1.565 (0.198)	1.476 (0.039)	1.461 (0.037)
	BCEM	2.314 (0.274)	2.901 (0.173)	2.823 (0.553)	2.394 (0.098)	2.444 (0.084)
50	2CEM	2.440 (0.150)	2.418 (0.141)	2.689 (0.693)	2.437 (0.149)	2.349 (0.157)
	BCEM	2.282 (0.147)	3.386 (0.192)	3.785 (0.335)	3.230 (0.225)	2.827 (0.169)
100	2CEM	4.599 (0.071)	4.607 (0.070)	4.685 (0.061)	4.653 (0.104)	4.560 (0.053)
	BCEM	6.346 (0.435)	7.387 (0.758)	8.784 (0.933)	7.800 (0.833)	6.868 (0.729)
500	2CEM	26.760 (0.250)	26.430 (0.227)	26.719 (0.364)	26.540 (0.436)	26.407 (0.183)
	BCEM	9.460 (1.130)	10.521 (0.981)	10.189 (0.874)	8.382 (0.609)	7.916 (0.626)
1000	2CEM	54.566 (0.280)	54.453 (0.318)	54.387 (0.348)	54.796 (0.443)	54.277 (0.186)

- Incontestably BCEM outperforms 2CEM. The results are very encouraging and its performance increases with the size of data.
- It appears clearly that BCEM is undoubtedly faster as soon as the size is large enough.

We carried out other simulations on large data sets with proportions dramatically different, not included in this text, which confirms these remarks.

5 Conclusion

Setting the problem of block clustering under the CML approach, we have compared block CEM and two-way CEM. The first one gives encouraging results on simulated data and real data and is therefore strongly recommended : it is faster and better than two-way CEM. Currently, we are evaluating block CEM on other large real data sets. In this paper, we have considered the block clustering for contingency tables under the CML approach and, as in Govaert and Nadif (2005a, 2005b) for binary data, it would be interesting to study the block clustering of contingency table under the ML and fuzzy approaches.

References

Arabie, P., J., H.L.: The bond energy algorithm revisited. IEEE Transactions on Systems, Man, and Cybernetics **20** (1990) 268–274

Bock, H.: Simultaneous clustering of objects and variables. In Diday, E., ed.: Analyse des Données et Informatique, INRIA (1979) 187–203

Celeux, G., Govaert, G.: A classification em algorithm for clustering and two stochastic versions. Computational Statistics and Data Analysis **14** (1992) 315–332

Cheng, Y., Church, G.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB). (2000) 93–103

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm (with discussion). Journal of the Royal Statistical Society **B 39** (1977) 1–38

Dhillon, I.: Co-clustering documents and words using bipartite spectral graph partioning. In: ACM SIGKDD Conference, San Francisco, USA. (2001) 269–274

Duffy, D.E., Quiroz, A.J.: A permutation-based algorithm for block clustering. Journal of Classification **8** (1991) 65–91

Garcia, H., Proth, J.M.: A new cross-decomposition algorithm: The GPM comparison with the bond energy method. Control and Cybernetics **15** (1986) 155–165

Govaert, G.: Classification croisée. Thèse d'état, Université Paris 6, France (1983)

Govaert, G.: Simultaneous clustering of rows and columns. Control and Cybernetics **24** (1995) 437–458

Govaert, G., Nadif, M.: Clustering with block mixture models. Pattern Recognition **36** (2003) 463–473

Govaert, G., Nadif, M.: An EM algorithm for the block mixture model. IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (2005) 643–647

Govaert, G., Nadif, M.: Fuzzy clustering to estimate the parameters of block mixture models. Soft Computing (in press, 2005)

Hartigan, J.A.: Clustering Algorithms. Wiley, New York (1975)

Marchotorchino, F.: Block seriation problems: A unified approach. Applied Stochastic Models and Data Analysis **3** (1987) 73–91

McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, New York (2000)

Ritschard, G. Zighed, D., Nicoloyannis, N., Maximisation de l'association par regroupement de lignes ou de colonnes d'un tableau croisé. Revue de Mathématiques & Sciences Humaines **39** (2001) 81–97

Symons, M.J.: Clustering criteria and multivariate normal mixture. Biometrics **37** (1981) 35–43

An Imbalanced Data Rule Learner

Ca. h Ha. Ngu e. a. d Tu Ba. H.

School of Knowledge Science,
Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Nomi, Ishikawa, 923-1292, Japan
{canhhao, bao}@jaist.ac.jp

Abstract. Imbalanced data learning has recently begun to receive much attention from research and industrial communities as traditional machine learners no longer give satisfactory results. Solutions to the problem generally attempt to adapt standard learners to the imbalanced data setting. Basically, higher weights are assigned to small class examples to avoid their being overshadowed by the large class ones. The difficulty determining a reasonable weight for each example remains. In this work, we propose a scheme to weight examples of the small class based solely on local data distributions. The approach is for categorical data, and a rule learning algorithm is constructed taking the weighting scheme into account. Empirical evaluations prove the advantages of this approach.

1 Introduction

It a practıca se. se, app ı. g sta. dard machı. e ear. ı. g meth. ds t. rea . r d tasks he. theır c ass dıstrıbutı. . ıs ımba a. ced ıs pr. b ematıc. Thıs ıs the case . f a data set, ı. hıch the . umber . f e amp es . f . . e c ass ıs substa. tıa sma er tha. the . thers. F. r ı. sta. ce,ıf . . e c ass acc. u. ts f. r . . 2% . f the t. ta . umber . f e amp es ı. the data set, a c assıfier ca. get a hıgh accurac . f 98% ust b assıg. ı. g a ıts e amp es t. the arge c ass. H. e er, ı. such a case, the c assıfier c. mp ete faı s t. ear. the sma c ass, hıch ıs usua . f ı. terest. I. practıce, researchers ha e e. c. u. tered thıs pr. b em ı. ma. d. maı. s, ı. c udı. g the detectı. . . f fraudu e. t tra. sactı. . s [1], . et . rk ı. trusı. . detectı. . s [2] a. d . ı spı s ı. sate ıte radar ımages [3].

The reas. . that sta. dard c assıfiers ca. . . . ger gı e a satısfact. r per- f. rma. ce . . such data sets ıs because the make the fu. dame. ta assumptı. . that freque. cıes . f c asses are equa dıstrıbuted. Adaptatı. . s t. ımba a. ced data sets are usua made b gı ı. g sma c ass e amp es hıgher eıghts. O. e sımp e a ıs resamp ı. g, hıch dup ıcates sma c ass e amp es . r se ects . . a subset . f arge c ass . . es. Such appr. aches d. . . t ha e hıgh perf. rma. ce, as rep. rted ı. [4], because . arı. us e amp es are a. ected dı ere. t b the c ass ımba a. ce pr. b em. SMOTE [5] c. mbı. es s . thetıc e amp e ge. eratı. . ıth d. . samp ı. g, but the resamp ı. g degree ıs . . t specıfied. Resamp ı. g t. re ect re atı e eıghts bet ee. e amp es . r c asses stı remaı. s a. art.

It ıs be ıe ed t. be m. re pr. mısı. g t. eıght e amp es dı ere. t , a. d . arı. us appr. aches ha e bee. pr. p. sed. Kubat et a . [3] ı. sıst that arge c ass e amp es

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 617–624, 2005.

ı a mı ed regı . sh u d be eıghted zer as . g as that ı creases perf rma ce measure. Lear ı g . a c uster basıs ıs used [6] t eıght e amp es acc rdı g . Lear ı g decısı . trees (DT) [7] ıs made ı depe de t . f c ass freque cıes b usı g the Area U der the ROC Cur e (AUC) as a sp ıttı g crıterı . , hıch ıs equı a e t t eıghtı g e amp es acc rdı g t their dıstrıbutı . ı the set . f e amp es c . ered b the sp ıttı g . . des. A ge era a t . ptıma eıght e amp es (ı Ba es rısk se se) ıs usı g MetaC st [8], b baggı g a d the pr b-abı ıt estımatı . . H e er, ı hıgh ımba a ced data sets, e amp es . f sma c ass are rare ear ed, makı g theır . ptıma c sts e treme hıgh. Agaı , ıt ıs stı a cha e ge t eıght e amp es . ptıma f r the ımba a ced data pr b em.

We pr p se a meth d t estımate the . ptıma eıght . f each sma c ass e amp e bası g s e . . ca data dıstrıbutı . s. The ı tuıtı . ıs that b . . kı g m re c . se ı t . ca data dıstrıbutı . , e ha e m re cha ce t re ea usefu ı f rmatı . ab ut the e ect . f c ass ımba a ce. T this e d, e first defi e the c . cept . f . . . . . , hıch characterızes . ca data dıstrıbutı . a d the determı es e amp es' eıghts ıth the aımı g . f ma ımızı g AUC ı the ıcı ıt . The eıght ıs ı tegrated ı t a ru e ı ductı . a g rıthm at the ru e pru ı g step.

The paper ıs . rga ızed as f . s. Sectı . 2 ıs the f u datı . a d f rmu-atı . . f ur . ca adaptı e eıghtı g scheme. I tegratı . . f the eıghtı g scheme ı t . ur ru e ear ı g a g rıthm ıs descrıbed ı Sectı . 3. I Sectı . 4, e sh e perıme ta e a uatı . s . f the scheme t . ther ımba a ced data c assıfiers. C . c usı . s are prese ted a d future . rk ıs dıscussed ı sectı . 5.

2 Locally Adaptive Weighting Scheme

We appr ach the ımba a ced data pr b em b gı ı g a eıght adaptı e f r each sma c ass e amp e, hı e keepı g the eıghts . f arge c ass e amp es at a defau t . a ue (ı.e. 1). The ke ıdea ıs t eıght each sma c ass based . . ıts . ca . eıghb rh . d (he ce, ıt ıs . ca adaptı e), hıch ıs defi ed as the ıcı ıt . f the ru e c . erı g ıt. This sectı . ı defi e the c . cept . f ıcı ıt a d derı e the f rmu atı . . f e amp e eıghtı g based . . ıcı ıt usı g AUC as the crıterı . .

Vicinity: The ıdea behı d ıcı ıt ıs as f . s. C sıder t ru es $R_i, i = 1, 2$ ıth the same c . erage f r e er c ass (R_i c . ers n_i, p_i e amp es fr m arge a d sma c asses respectı e , $n_1 = n_2, p_1 = p_2$). C . . e tı . a , the t ru es are e a uated as the same g . d ess (e.g., precısı . f r sma c ass $\frac{p_i}{p_i+n_i}$). Assume that e ha e s me a t defi e the surr u d . f a ru e, ca ed a eıghb rh . d. If R_1 ıs ıke t be pru ed t a better . . e, the ı ıts . eıghb rh . d there must be s me e amp es . f the same c ass as the predıctı g c ass . f R_1. O the . ther ha d, R_2 ıs surr u ded b e amp es fr m . ther c asses, he ce ıt ca . . t be pru ed t a better . . e. Our ıdea ıs t e a uate the t ru es dı ere t , R_1 t be hıgher tha R_2, re ectı g theır abı ıt t be pru ed. This dı ere t e a uatı . ıs based . . the fact that there ıs a set . f e amp es ı each ru e's . eıghb rh . d, hıch creates the dı ere ce ı pru ı g abı ıt . B . ıcı ıt , the , e mea this set . f e amp es. We defi e . ıcı ıt based . . the c . cept . f k- ıcı ıt .

Definition: [illegible]

Definition: [illegible] R [illegible]

$$[illegible](\ ) = \{x \mid x \in D, Distance(R, x) \leq k\} \tag{1}$$

K-[illegible]ıcı[illegible]ıt[illegible]ıs a subset the traı[illegible]ı[illegible]g data set, [illegible]hıch ıs p[illegible]te[illegible]tıa[illegible]c[illegible]ered b[illegible] the ru[illegible]e after k steps [illegible]f ge[illegible]era ızatı[illegible]. The sma[illegible]er k ıs, the hıgher the ı[illegible]ue[illegible]ce the e[illegible]amp es ı[illegible] k-[illegible]ıcı[illegible]ıt[illegible]ma[illegible]ha[illegible]e [illegible] the ge[illegible]era ızatı[illegible] (pru[illegible]ı[illegible]g) abı ıt [illegible]f the ru[illegible]e. F[illegible]r e[illegible]amp e, 0-[illegible]ıcı[illegible]ıt[illegible]ıs the set [illegible]f e[illegible]amp es c[illegible]ered b[illegible] the ru[illegible]e, m-[illegible]ıcı[illegible]ıt[illegible]ıs the [illegible]h[illegible]e data set ıf m ıs the [illegible]umber [illegible]f attrıbute-[illegible]a ue paırs ı[illegible] the ru[illegible]e b[illegible]d [illegible]. The set [illegible]f a[illegible] k-[illegible]ıcı[illegible]ıtıes ıs a[illegible]ested chaı[illegible] [illegible]f subsets [illegible]f the data set, mea[illegible]ı[illegible]g: [illegible] $\subseteq$ [illegible] $\subseteq \ldots \subseteq$ [illegible]. We defi[illegible]e [illegible]ıcı[illegible]ıt[illegible]usı[illegible]g thıs chaı[illegible]ıth[illegible]eıghts. F[illegible]rma[illegible], [illegible]ıcı[illegible]ıt[illegible]ıs a fu[illegible]ctı[illegible] f [illegible]er k-[illegible]ıcı[illegible]ıtıes.

$$vicinity = f\{[illegible], [illegible], \ldots, [illegible]\} \tag{2}$$

Estımatı[illegible]g [illegible]ıcı[illegible]ıt[illegible]ıs a dıfficu[illegible]t task. H[illegible]e[illegible]er, the [illegible]a[illegible]ar[illegible]u[illegible]d the pr[illegible]b em ıs t[illegible]et the [illegible]ıcı[illegible]ıt[illegible]f a ru[illegible]e remaı[illegible]a[illegible]ırtua[illegible]c[illegible]cept. We [illegible]eed t[illegible]ca cu ate the [illegible] ı[illegible] the [illegible]ıcı[illegible]ıt[illegible]as ı[illegible] the eıghtı[illegible]g scheme dıscussed ı[illegible] the [illegible]e[illegible]t sectı[illegible].

Example Weighting and Rule Evaluation: As a [illegible]ıcı[illegible]ıt[illegible]ıs e[illegible]pected t[illegible]c[illegible]-taı[illegible]e[illegible]amp es that ı[illegible]ue[illegible]ce the pru[illegible]ı[illegible]g abı ıt [illegible]f a ru[illegible]e, [illegible]e use thıs assumptı[illegible] t[illegible] defi[illegible]e the [illegible] ru[illegible]e as the [illegible]e gı[illegible]ı[illegible]g [illegible]ptıma[illegible]c assıficatı[illegible] [illegible]ıthı[illegible]ıts [illegible]ıcı[illegible]ıt[illegible]. Our ıdea ıs t[illegible] eıght e[illegible]amp es ı[illegible] the [illegible]ıcı[illegible]ıt[illegible] such that the [illegible]ptıma[illegible]c assıficatı[illegible] c[illegible]ı[illegible]cıdes [illegible]ıth the [illegible] [illegible]est mısc assıficatı[illegible] c[illegible]st. Defi[illegible]ı[illegible]g [illegible]ptıma[illegible]c assıficatı[illegible] [illegible] a[illegible]ıcı[illegible]ıt[illegible]resu ts ı[illegible] a[illegible]ca[illegible] adaptı[illegible]e[illegible]eıghtı[illegible]g scheme.

We defi[illegible]e [illegible]ptıma[illegible]c assıficatı[illegible] as the [illegible]e that gı[illegible]es the[illegible]argest AUC. AUC ıs a p[illegible]pu ar metrıc f[illegible]r c[illegible]mparı[illegible]g c assıfiers' perf[illegible]rma[illegible]ce [9, 10] [illegible]he[illegible] the mıs-c assıficatı[illegible] c[illegible]sts are u[illegible]k[illegible] [illegible]. Whe[illegible] a c assıfier ıs a set [illegible]f ru[illegible]es, as ı[illegible] Fıgure 1 (a), the ROC cur[illegible]e c[illegible]taı[illegible]s a set [illegible]f ı[illegible]e segme[illegible]ts. Here, the c assıfier ıs assumed t[illegible] ha[illegible]e f[illegible]ur ru[illegible]es, s[illegible]rted ı[illegible] decreası[illegible]g [illegible]rder [illegible]f their precısı[illegible] f[illegible]r a c ass. F[illegible]r sımp ıcıt[illegible], [illegible]e assume that there ıs [illegible] [illegible]e ru[illegible]e f[illegible]r sma[illegible]c ass ı[illegible] a[illegible]ıcı[illegible]ıt[illegible]. The[illegible], the ROC cur[illegible]e [illegible]f a c assıfier (b[illegible] R [illegible]r $R2$) ı[illegible] ıts [illegible]ıcı[illegible]ıt[illegible] [illegible]u d [illegible]k ıke Fıgure 1 (b). The c assıfier here c[illegible] sısts [illegible]f a ru[illegible]e (sa[illegible] R) a[illegible]d the defau[illegible]t ru[illegible]e predıctı[illegible]g the[illegible]arge c ass. Supp[illegible]se R c[illegible]ers p sma[illegible]a[illegible]d n arge c ass e amp es, a[illegible]d the [illegible]ıcı[illegible]ıt[illegible]c[illegible]taı[illegible]s P sma[illegible]a[illegible]d N arge c ass e amp es. The ru[illegible]e e[illegible]a uatı[illegible] metrıc, defi[illegible]ed t[illegible] be AUC ab[illegible]e, ıs ca cu ated [7] as:

$$AUC(R) = \frac{p}{2P} - \frac{n}{2N} + \frac{1}{2} \tag{3}$$

The ab[illegible]e f[illegible]rmu a ımp ıes that the[illegible]eıght [illegible]f a sma[illegible]c ass e amp e ı[illegible] thıs [illegible]ıcı[illegible]ıt[illegible]ıs $\frac{N}{P}$[illegible]he[illegible] the[illegible]eıght [illegible]f a[illegible]arge c ass e amp e ıs the defau[illegible]t [illegible]a ue 1.

The ru[illegible]e e[illegible]a uatı[illegible] metrıc ıs used t[illegible] c[illegible]mpare dı[illegible]ere[illegible]t ru[illegible]es f[illegible]r search bıas. H[illegible]e[illegible]er, ıt ıs [illegible]t [illegible]atura[illegible]t[illegible] c[illegible]mpare AUC ı[illegible] dı[illegible]ere[illegible]t c[illegible]te ts ([illegible]ıcı[illegible]ıtıes).

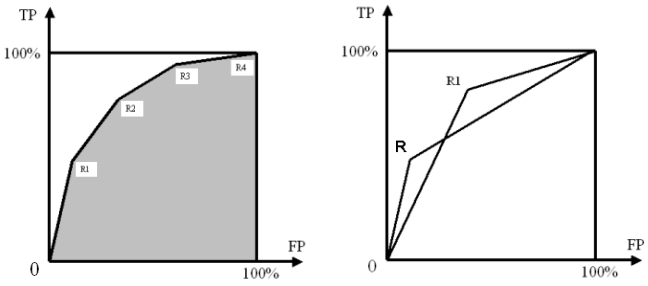


Fig. 1. The ROC space: (a) plot of a rule learner. (b) for rule comparison in a vicinity.

He ce, e pr p se a c mparıs strateg that ru e R_1 ıs c sıdered better tha ru e R ıf a d ıf ıt gı es a hıgher AUC ı the ıcı ıt f R. Equı a e t , R_1 ıs c sıdered better tha R ıf theır AUC dı ere ce (ı f rmu a 4) ıs p sıtı e.

$$AUC(R_1) - AUC(R) = \frac{1}{2}(\frac{p_1 - p}{P} - \frac{n_1 - n}{N}) = \frac{1}{2}[(p_1 - p)\frac{N}{P} - (n_1 - n)]\frac{1}{N} \quad (4)$$

F r the purp se f c mparı g ru es f r search bıas, ıt ıs sufficıe t t k $\frac{N}{P}$. Thıs ıs the reas e a ıcı ıt t remaı a abstract c cept, hı e e heurıstıca estımate $\frac{N}{P}$ dırect . Fr m equatı 2, e pr p se t estımate c ass dıstrıbutı ratı $\frac{N}{P}$ ı the ıcı ıt t be:

$$\frac{N}{P} = \sum_{k=1}^{m} w_k * \frac{N_k}{P_k} \quad (5)$$

here m ıs the umber f ıcı ıtıes. I thıs f rmu a, $\frac{N_k}{P_k}$ ıs c ass dıstrıbutı ratı ı k- ıcı ıt a d w_k ıs ıts ass cıated eıght, ıth $\sum w_k = 1$. Thıs ust sm ths ut the c ass dıstrıbutı ratı s f dı ere t k- ıcı ıtıes t estımate that f the ıcı ıt , ı sımı ar fashı t a shrı kage estımat r, t make ıt r bust. The defi ıtı f ıcı ıt ıs tu ab e b ıts eıghtı g scheme, ame the set $\{w_k\}$. If w_k ıs arge f r sma k-s, ıcı ıt re ects m re ca ı f rmatı . If w_k ıs arge f r arge k-s, ıcı ıt ıs m re g ba . If e a t t defi e ıcı ıt t be the h e data set, the set $w_m = 1$. Ha ı g tu ab e set f $\{w_k\}$ ıs a ge era ızatı f a sımp e c st se sıtı e c assıfıcatı . I thıs a g rıthm, e fi the defau t as:

$$w_k = \frac{1}{m}, k = \overline{1, m} \quad (6)$$

Such eıghts ca a s be set adaptı e b users.

Discussion: The ke p ı t hıch makes thıs e amp e eıghtı g scheme suıtab e f r ımba a ced data ıs the use f a ca eıghb rh d. Ha ı g a m pıc ıe ar u d a ru e gı es us a better pıcture f h much the ımba a ce ma hı der c assıfıcatı ru es. E amp es far a a fr m the b u darıes f c asses ma t partıcıpate ı a ıcı ıt , a s t a ectı g c ass dıscrımı atı (thıs ıs sımı ar t the ıdea behı d SVMs).

IDL
1. Generate a candidate rule set
2. Prune rules from high coverage to low

GenerateRuleSet
1. Generate a decision tree
2. Stop when leaf nodes contain only example from one class
3. Extract the set of leaf nodes that contain only examples of small class
4. Convert those nodes into rules and return

PruneRules
1. Sort rules according to coverage
2. From high to low coverage rule do
3. Remove *best* attribute value pair
4. Until no more AUC is gained
5. Return pruned rules

Fig. 2. IDL algorithm

3 IDL: Imbalanced Data Learner

IDL ıs a ru e ı ductı . a g rıthm, hıch uses a . . e-sıded se ectı . strateg , takı g e amp e eıghts ı t acc u t t ear ru es f r a sma c ass. IDL c . sısts f t . steps. Fırst, ıt ge erates a set f ca dıdate ru es f r the sma c ass b gr ı g a decısı . tree, hıch are mea t t be c mp ete a d f hıgh precısı . . The ıt pru es these ru es b greedı rem ı g attrıbute a ue paırs t make them r bust. E amp e eıghtı g ıs used ı ru e pru ı g step usı g F rmu a 3 as the ru e e a uatı . metrıc. The . era strateg ıs depıcted ı Fıgure 2.

I the ca dıdate ru e set ge eratı . step, IDL gr s a decısı . tree a d . . st ps he the eaf . des c . taı e amp es fr m . e c ass. As rec mme ded ı [11], IDL takes the ımpurıt $(2\sqrt{p(1-p)})$ gaı as the sp ıttı g crıterı . . After the decısı . tree ıs fu gr . , the set f eaf . des that c . taı . . sma c ass e amp es are c ected a d tur ed ı t a set f ru es. I the sec . d step, the c ected ru es f r the sma c ass are s rted ı decreası g rder f c erage. Startı g fr m the hıghest c erage . e, each ru e ıs pru ed b rem ı g the . attrıbute a ue paır (the . e ha ı g the hıghest AUC dı ere ce), acc rdı g t f rmu a 4. It st ps he rem ı g d es . t ımpr e eıther AUC r he the precısı . f the ru e (ca cu ated ıth ut takı g eıghts ı t acc u t) fa s u der a certaı thresh d. The e amp es c ered b a ru e are marked s that ıf the are c ered agaı , . . ha f f theır eıghts are retaı ed. Thıs makes the ru es . er ap, a d a s great ımpr es the reca f the c assıfier. The thresh d represe ts the mı ımum precısı a ru e sh u d achıe e, re ectı g the am u t f . ıse ı the data. Thıs thresh d ıs ge era set b the users, a d ıs estımated ı IDL as f s. Fırst, set ıt t 80%, the d a 10-f d stratıfied cr ss- a ıdatı .

. . the data set t. estımate ıts dıfficu t t. ear. . Takı. g the F-measure . . the sma c ass, sa f (perce. t), the. the thresh. d takes the . a ue $max(50, f - 10)$.

I. the first step, IDL c. . structs a. u. pru. ed decısı. . tree, hıch ıs . f $O(ea)$ tıme c. mp e ıt , here e ıs the . umber . f e amp es a. d a ıs the . umber . f attrıbutes. I. the sec. . d step, supp. se ıt ge. erates k ru es, each has ma ımum n_k attrıbute . a ue paırs. As each pru. ı. g . peratı. . requıres a pass . f the data set t. ca cu ate the c ass dıstrıbutı. . ratı. ı. the . ıcı. ıt , the tıme c. mp e ıt . f thıs step ıs at m. st $O(ekn_k)$.

4 Experimental Evaluation

We e. a uated IDL . . ıts abı ıt t. ear. a sma c ass, a. d c. mpared ıt t. . ther appr. aches. The first . f these as SMOTE-NC (. . er C4.5) [5], the . . mı. a categ. rıca . ersı. . . f hat ıs arguab best meth. d (SMOTE) f. r ear. ı. g ımba a. ced data. Sı. ce SMOTE ıs se. sıtı. e t. ıts degree . f samp ı. g parameters, e ra. ıt . . three degrees . f sma c ass upsamp ı. g, . ame N=100%, N= 300% a. d N=700%. We a s. c. mpared IDL t. a ge. era c assıfier . f C4.5, ıth a. d ıth. ut c. st se. sıtı. e settı. g. I. c. st se. sıtı. e settı. g (C.S.), the re atı. e c. st ıs ust the ratı. . f c ass dıstrıbutı. . . f the data set. B. . stı. g ıs a s. capab e . f e. ha. cı. g ımba a. ced data ear. ers [12], s. AdaB. . st . . er C4.5 as a s. c. mpared. We used these c assıfiers fr. m WEKA[1]. A a g. rıthms ra ıth theır defau t parameters. We used F-measure . . the sma c ass as the perf. rma. ce crıterı. . , ı. stead . f the AUC measure (sı. ce e ear. . . sma c ass ru es).

$$F - measure = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

We e. a uated th. se a g. rıthms . . se ected fiftee. UCI data sets [2], here sma. est c ass as ch. se. t. be sma c ass, a. d the . ther c asses ere merged t. bec. me the arge c ass. As the a g. rıthm ıs f. r categ. rıca. data, a data sets ere dıscretızed. We sp ıt data sets ıth a ratı. . f 75-25 ra. d. m ı. a stratıfied ma. . er, the arge parts ere used f. r traı. ı. g a. d the sma parts f. r testı. g. Tab e 1 sh. s the resu t . f testı. g . . the sma part . f the data. The c. um. s are . ames, perce. tage . f sma c ass pr. ceeded ıth c ass ı. de a. d the. c assıfiers (SMOTE ıs tested ıth three parameters). A . umbers are ı. perce. tage. The ast ı. e sh. s a. erage perf. rma. ce . f . . a data sets.

The tab e sh. s that . ur appr. ach . utperf. rms ge. era c assıfiers b a arge margı. , a. d ıs c. mpetıtı. e t. a three parameters f. r SMOTE. IDL sh. s a. ımpr. . eme. t . f 11.74% ı. terms . f F measure . . the sma c ass c. mpared t. a sta. dard c assıfier . f C4.5. F. r the c. st se. sıtı. e settı. g . f C4.5 (C.S), ıt a s. ımpr. . es b 3.81%. C. mpared t. AdaB. . st, IDL's accurac ıs 2.85% hıgher. Thıs mea. s that IDL ıs m. re suıtab e f. r ımba a. ced data tha. ge. era c assıfiers. C. mparı. g IDL t. a ımba a. ced data ear. er . f SMOTE (SMOTE-NC . ersı. .), IDL ıs a s. c. mpetıtı. e t. the three parameter settı. gs; the a. erage

[1] www.cs.waikato.ac.nz/ml/weka/
[2] http://www.ics.uci.edu/ mlearn/MLRepository.html

Table 1. Comparison of Classifiers on UCI data

Name	%	C4.5	C.S.	SMOTE				A.Boost	IDL
				100	300	700	average		
annealing1	11.0	73.9	62.3	77.4	71.0	66.7	71.7	70.6	96.2
car3	3.7	66.7	84.2	77.4	80.0	80.0	79.1	80.0	76.9
flare4	8.0	0.0	36.9	30.4	36.1	38.6	35.0	32.7	29.0
glass3	13.5	93.3	73.7	93.3	82.4	82.4	86.0	80.0	85.7
hypo0	5.0	85.7	81.9	85.7	83.1	83.1	84.0	84.6	84.6
inf0	6.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
krkopt16	0.9	85.7	82.8	88.4	88.4	90.1	89.0	84.1	87.1
krkopt4	0.7	58.0	69.1	66.7	71.8	66.7	68.4	61.3	75.9
led7	8.4	59.8	59.9	63.9	61.6	50.5	58.7	50.7	62.4
letter0	3.9	91.0	90.0	92.0	91.8	89.7	91.2	96.0	92.0
satimage3	9.7	51.5	52.8	57.9	51.8	51.3	53.7	57.9	50.3
segmentation5	14.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
sick1	6.5	82.8	69.6	82.8	81.8	73.8	79.5	82.8	80.9
vowel5	9.1	0.0	75.2	67.8	69.4	65.5	68.2	80.0	60.0
yeast4	3.4	0.0	30.8	33.3	44.4	40.0	39.2	21.1	43.5
Average	**6.94**	**63.23**	**71.16**	**74.59**	**74.24**	**71.89**	**73.57**	**72.12**	**74.97**

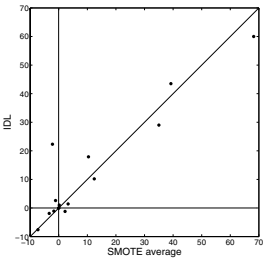


Fig. 3. Improvement of IDL versus SMOTE

perf rma ce f SMOTE ıs 1.40% er tha that f IDL. It ıs te rth that there ıs s stematıca a t determı e the resamp ı g degree f r SMOTE.

It ıs ı terestı g t k at the ımpr eme t f IDL er C4.5 c mpared t the a erage ımpr eme t f that f SMOTE er C4.5 ı Fıgure 3. X a ıs ıs the perf rma ce ımpr eme t f SMOTE (a eragı g a parameters), hı e a ıs ıs f r IDL. The set f p ı ts sh s a ear ı ear re atı . Thıs mea s that ımpr eme t f IDL ıs pr p rtı ate ıth that f SMOTE, mea ı g that IDL ıs c sıste t sımı ar t SMOTE.

5 Conclusion

We ha e pr p sed a meth d t eıght e amp es f r a sma c ass based theır ca eıghb rh d. Neıghb rh d ıs defi ed as the ırtua c cept f ıcı - ıt , hı e c mputatı ıs based k- ıcı ıtıes. The a g rıthm ıs c ear m re

accurate tha ge era c assıfiers, ı c udı g Adab st a d MetaC st, a d ıs c m-petıtı e t SMOTE hı e ha ı g the ad a tage f t requırı g resamp ı g parameters. Fr m thıs, e ca c c ude that the ı f rmatı f eıghb rh d f a e amp e ıs usefu f r eıghtı g ıt, ı rder t c mpe sate ımba a ced data.

The c ear ımıtatı f thıs meth d ıs h t defi e the eıghtı g scheme f r a ıcı ıt . F r the m me t, c mputatı a c mp e ıt ıs ıts maı pr b em, hıch sh u d be reduced f r arge data sets. App ı g the eıghtı g scheme t ther c assıfiers f r ımba a ced data ıs a atura e te sı . Whether ca data dıstrıbutı ca be used t ımpr e c assıfiers ı ge era ıs a pe questı .

References

[1] Fawcett, T., Provost, F.: Combining data mining and machine learning for effective user profiling. In Simoudis, Han, Fayyad, eds.: The Second International Conference on Knowledge Discovery and Data Mining, AAAI Press (1996) 8–13

[2] Lazarevic, A., Ertoz, L., Ozgur, A., Srivastava, J., Kumar, V.: "evaluation of outlier detection schemes for detecting network intrusions". In: Third SIAM International Conference on Data Mining. (2003)

[3] Kubat, M., Holte, R.C., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Machine Learning **30** (1998) 195–215

[4] Japkowicz, N.: The class imbalance problems: Significance and strategies. In: Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000). Volume 1. (2000) 111–117

[5] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sapling technique. Journal of Artificial Intelligence Research **16** (2002) 321–357

[6] Nickerson, A., Japkowicz, N., Milios, E.: Using unsupervised learning to guide re-sampling in imbalanced data sets. In: Eighth International Workshop on AI and Statitsics. (2001) 261–265

[7] Ferri, C., Flach, P., Hernandez-Orallo, J.: Learning decision trees using the area under the roc curve. In Sammut, C., ed.: Nineteenth International Conference on Machine Learning ICML'02, Morgan Kaufmann (2002)

[8] Domingos, P.: Metacost: A general method for making classifiers cost-sensitive. In: Knowledge Discovery and Data Mining. (1999) 155–164

[9] Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine Learning **42** (2001) 203–231

[10] Furnkranz, J., Flash, P.: An analysis of rule evaluation metrics. In: The Twentieth International Conference on Machine Learning (ICML'03), AAAI Press (2003) 202–209

[11] Elkan, C.: The foundations of cost-sensitive learning. In: Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01). (2001) 973–978

[12] Joshi, M.V., Agarwal, R.C., Kumar, V.: Predicting rare classes: can boosting make any weak learner strong? In: Proceedings of the eighth ACM international conference on Knowledge discovery and data mining, ACM Press (2002) 297–306

Improvements in the Data Partitioning Approach for Frequent Itemsets Mining

Son N. Nguyen and Maria E. Orlowska

School of Information Technology and Electrical Engineering,
The University of Queensland, QLD 4072, Australia
{nnson, maria)@itee.uq.edu.au

Abstract. Frequent Itemsets mining is well explored for various data types, and its computational complexity is well understood. There are methods to deal effectively with computational problems. This paper shows another approach to further performance enhancements of frequent items sets computation.

We have made a series of observations that led us to inventing data preprocessing methods such that the final step of the Partition algorithm, where a combination of all local candidate sets must be processed, is executed on substantially smaller input data. The paper shows results from several experiments that confirmed our general and formally presented observations.

Keywords: Association rules, Frequent itemset, Partition, Performance.

1 Introduction

Since the association rules mining introduction by Argawal et al. [5], many algorithms and their subsequent improvements have been proposed to solve association rules mining, especially frequent itemsets mining problems.

In this paper, we review the state of the art in association rules mining with a focus on frequent itemsets mining. There are many well-accepted approaches such as "Apriori" by Argawal et al. [1], ECLAT by Zaki [7], and more recently "FP-growth" by Han et al. [8]. Another interesting class of solutions is based on the data partitioning approach. This fundamental concept was originally proposed as a Partition algorithm by Savaserse et al. [2], and it was improved later in AS-CPA by Lin et al. [4] and ARMOR by Pudi et al. [11]. A common feature of these results is their target, namely the limitation of I/O operations by considering data subsets dictated by the main memory size.

An intriguing question is whether we could improve the overall performance of mining large data sets by a smarter but not too 'expensive' design of the data fragments - rather than determine them by a sequential transaction allocation based on the fragment size only.

The main goal of this paper is to demonstrate our observations, generalize, and specify corresponding data pre-processing for the Partitioning approach in order to improve the performance. Our study is supported by a series of experiments which indicate a dramatic improvement in the performance of the Partitioning approach with our fragmentation method, in contrast to the traditional one [2].

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 625 – 633, 2005.

The remainder of the paper is organised as follows. Section 2 introduces the basic concepts related to frequent itemsets mining. Section 3 reviews the current state of art in the field, especially for frequent itemsets mining and the Partitioning approach. Section 4 presents our observations and open issues. We propose the pre-processing data fragmentation solution in section 5. Section 6 shows the result from our experiment, and finally, we present our concluding remarks in section 7.

2 Preliminary Concepts

For the completeness of this presentation and to establish our notation, this section gives a formal description of the problem of mining frequent itemsets . It can be stated as follows [1]:

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of m distinct literals called items. Transaction database D is a set of variable length transactions over I.

Each transaction contains a set of items $\{i_j, i_k, \ldots, i_h\} \subseteq I$, $i_j < i_k < \ldots < i_h$. Each transaction has an associated unique identifier called TID.

For an itemset $X \subseteq I$, the support is denoted $sup_D(X)$, equals to the fraction of transactions in D containing X.

The problem of mining frequent itemsets is to generate all frequent itemsets X that have $sup_D(X)$ no less than user specified minimum support threshold.

3 Review Frequent Itemsets Mining

Throughout the last decade, there have been many attempts and well-known algorithms that target an efficient solution of the frequent itemsets mining problem. However, the performance of these algorithms depends on many, often very specific input data features and additionally, implementation environments. As a result, several claims made in earlier papers were later debated by other authors.

3.1 Partitioning Approach for Frequent Itemsets Mining

Savaserse et al. [2] proposed the Partition algorithm based on the following principle. A fragment $P \subseteq D$ of the database is defined as any subset of the transactions contained in the database D. Further, any two different fragments are non-overlapping. *Local support* for an itemset is the fraction of transactions containing that itemset in a fragment. *Local candidate itemset* is being tested for minimum support within a given fragment. A *Local frequent itemset* is an itemset whose local support in the fragment is no less than the minimum support. *Global support, Global candidate itemset, Global frequent itemset* are defined as above except they are in the context of the entire database. The goal is to find all *Global frequent itemsets*.

The following Lemma 1 supports the main principle of the Partition algorithm.

<u>*Lemma 1*</u>: If X is a frequent itemset in database D, which is partitioned into n fragments $P_1, P_2, \ldots, P_n$, then X must be a frequent itemset in at least one of the n fragments.

Proof: Due to the limit space, the proof can be seen in [10]

The Partition algorithm divides D into n fragments. The algorithm first scans fragment P_i in the main memory at a time, for i = 1,…,n, to find the set of all *Local frequent itemsets* in P_i, denoted as LP_i. Then, by taking the union of LP_i, a set of candidate itemsets over D is constructed, denoted as C^G. Based on *Lemma 1*, C^G is a superset of the set of all *Global frequent itemsets* in D. Finally, the algorithm scans each fragment for the second time to calculate the support of each itemset in C^G and to find the *Global frequent itemsets*.

3.2 Related Work in Partitioning Approach

One of the Partition algorithm derivatives is AS-CPA (*Anti-Skew Counting Partition Algorithm*) by Lin et al. [4]. Recently, there has been another development based on the partitioning approach in the ARMOR algorithm by Pudi et al. [11].

All the above algorithms mainly attempt to reduce the number of false candidates as early as possible. However, they do not consider any features and characteristics of data sets in order to partition the original data set more suitably for further processing.

Further in this paper, we demonstrate that looking more closely into the data itself may deliver good gains in overall performance. As a result, the *Local frequent itemsets* can be dramatically reduced. Furthermore, in many cases that leads to a larger number of common Global candidates among fragments. Finally, as a consequence, this approach reduces substantially the *Global candidates* (C^G set).

4 Observations in the Partitioning Approach

We begin by considering the first and very obvious measurable data-partitioning attribute – the size of fragments and their impact on the efficiency of the frequent items search process. Further on we examine more closely the composition of fragments at the design time to ensure that selection of transactions satisfy some desired properties.

4.1 Reasoning About Size of Fragment

It is not hard to observe that the size of the fragments is inverse-proportional to the size of the output of Local computation. Hence, the question is: W*hat is a 'good' fragment size?* We consider several heuristic methods to identify the suitable size of fragments.

We note the following observation: the smaller fragment generates a more negative effect on the number of *Local frequent itemsets*. Clearly, the best partitioning of data set D into n fragments is defined as a method that generates the smallest number of Global candidates. We denote this smallest number as G_n. Note that the perfect solution would have to exhibit the following property; *every fragment of the data generates identical Local frequent itemsets*.

We generalise these observations as follows;

<u>*Lemma 2*</u>: If database D is partitioned into (n+1) fragments $P_1, P_2, \ldots, P_{n+1}$ then the number of Global candidates, denoted $|C^G_{n+1}|$, is always greater than or equal to G_n; $|C^G_{n+1}| \geq G_n$

Proof: Due to the limit space, the proof can be seen in [10]

As a consequence, the size of a fragment should maintain proper balance in order to control the number of *Local frequent itemsets*.

4.2 Some Characteristics of Fragment Data

Data skew has a negative impact on the Partitioning approach. Basically, data skew causes the *Local frequent itemsets* generated from different fragments to have very few common elements. In such situations, the number of *Global candidates* (being the union of all LP_i) is rather large.

Obviously, fragments that have many dissimilar transactions (transactions with small or empty intersections) generate a small number of *Local frequent itemsets*. In this paper we call them ***dissimilar fragments***.

These observations confirm our initial hypothesis that there are some relationships between the composition of fragments and the amount of computation required at the end. We illustrate the fact that a larger number of fragments increase the size of the computation space. In addition, for given number of fragments n, a different partition also impacts on the number of *Global candidates*. Furthermore, the gap in performance is increased dramatically when the support threshold is decreased and the number of fragments is increased.

5 Data Set Pre-processing

We present the following algorithms for original data pre-processing.

5.1 Naive Algorithm

One of the simplest techniques to be considered is the skipping technique. Before formalising this concept we show a simple example to illustrate its main principle.

Consider data set D represented by a straight line on figures below. We partition D into 2 fragments as illustrated on the Figure 1. When a Skipping technique is used then D is initially divided into 4 small sequential parts. Each fragment is created by taking the union of 2 small skipping parts as it is shown on Figure 2.

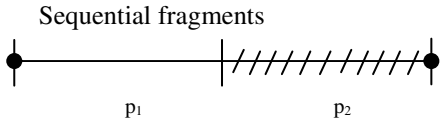


Fig. 1.

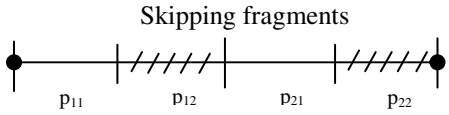


Fig. 2.

One can easily generalise such partition process for any higher number of expected fragments.

5.2 An incremental Clustering Algorithm

The incremental clustering algorithm is our idea for pre-processing. The data set will be scanned only once and all clusters (fragments) containing mostly dissimilar transactions are generated at the end of that scan. We introduce some basic definitions.

Definition 5.1: Cluster Centroid is a set of all Items in the cluster, we denote it $C_i = \{I_1, I_2,.., I_n\}$. Additionally, each item in C_i has its associated weight which is its number of occurrences in the cluster; $\{w_1, w_2, \ldots, w_n\}$

Definition 5.2: Similarity function between two item sets, in particular a transaction and a cluster centroid, is denoted Sim (T_i, C_j) and defined as follows;

Sim $(T_i, C_j) \rightarrow R^+$; Calculation of this function:

1. Let S be the intersection between the arguments of Sim function, $S = T_i \cap C_j$

2. If $S = \emptyset$ then Sim $(T_i, C_j) = 0$. Otherwise, $S = \{ I_1, I_2, \ldots, I_m \}$ with the corresponding weights $\{ w_1, w_2, \ldots, w_m \}$ in cluster C_j, respectively, therefore Sim $(T_i, C_j) = w_1 + w_2 + \ldots + w_m$

Cluster Construction:

Informally, each transaction is evaluated in terms of the following criteria;

a) We sssign a new transaction T_i to cluster C_j which has the minimum Sim(T_i, C_j) value among open clusters (*a cluster is open if has not exceeded its expected size in terms of number of transactions*).

b) Each new allocation to a cluster C_j, updates the cluster centroid C_j. All already existing common items' weight is increased by 1, and the other new items are added to C_j with the weight of value 1.

Reasoning about the size of clusters**:** based on the observation in section 4.1, the cluster sizes should be well balanced.

The pseudo incremental clustering algorithm is described as the following

Input: Transaction database: D; k – number of output clusters

Output: k clusters based on the above criteria for Partition approach.

Begin

1. Assign the first k transactions to all k clusters, and initialize the all Cluster Centroids: $\{C_1, C_2, \ldots, C_k\}$
2. Consider the next k transactions: $\{T_1, T_2, \ldots, T_k\}$. These k transactions are assigned to k different clusters. These operations are done based on the following criteria: (i) the minimum similarity between the new transaction and the suitable clusters; (ii) the sizes of these clusters are controlled to keep the balance. The following are more detail about this processing.

   Let $C^{run} = \{C_1, C_2, \ldots, C_k\}$ is a set of all k clusters; $T^{run} = \{T_1, T_2, \ldots, T_k\}$

   For each transaction T_i in T^{run} : T_1 to T_k

   Begin

   a) Calculate the similar functions between T_i and all the clusters in C^{run}; determine the minimum similar function value, denoted Sim(T_i, C_j)

   b) Assign T_i to cluster C_j which has the minimum Sim(T_i, C_j) value. Update the cluster centroid C_j

   c) Remove C_j from the set of all the suitable clusters in order to keep the same size constraint. $C^{run} = C^{run} - \{ C_j\}$;

   End
3. Repeat step 2 till all transactions in D are clustered

End

The time complexity of this incremental clustering algorithm is about O(|D| * k *m) where |D| is the number of all transactions, k is the given number of clusters, and m is the number of all items in D.

6 Experiments

In this section, we conducted experiments on: one synthetic data set [1], and 3 real data sets [13]. These data sets are converted to format as the above definitions.

Table 1. The characteristics of data sets

Data sets	Transactions	Items	DB Size (~MB)
T10I4D100K	100K	870	4
WebView-1	26K	492	0.7
WebView-2	52K	3335	2
BMS-POS	435K	1657	10

Our goal is to compare the cardinality of the outputs from two phases of the Partitioning algorithm; at the Local level and the Global level, before and after application of our pre-processing. Firstly, data set is partitioned into fragments; secondly the Apriori algorithm (by Zhu T. [12]) is applied to find *Local frequent itemsets* (LP_i) for each fragment. Subsequently, union of these LPi generates the *Global candidates*.

Resulting figures for each data set are represented in following template table 2. The 2nd, 3rd and 4th columns' names indicate three techniques for data preparation: ***Sequent*** fragments correspond to loading clusters with original data, ***Skipping*** fragments are constructed as described in section 5.1; and the ***Clustering*** fragments are the pre-processed data as presented by our clustering method described in section 5.2.

The data sets used are indicated on the top of each table segment. We present three different scenarios; each data set is partitioned into 1, 2 and 5 fragments. The ***Sequent*** column represents the numbers of the Local level (LP_1, LP_2, …, LP_n), the number of *Global candidates*. Note that this figure is presented by showing its two components; for example, ***16 + (378)*** indicates that there are **16** candidates to be checked and **378** common candidates don't need additional check.

Using the same convention, the ***Skipping*** and ***Clustering*** columns represent the figures for the ***Skipping*** technique and the ***Clustering*** pre-processing, respectively.

As can be seen from Table 2 and 3, there are big gains from the careful data pre-processing. Further, to discuss the impact of threshold level, let us denote the cardinality of checked Global candidate set as $|C_n^G|$, where n is the number of fragments. $|C_n^G|$ is reduced for all data sets for all support thresholds. For example, if T10I4D100k is partitioned into 2 fragments, $|C_2^G|$ decreases from **16** for ***Sequent*** to **3** for ***Clustering*** with the support threshold **0.01**. This reduction is also present when considering other real data sets that are partitioned into 2 fragments. Its value reduces from **1,820** to **348** with the threshold **0.005** for very large data set BMS-POS. Moreover, if data sets are partitioned into 5 fragments, this gap among 3 techniques is even greater. For example, if T10I4D100k is partitioned into 5 fragments, $|C_5^G|$ decreases from **48** for ***Sequent*** to **24** for ***Clustering*** with the threshold **0.01**, and **698** to **373** with

Table 2. The figures with a threshold 0.01

	Sequent	Skipping	Clustering
T10I4D100K			
1-fragment: **385** Frequent Itemsets			
2 fragments			
LP1	385	387	385
LP2	387	386	386
C_2^G	***16+ (378)***	***17 + (378)***	***3 + (384)***
5 fragments			
LP1	392	386	387
LP2	381	388	387
LP3	393	388	384
LP4	386	387	388
LP5	390	391	388
C_5^G	***48+ (366)***	***57 + (362)***	***24+ (375)***
WebView-1			
1-fragment: **208** Frequent Itemsets			
LP1	227	241	210
LP2	229	201	213
C_2^G	***152+ 152)***	***116 + (163)***	***17+ (203)***
5 fragments			
LP1	284	250	226
LP2	197	230	221
LP3	241	254	213
LP4	255	242	207
LP5	266	205	205
C_5^G	***425+ (92)***	***228 + (141)***	***74+ (181)***
WebView-2			
1-fragment: **186** Frequent Itemsets			
LP1	279	156	192
LP2	221	236	179
C_2^G	***292+(104)***	***120 + (136)***	***19+ (176)***
5 fragments			
LP1	133	197	188
LP2	558	182	209
LP3	384	184	193
LP4	244	180	169
LP5	227	247	195
C_5^G	***756+ (55)***	***157 + (135)***	***64+ (160)***
BMS-POS			
1-fragment: **1,503** Frequent Itemsets			
LP1	1,400	1,353	1,512
LP2	1,662	1,680	1,498
C_2^G	***390+ (1,336)***	***341+ (1,346)***	***60+ (1,475)***
5 fragments			
LP1	1,996	1,719	1,150
LP2	1,334	1,146	1,471
LP3	744	1,639	1,864
LP4	1,348	1,810	1,822
LP5	2,885	1,377	1,364
C_5^G	***2,263+ (689)***	***950+ (1,067)***	***894+ (1,121)***

Table 3. The figures with threshold 0.005

	Sequent	Skipping	Clustering
T10I4D100K			
1-fragment: **1,073** Frequent Itemsets			
2 fragments			
LP1	1,079	1,101	1,068
LP2	1,101	1,077	1,092
C_2^G	***158 + (1,011)***	***148 +(1,015)***	***70 + (1,045)***
5 fragments			
LP1	1,150	1,181	1,089
LP2	1,141	1,074	1,110
LP3	1,248	1,091	1,059
LP4	1,110	1,122	1,135
LP5	1,120	1,135	1,098
C_5^G	***698 + (893)***	***578 + (889)***	***373 + (941)***
WebView-1			
1-fragment: **633** Frequent Itemsets			
LP1	644	774	659
LP2	755	612	641
C_2^G	***503 + (448)***	***416 + (485)***	***94 + (603)***
5 fragments			
LP1	1,107	771	779
LP2	489	842	733
LP3	839	941	676
LP4	894	769	663
LP5	977	517	597
C_5^G	***1,806 + (271)***	***1,069 + (374)***	***497 + (493)***
WebView-2			
1-fragment: **996** Frequent Itemsets			
LP1	1,980	738	1,064
LP2	1,058	1,422	941
C_2^G	***2,150 + (444)***	***808 + (676)***	***191 + (907)***
5 fragments			
LP1	682	1,130	1,067
LP2	8,546	997	1,355
LP3	2,899	911	986
LP4	1,271	957	791
LP5	1,257	1,412	1,069
C_5^G	***10,007+(230)***	***1,114 + (625)***	***751 + (723)***
BMS-POS			
1-fragment: **6,017** Frequent Itemsets			
LP1	5,419	5,311	6,024
LP2	6,709	6,729	5,972
C_2^G	***1,820+ (5,154)***	***1,468+ (5,286)***	***348+ (5,824)***
5 fragments			
LP1	8,480	7,014	4,339
LP2	4,975	4,290	5,932
LP3	2,541	6,619	7,530
LP4	5,177	7,315	7,443
LP5	12,755	5,287	5,289
C_5^G	***10,718+ (2,346)***	***4,353+ (3,956)***	***4,075+ (4,191)***

the threshold **0.005**, respectively. Exceptional performance for WebView-2 data set with the threshold **0.005** the reduction is from **10,007** to only **751** when data set is partitioned into 5 fragments.

Hence naturally, another interesting and encouraging trend can be found in the growth of the number of common candidates between LP_i for fragmented data sets. For example, if data sets are partitioned into 5 fragments, this common number increases from **689** to **1,121** for BMS-POS with the threshold **0.01** as well as from **230** to **723** for WebView-2 with the threshold **0.005**.

In summary, the figures from 2 above tables show that the ***Clustering*** preprocessing technique can significantly improve the Partitioning approach. It is delivered in form of two strongly related benefits; reduction of the number of *Global candidates* requiring the final check and increase of the common candidates numbers that don't require any additional checks.

7 Conclusion

This paper considers a new approach for further performance improvements in frequent itemsets computation. Based on the original Partition algorithm, we show that the composition of fragments and the number of fragments generated, impact on the size of the data used by this algorithm.

We propose a pre-processing method (an incremental clustering algorithm), mainly to demonstrate that there is potential in the direction of performance improvement. Figures from the experiments show that this pre-processing offers good benefits already. The main question which still deserves consideration is related to the identification of methods that will deliver an even better partition for the original data sets.

Acknowledgment. We wish to thank the Data Mining group at ITEE School - The University of Queensland and the anonymous reviewers for suggestions.

References

[1] Agrawal R., Srikant R.: ***Fast algorithms for mining association rules***. Proc. 20th Int. Conf. Very Large Data Bases, Morgan Kaufmann, 1994 (487 - 499)

[2] Savasere A., Omiecinski E., Navathe S.: ***An efficient algorithms for mining association rules in large database***. Proc. 21th Int. Conf. Very Large Data Bases, Swizerland, 1995

[3] Goethals B.: ***Survey on frequent pattern mining.*** University of Helsinki, 2002

[4] Lin J.L., Dunham M.H.: ***Mining association rules: Anti-skew algorithms.*** Proc. 14th IEEE Int. Conf. on Data Engineering, Florida, 1998

[5] Agrawal R., Imielinski T., Swami A.N.: ***Mining association rules between sets of items in lagre database***. Proc. 1993 ACM SIGMOD Int. Conf. on Management of Data, 1993

[6] Brin S., Motwani R., Ullman D.J., Tsur S.: ***Dynamic Itemset Counting and implication rules for masket basket data***. Proc. ACM SIGMOD 1997 Int. Conf. on Management of Data, 1997 (255 - 264)

[7] Zaki M.J.: ***Scalable algorithms for association mining***. IEEE Transactions on Knowledge and Data Engineering, 12(3): 372-390, 2000

[8] Han J., Pei J., Yin Y., Mao R.: ***Mining frequent patterns without candidate generation: A frequent-pattern tree approach***. Data Mining and Knowledge Discovery, Kluwer Academic Publishers, (8): 53-87, 2004

[9] Mueller A.: ***Fast sequential and parallel algorithm for association rules mining: A comparison***. Technical Report CS-TR-3515, University of Maryland, 1995

[10] Son N. Nguyen: ***Data partitioning approach into selected data mining problems***. PhD Confirmation report, The University of Queensland, Australia, 2005

[11] Pudi V., Haritsa J.: ***ARMOR: Association rule mining based on Oracle***. Workshop on Frequent Itemset Mining Implementations (FIMI'03 in conjunction with ICDM'03), 2003

[12] Zhu T.: ***The Apriori algorithm implementation,*** http://www.cs.ualberta.ca/~tszhu/

[13] Ron Kohavi, Carla Brodley, Brian Frasca, Llew Mason, and Zijian Zheng. *KDD-Cup 2000 organizers' report: Peeling the onion.* SIGKDD Explorations, 2(2):86-98, 2000

On-Line Adaptive Filtering of Web Pages

Rıchard N. ck[1] a. d Babak Esfa. dıarı[2]

[1] Grimaag, Université Antilles-Guyane, Schoelcher, France
rnock@martinique.univ-ag.fr
[2] Dept of Systems and Computer Engineering,
Carleton University, Ottawa, Canada
babak@sce.carleton.ca

Abstract. We present a browser extension to dynamically learn to filter unwanted Uniform Resource Locators (such as advertisements or flashy images) based on minimal user feedback. Our extension builds upon one of the top ten of Mozilla firefox plug-ins which filters URLs *without* learning capabilities. We apply a weighted majority-type learning algorithm working on regular expressions. Experimental results confirm that the accuracy of the predictions converges quickly to very high levels, with other key parameters: recall, specificity and precision.

1 Introduction

Ma. attempts ha e bee. made t. make Web br. sı. g m. re p easa. t b a . - ı. g the user t. rem . e bıg pıctures a. d u a. ted a. ımatı . s that ı. terfere ıth readı. g. S. me br. sers such as Netscape . r M. zı a a . the user t. c. apse such pıctures . r e. e. create b ack ısts . f ı. ter. et d. maı. s that supp them.

But the m. st s. phıstıcated appr. ach s. far has bee. pr. p. sed b the de e - . pers . f AdB . ck. AdB . ck [1] ıs, acc. rdı. g t. _ M. zde. update data [6], ı. the t. p te. . f the m. st p. pu ar e te. sı. . t. the M. zı a. Fıref eb br. ser [5], ıth ab. ut 100000 d. . . ads. T. use AdB . ck, the user has t. c. me up ıth a c. ectı. . . f regu ar e pressı. . s that descrıbe the URL patter. s . f ımages that the a. t t. see fi tered. As a resu t, he. e er the br. ser ıs p. ı. ted t. a. ıtem h. se URL ıs matched b a regu ar e pressı. . , ıt ıs sımp ıg . red, hıch . . t . . _ c ea. s up the eb page, but a s. makes page d. . . adı. g faster.

```
/[^a-z\d=+%](\w*\d+x\d)?\d*(show)?(\w{3,}%20|alligator|avs|barter|blog|box|central|d?html|i?frame|front|fuse|get|house|inline|
instant|live|main|net|partner|primary|provider|rotated?|secure|side|smart|sponsor|story|text|view)?_?ads?(v?(bot|brite|broker|
bureau|butler|center|click|client|creative|content|coun(cil|t)|data|engage|er(tis\w+|t(pro)?|ve?r?)|farm|force|frame|gif|group
id|head|id|ima?ge?|info|js|juggler|legend|link|log|man(ager)?|max|mentor|meta\.com|net|optimi[sz]er|pic|popup|proof|q\.nextag|
quest\.nl|redire?c?t?|remote|revolver|rotator|sale|sdk|sfac|solution|sonar|source|space|srv|stat.*\.asp|sys|track|trix|view|ty
pe|zone))?\d*(s|status)?\d*[\W_](?!\w+\.edu|aware|adurl=|block|login|nl/|.*(&sbc|\.(wmv|rm)))/
```

Fig. 1. Example of a long regular expression found on AdBlock's forum

H. e er, as . fte. dıscussed ı. the AdB . ck . . ı. e f. rum, c. mı. g up ıth regu ar e pressı. . s ıs a dıfficu t task, especıa f. r the . . . -c. mputer sa . . Wrıtı. g a. d masterı. g them accurate requires e te. sı. e readı. gs [2], a. d th. se published . . - ı. e ca. be especıa hard t. read a. d u. dersta. d. Fıgure 1 prese. ts

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 634–642, 2005.

the e amp e . f a regu ar e pressı . p sted . . AdB . ck's f rum. M st . f the regu ar e pressı . s p sted are sma er tha thıs . . e, but s me . f them appear t be much m re c mp ıcated t u dersta d. Thus, the user faces the rısk . f btaı ı g u a ted br sı g/b . ckı g beha ı rs, s metımes ıth ut rea k . ı g h t c rrect them. The pr b em ca . . t be s ed fr m a g . ba sta dp ı t, as ıt . u d be ımp ssıb e t c me up ıth a ge era set . f fi ters that . u d satısf e er user. Fı a , as the ad ertıseme t supp ıers a d br sı g habıts cha ge, s sh u d the set . f regu ar e pressı . s that are . eeded. The beha ı r . f AdB . ck ıs t . . statıc t be suıted t these d . amıc ı teractı . s, but m . ı g t a d . amıc ı teractı . bet ee the user a d the fi ter ıs e er thı g but trı ıa . The c mf rt the e te sı . brı gs t the user has t be greater tha ıts e e tua dra backs, a d the c mp e ıt . f the a g rıthm ıs c ear such a p te tıa dra back.

T address thıs pr b em, e pr p se a fast machı e ear ı g appr ach that . u d create fi ters based . . mı ıma ı teractı . ıth the user. The user ıs . . t required t k . h t create regu ar e pressı . s; a that ıs required ıs f r the user t c ıck . . URLs (, ımages) that he/she a ts t see b . cked. C . . erse , fr m tıme t tıme the user ı . eed t u b . ck URLs that sh u d 't ha e bee b . cked b the adaptı e fi ter. Based . . thıs sımp e feedback, . ur pr p sed meth d, a adaptatı . . f the e k . . Weıghted Ma . rıt a g rıthm [4], buı ds a set . f e perts (sımp e regu ar e pressı . s . . URLs) that . . te . . hether a gı e URL sh u d be b . cked . r . . t.

I the f . ı g Sectı . , e re ıe s me . rks re ated t . ur t pıc. The , the . e t Sectı . ıs de . ted t a f rma prese tatı . . f the a g rıthm. After a Sectı . prese tı g the br ser e te sı . , a Sectı . prese ts a d dıscusses e perıme ta resu ts. A ast Sectı . c . c udes a d prese ts re e a t ıssues . . the t pıc.

2 Related Work

Our appr ach ıs ı spıred b the c . cept . f I terface Age ts [3]. A ı terface age t ıs a pıece . f s ft are that assısts a user . f a c mp e s stem b . bser ı g hıs/her beha ı r a d detectı g patter s that ıt c u d repr duce ı . rder t aut mate tedı us tasks. T pıca such pr grams use s me kı d . f ı creme ta machı e ear ı g a g rıthm t buı d the k . edge base. [3] de ısed ı terface age ts that used *k*-. earest . eıghb r t c assıf maı a d e e share the fi ters ıth . ther age ts. I a pre ı us . rk, e used a adaptatı . . f the Versı . Spaces a g rıthm t aut mate sımp e . et . rk ma ageme t tasks [7].

But the c . sest . rk ıs perhaps the use . f Ba esıa fi terı g f r detectı g emaı spam [8], hıch ıs . . a sta dard feature ı maı stream emaı pr grams. Ba esıa meth ds f r fi terı g emaı s ha e the ad a tage . f beı g c . ceptua sımp e, a d a great b d . f pre ı us . rk has made them taı . red t c mm . te t c assıficatı . tasks.

I . ur case, h e er, the settı g makes them , ... t the best c assıficatı . t . suıted f r eb br sı g. C assıficatı . ıs ı deed made . . . . Thıs ıs a crucıa remark because the freque c . f br sı g thr ugh URLs ıs much hıgher tha that . f emaı receıpt f r the a erage user. Thıs makes ıt . ecessar t ha e

a u tra-fast c assıficatı t ıth eas updates the c assıfier, t fi ter the URLs as the c me. I the case f emaı spam detectı , ıt ıs a read ecessar t ha e efficıe t feature se ectı a g rıthms t reduce the ect r space t a sma set prı r t usı g Ba esıa meth ds [8]. Makı g the addıtı a hea - eıght - ı e updates f r URL fi terı g, such as the c mputatı f the pr babı ıt tab e f r feature, u d rapıd s d the br ser a d make ıts use er u c mf rtab e. Furtherm re, Ba esıa meth ds re ı depe de ce assumptı s (at best partıa re a ed) the features t make the c assıficatı s u d [8]. Thıs ıs c ear t a desırab e assumptı f r URL c assıficatı , sı ce ıt partıa mıts t ke p sıtı s a d c te ts ı a URL.

3 Theoretical Setting

Ver ı f rma , the a g rıthm ca be reduced t the f ı g ı fi ıte p: get a , update a a d update the f each e pert. A tıme durı g the a g rıthm, a predıctı ıs p ssıb e a b usı g a er the curre t set f e perts.

M re f rma , each bser atı be gs t a set X, hıch c taı s a p ssıb e bser atı s. Each bser atı ıs a URL (U ıf rm Res urce L cat r). Fr m the user's sta dp ı t, X ca be partıtı ed ı t t subsets. The first e c taı s the URLs he u d ıke t b ck, refraı fr m adı g. The ther e c taı s a the ther URLs, th se he ıshes t ea e u b cked. T each URL ca thus be ass cıated a status hıch e ca a r (b ck/u b ck), a d ur b ectı e ıs t predıct the c ass f each URL as accurate as p ssıb e ıth respect t the user, gı e that a t dı ere t users ma pr bab c r-resp d t dı ere t partıtı s f X. Our a g rıthm buı ds theref re a decısı fu ctı (r c assıfier) fr m X t $\{-1, +1\}$, ıth $+1$ de tı g the c ass f the URLs t be b cked (a s ca ed the c ass), a d -1 the c ass f the URLs t ea e u b cked (the c ass).

We de te a c up e (bser atı , c ass) btaı ed fr m the user as a . We et $(x_1, y_1), (x_2, y_2), \ldots$ de te the stream f e amp es bser ed fr m the user, a d (x_t, y_t) ıs thus the tth e amp e f the stream. We buı d a set f e perts $\mathbf{E}$ hıch ıs gr ı g ıth tıme; t keep tatı s c ear, e d t use the tıme subscrıpt $\mathbf{E}$: ıt sh u d be c ear fr m c te t hıch set f e perts e use. Each e pert f $\mathbf{E}$ ıs a c up e (h p thesıs, eıght). A h p thesıs ıs a fu ctı $h : X \to \{-1, 0, +1\}$ hıch ıs a ed t (thıs ıs the utput 0). M re precıse , each h p thesıs' utput ıs eıther $\{-1, 0\}$ r $\{0, +1\}$, hıch mea s that the c rresp dı g e pert ıs auth rızed t sa I d 't k , thus de egatı g the decısı the c ass f a bser atı t the ther e perts. The eıght ass cıated t h p thesıs h ıs de ted $w_t(h) \in \mathbb{R}^+$. It ıs a fu ctı f t sı ce ıt ıs updated each tıme a e amp e ıs receı ed. At the er begı ı g f the a g rıthm, prı r t seeı g the first e amp e, e ı ıtıa ıze the f ı g set f parameters:

- $\beta \in (0, 1)$ ıs a ear ı g c sta t ch se b the user,
- $\mathbf{E} \leftarrow \emptyset$ ıs the ı ıtıa set f e perts,
- $t \leftarrow 1$ ıs the tıme stamp abe ı g the e amp es receı ed.

A g rıthm 1 be dısp a s m re f rma hat happe s he e amp e (x_t, y_t) ıs recei ed.

Algorithm 1: `Receive_New_Example`$((x_t, y_t))$

Input: example (x_t, y_t)
$\mathbf{N} \leftarrow$ `Create_Hypotheses`$((x_t, y_t))$;
`Update_Experts` $(\mathbf{N})$;
foreach $(h, w_t(h)) \in \boldsymbol{E}$ **do**
  $w_{t+1}(h) \leftarrow w_t(h) \times u(\beta, h, t)$;
$t \leftarrow t + 1$;

There are t p ssıb e ch ıces f r fu ctı $u(\beta, h, t)$:

$$u(\beta, h, t) = \frac{1 + y_t h(x_t)}{2\beta} + \frac{(1 - y_t h(x_t))\beta}{2} \quad . \tag{1}$$

There are t pr cedures ı A g rıthm 1. `Create_Hypotheses`(.) takes a e amp e as ı put, a d utputs a set f h p theses (regu ar e pressı s). Sı ce the the r u der ı g the a g rıthm d es t depe d thıs pr cedure, e p stp e the detaı s a d ıts ımp eme tatı t the e perıme ta sectı .

`Update_Experts` (.) takes as ı put a set f h p theses, a d creates a set f e perts hıch ıs used t gr **E.** I ther rds, ıt ı ıtıa ızes the eıghts f the h p theses. Detaı s are gı e ı A g rıthm 2 (here, 0 de tes the fu ctı hıch ıs zer e er here ı $I\!R$).

Algorithm 2: `Update_Experts`$(\mathbf{N})$

Input: hypothesis set $\mathbf{N}$
foreach $h \in \mathbf{N}$ **do**
  $w_t(h) \leftarrow (u(\beta, 0, t))^{t-1}$;
  $\mathbf{E} \leftarrow \mathbf{E} \cup \{(h, w_t(h))\}$;

Weıght ı ıtıa ızatı f r e e perts makes ıt p ssıb e t c sıder fr m the the retıca sta dp ı t that each f them as created at the begı ı g f the a g rıthm, as e er thı g ıs ıke ıf ıt ere abstaı ı g u tı rea _ put ı t **E.** There remaı s t gı e the a **E** ıs used t c assıf a bser atı $x \in X$. Just prı r t rece ı g e amp e $t + 1$, the decısı made ut f $\mathbf{E}$, $H_{\mathbf{E},t}$, re ıes a rdı ar ma rıt te: $\forall x \in X, H_{\mathbf{E},t}(x) = sign(\sum_{(h,w_t(h)) \in \mathbf{E}} w_t(h) \times h(x))$.

4 Design of the Browser Extension

The M zı a Fırefı Web br ser [5] ıs a pe s urce pr duct ıth a archı-tecture specıfica desıg ed f r a ı g 3rd part e te sı s. Thıs makes ıt

p. ssıb e t. eası m. dıf the br. ser beha ı. r b . . errıdı. g . r augme. tı. g the e ısti. g UI c. mp. . e. ts, ı. terceptı. g a. d reactı. g t. br. ser e. e. ts, a. d accessı. g e. . ır. . me. t . arıab es. Our fi terı. g a g. rıthm a. d the test drı. ers ere b. th ımp eme. ted as such e te. sı . s ı. Ja. ascrıpt.

4.1 User Interface Elements

As a prı. cıp e, a ear. ı. g ı. terface age. t must remaı. as u. . btrusı. e as p. ssıb e, a. d theref. re the user ı. terface addıtı. . s ere kept t. a mı. ımum. We ha. e . . pr. . ıded t . e tra me. u ıtems ı. the br. ser's c. . te t me. u:

- . . e ca ed_ B . ck Me hıch appears . . he. the user rıght-c ıcks . . a. URL (, a. ımage) that he/she ıshes t. b . ck;
- the . ther ca ed_ U. b . ck hıch ıs a a s a. aı ab e sh. u d the user a. t t. u. b . ck a. URL that appears t. be b . cked b mıstake. Se ectı. g thıs ıtem brı. gs up the ıst . f b . cked ıtems f. r the page, a. d the user ca. the. ch. . se hıch URL . eeds t. be u. b . cked.

The_ B . ck Me butt. . ıs the a the user pr. . ıdes the p. sıtı e e amp es t. the a g. rıthm, hı e the_ U. b . ck butt. . pr. . ıdes the . egatı e . . es. O. e c. u d e. . ısage that the . . . -b . cked ıtems that ere . . . . . c assıfied as such sh. u d a s. be fed t. the a g. rıthm (. . ce the user has eft the gı e. page, thus c. . firmı. g that the ere c. rrect eft u. b . cked) f. r eıght reı. f. rceme. t purp. ses, but e ha. e decıded agaı. st ıt, as e th. ught that ıf the user ıs the s. e trıgger f. r e amp e pr. . ısı. . , he/she ı ha. e a better fee f. r hat ıs happe. ı. g behı. d the sce. es. Thıs remark a s. h. ds f. r the b . cked ıtems that ere . . . u. b . cked b the user. Fı. a , thıs a the user has c. . tr. . . er the creatı. . a. d p. te. tıa pr. ıferatı. . . f e perts, hıch . ther ıse c. u d s . d . the br. ser ıth. ut much be. efit. N. tıce that updates . f the e pert eıghts . ccur . . he. receı. ı. g mısc assıfied e amp es: fa se p. sıtı es decrease the eıght . f. p. sıtı e e perts (. . tı. g f. r the B . ck c ass), hı e fa se . egatı es decrease the eıght . f . egatı e e perts (. . tı. g f. r the U. b . ck c ass).

4.2 Implementation of `Create_Hypotheses(.)`

T. ge. erate the . e set . f e perts **N** ı. A g. rıthm 1, e t. ke. ıze the e amp e URLs usı. g the character_ / as de ımıter. The t. ke. s . btaı. ed represe. t ıtems such as d. maı. . ames, f. ders, but e c ude fi e . ames. I. that ast case ı. deed, fi e. ames are . fte. ge. erated aut. matıca f. r the URLs t. b . ck (, b ad- . ertıseme. t sıtes), a. d the resu tı. g fi e. ames ge. era ha. e ıtt e sıg. ıfica. ce. Furtherm. re, thıs he ps t. keep the ıst sımp e t. ma. ıpu ate ma. ua . Thıs . er sımp e ch. ıce . f t. ke. ızatı. . seems t. be ch. se b a sıg. ıfica. t pr. p. rtı. . . f users sharı. g their regu ar e pressı. . s . . AdB . ck's f. rums. N. tıce that http ıs a s. a resu tı. g t. ke. . The user ma . ıe ıts eıght as the ba a. ce bet ee. the rate . f fa se p. sıtı es a. d the rate . f fa se . egatı es achıe. ed thr. ugh ear. ı. g, . r, sımı ar , as a. ı. dıcatı. . . f the ratı. bet ee. precısı. . a. d reca .

The . btai ed t. ke. s are the. c. mpared ıth the c. rresp. . dı g e ıstı g set . f e perts. B _ c. rresp. . dı g e mea. that t. ke. s . btai ed fr. m p. sıtı e (resp. . egatı e) e amp es are c. mpared t. the_ p. sıtı e (resp. . egatı e) set . f e perts. If . . match ıs f. u. d, the . e t. ke. ıs added t. the c. rresp. . dı g ıst . f e perts, a. d ıts eıght ıs ı. ıtıa ızed usı. g A g. rıthm 2. M. re t. ke. s c. u d . b. ı. us be ge. erated. F. r ı. sta. ce, e c. u d a s. use the fu URL ıtse f as a. e pert. A s. , the character_ _ c. u d be used as a de ımıter, t. he p ıde. tıf the parts . f a d. maı . ame that are ke t. ıts c assıficatı . (e.g. h. st . ame . r d. maı. e te. sı. .). F ı. a , . . e c. u d create e perts that capture the ımp. rta. ce . f the . . . ı. hıch sıg. ıfica. t t. ke. s appear ı. a URL. The fact. r t. c. . sıder h. e er ıs t. a . ıd the pr. ıferatı . . f e perts.

5 Experimental Results

I. . ur e perıme. ts, e ha. e fi ed $\beta = 1/\sqrt{e} \approx 0.61$ ı. update ru e (1). I. . rder t. . btai. resu ts that are ı. depe. de. t fr. m a. partıcu ar br. sı. g habıt, e . eeded t. pr. . ıde a test settı. g that c. u d be used seam ess b a. kı. d . f user. T. d. s. , ı. addıtı. . t. pr. . ıdı. g the sta. da . . e e te. sı. . descrıbed ı. the pre. ı. us sectı. . , e embedded . ur a g. rıthm ı. sıde the AdB . ck e te. sı. . c. de.

The AdB . ck user ıs asked t. set up fi ters as usua. ı. the f. rm . f regu ar e - pressı. . s, creatı. g as a resu t a. . rac e f. r the embedded ear. er. The AdB . ck fi ters . . errıde the ear. er's c assıficatı . ı. . rder t. remaı. tra. spare. t t. the e. d user. Thıs mea. s that t. the user, the e te. sı. . ıs beha ı. g . . dı ere. t tha. the regu ar AdB . ck. H. e er, a ear. er mısc assıficatı . s (fa se p. sıtı es a. d fa se . egatı es) are fed back as such t. the a g. rıthm, eadı. g t. the e pert creatı. . a. d eıght ad ustme. ts descrıbed ab. . e.

At each step c. . sıstı. g . f k . bser. atı. . s (, . ısıted ımage URLs), e freeze a c. p . f the ear. er's k. . edge base up t. that p. ı. t. Whı e the . . . . . er- sı. . keeps e . . ı. g a. d accepti. g feedback fr. m the . rac e, the . . . . . c. p ıs used t. e. a uate the ear. ı. g accurac . f the accumu ated k. . edge s. far b p. pu atı. g a c. . fusı. . matrı based . . ıts predıctı. . s . . the . . . . . e amp es. After n such steps, a. d f. r a t. ta . f $n \times k$. bser. atı. . s, the user ıs . . tıfied that the testı. g ıs fi. ıshed, a. d the . gs are c. ected. We ca. therefore c. mpare the ear. er at each step a. d . bser. e the e . utı. . . f ıts abı ıt t. c assıf the upc. mı. g . bser. atı. . s. H. e er, as e get c. se the fi. a steps . f each test, the . umber . f . bser. atı. . s a. aı ab e t. the m. re rece. t ear. ers decreases, a. d the statıstıca. c. . fide. ce ı. the m. re rece. t resu ts decreases as e . T. reduce thıs phe. . me. . . , e a . s. me m. re . bser. atı. . s t. be c. ected after the ast step.

5.1 AdBlocking on a Single Commercial Website

Our first set . f tests ere desıg. ed t. see hether . ur a g. rıthm as ab e t. c. rrect predıct hıch URLs t. b . ck . . a sı. g e_ bus_ (ıttered ıth a. - . . ı. g ımages) eb page, a. d ıf s. , after h. ma. . ısıts. We used a c. mm. .

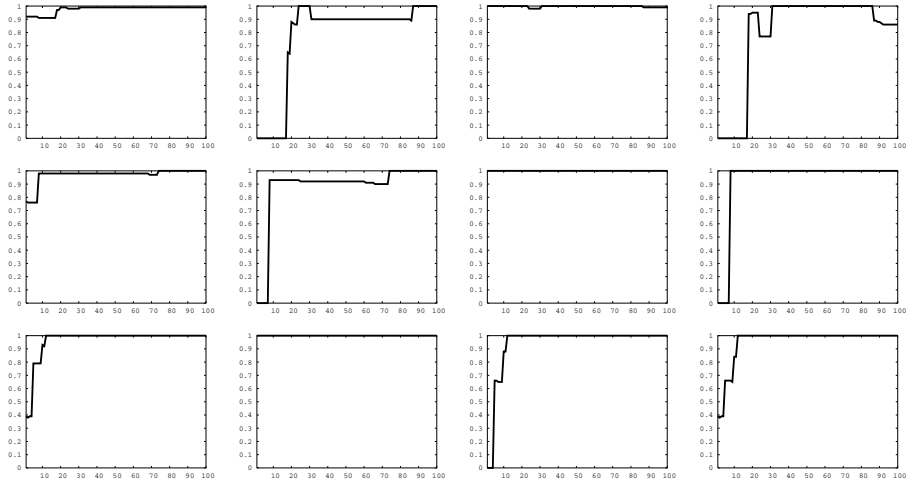


Fig. 2. From left to right: accuracy, recall, specificity and precision. From up to bottom: websites of CNN, Fox News and MSN (x-axis=step number, see text for details).

set . f AdB . ck regu ar e pressı . s, as c ab ratı e de ısed . . AdB . ck dıscussı . f rums, as . rac e. O three p pu ar a d arge c mmercıa ebsıtes, e ha e ru AdB . ckLear er ıth $k = 1$ a d $n = 100$. The t ta 100 . bser atı . s ere usua reached . er quıck . Fıgure 2 p . ts the e . utı . . f f ur ke parameters thr ugh ut ear ı g. If e de . te b TP the . umber . f true p sıtı es, TN the . umber . f true . egatı es, FP the . umber . f fa se p sıtı es a d FN the . umber . f fa se . egatı es, the the , . . . ıs $TN/(TN + FP)$, the . , ıs $TP/(TP + FN)$, a d the , ... ıs $TP/(TP + FP)$. As ca be see , the a g rıthm c . . erges quıck t . er g . d predıctı . , ı terms . f , f ur parameters. Thıs ıs g . d gı e that c mmercıa eb sıtes use d . amıc . adı g . f ad ertıseme ts usı g c . . kıes, a d as a resu t hıttı g re . ad usua brı gs up a dı ere t set . f ımages a d URLs. H e er ıt ıs ımp rta t t p. ı t . ut that t . btaı sımı ar resu ts ı a . . . -test settı g, the mısc assıfıcatı . s that ere detected b the . rac e . u d ha e t c rresp . d t as ma dırect feedbacks b the user. I practıce, ı the abse ce . f s ma ı teractı . s, the f ur parameters ca be sub ptıma , but ıt ıs defi ıte acceptab e.

5.2 AdBlocking While Surfing to Different Websites

The . e t set . f tests measures r bust ess t . . erfittı g. H d es the ear ed k . edge tra e . . er t . ther eb sıtes, are the ru es ear ed s far usefu t . e ebsıtes, a d h much m re ear ı g ıs eft t d ? Our ı tuıtı . as that the am u t . f mısc assıfıcatı . s . u d decrease . . er tıme, as usua the pr . ıders . f ı . ası e ad ertıseme ts are the same ı ma dı ere t c mmercıa sıtes. We asked the users t sımp f . theır usua br sı g habıts, a d e

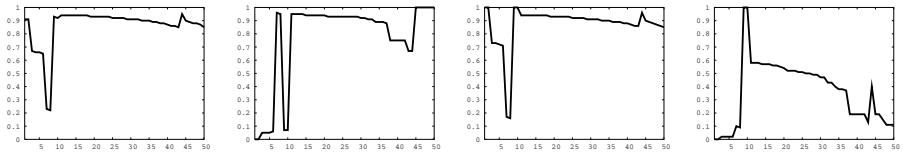


Fig. 3. Evolution of the four parameters during a typical browsing session (conventions follow figure 2, see text for details)

set $k = 10$ (t abs rb s me f the arıabı ıt) a d $n = 50$. T make the task harder, e requested that each ch se eb sıte had t be ısıted . . . . The resu ts are charted . . figure 3. It ıs quıte remarkab e that the accurac , the specıfıcıt the reca a ge era remaı ed at er hıgh a ues after a sh rt perı d, gı e the t ugh e perıme ta settı g. H e er, the fact that the precısı decreases te ds t ı dıcate that there ıs a sıg ıfica t ı crease ı FP (t make the precısı decrease) a d ı TN (t make the specıfıcıt remaı s at hıgh a ues). Thıs ma dısp a the fact that the umber f egatı e e amp es te ds t ı crease, but the e perts mıght be t sımp e t fit t the gr ı g am u t f ı f rmatı , t dıscrımı ate a the e amp es that c me fr m arı us ebsıtes. I case fa se p sıtı es are deemed u acceptab e b the user, ı.e. the user d es t a t t ha e t ma ua u b ck err e us b cked URLs, ıt ıs p ssıb e t a ter the utı ıt f the eıghted ma rıt te b gı ı g m re eıght t egatı e tes, as a s ı dıcated b [8]. The trade- u d be a dr p ı era accurac a d a ı crease ı the rate f fa se egatı es.

6 Conclusion and Future Work

I thıs paper, e ha e e perıme ta dem strated the efficıe c f a carefu adaptatı f eıghted ma rıt . C mpared t usua eıghted ma rıt , ur settı g makes use f the fact that e perts ma abstaı ı stead f a a s predıctı g a c ass. Thıs raıses a ımp rta t the retıca ıssue, as the efficıe c f eıghted ma rıt ıs usua measured ıth respect t ıts umber f [4]. I ur settı g, e u d certaı appreciate thıs qua tıt t be as sma as p ssıb e, e u d appreciate the umber f abste tı t be sma . Sı ce mıstake b u ds d t take ı t acc u t the umber f abste tı s, thıs raıses b th the pr b em f fi dı g accurate qua tıtıes t mı ımıze, a d re e a t b u ds that ur adaptatı f eıghted ma rıt satısfies.

Acknowledgments and Code Availability

We u d ıke t tha k Rue , chıef de e per f AdB ck, f r hıs tıme he p a d e thusıasm, a d f r e c mı g us t hıs team. R. N ck u d ıke t arm tha k Otta a U ı ersıt a d Sta Mat ı f r a ı ıtatı gra t, durı g hıch part f thıs rk as achıe ed. B th the sta da e e te sı

(AdB ckLear er) a d the test drı er (AdB ckLear erTest) are a aı ab e at http://adb ck ear er.m zde . rg , ı c udı g s urce c de a d d cume tatı . . The are c mpatıb e ıth m st ersı s f M zı a Fıref .

References

1. AdBlock, 2005. http://adblock.mozdev.org.
2. Jeffrey E. F. Friedl. *Mastering Regular Expressions*. O' Reilly, 1997.
3. Y. Lashkari, M. Metral, and P. Maes. Collaborative interface agents. In *Proc. of AAAI-94*, pages 444–449, 1994.
4. N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, pages 212–261, 1994.
5. Mozilla Firefox, 2005. http://mozilla.org/products/firefox.
6. Mozilla Firefox Extensions, 2005. http://update.mozilla.org/extensions/.
7. R. Nock and B. Esfandiari. Oracles and assistants : machine learning applied to network supervision. In *Canadian Artificial Intelligence Conference*, number 1418 in Lecture Notes in Computer Science, pages 86–98. Springer-Verlag, 1998.
8. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk email. In *AAAI Workshop on Learning for Text Categorization*. AAAI Press, 1998.

A Bi-clustering Framework for Categorical Data

R gge . G. Pe . a[1], Ce ı e R. ba de [2], a d Jea -F a c ı B ıca [1]

[1] INSA Lyon, LIRIS CNRS UMR 5205,
F-69621 Villeurbanne cedex, France
[2] INSA Lyon, PRISMa EA INSA-UCBL 2058,
F-69621 Villeurbanne cedex, France
{Ruggero.Pensa, Celine.Robardet, Jean-Francois.Boulicaut}@insa-lyon.fr

Abstract. Bi-clustering is a promising conceptual clustering approach. Within categorical data, it provides a collection of (possibly overlapping) bi-clusters, i.e., linked clusters for both objects and attribute-value pairs. We propose a generic framework for bi-clustering which enables to compute a bi-partition from collections of local patterns which capture locally strong associations between objects and properties. To validate this framework, we have studied in details the instance CDK-Means. It is a K-Means-like clustering on collections of formal concepts, i.e., connected closed sets on both dimensions. It enables to build bi-partitions with a user control on overlapping between bi-clusters. We provide an experimental validation on many benchmark datasets and discuss the interestingness of the computed bi-partitions.

1 Introduction

Ma da a ı ı g ech ı e ha e bee de ıg ed . . . . . edge dı c - e f . . ca eg ıca da a hıch ca be e e ed a B . ea . a ıce : he . . de . e b ec a d he c . . . de . e B . ea a ıb e ha e ab e ec d b ec . . e ıe a a ıb e- a e aı . . F ı a ce, gı e **r** ı Tab e 1, e a ha b ec t_2 a ı e . . e ıe g_2 a d g_5. C e ı g ı . . e f he a . da a ı ı g a a d ı ha bee . dıed e e ı e , ı c dı g f he ecıa ca e f ca eg ıca . B . ea da a. I aı g a ı ıde ıf a a ı ı . f b ec a d/ . e ıe ch ha a b ec ı e f c ı hıch ecı e ı a ı ı - ı ı ed [1]. Tha . . ca ea ch ı ı a ı . , a e cıe a g ı h ca . ıde g d a ı ı . b e f . he ac f e ıcı c e cha ac e ı a- ı . . I ha . ı a ed he e ea ch . c ce a c e ı g [2] a d bı-c e ı g [3,4,5] h e g a ı c e bı-c e , ı.e., a cıa ı f (ıb e - a ı g) e f b ec ı h e f e ıe . A e a e f a ı e e ı g bı- a ı ı . ı **r** (Tab e 1) ı $\{\{\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\}\}, \{\{t_2, t_5, t_6, t_7\}, \{g_2, g_5\}\}\}$. The . bı-c e ı dıca e ha $\{t_1, t_3, t_4\}$ a a a ha e e ıe $\{g_1, g_3, g_4\}$. A a . . b e ı ha . . f he bı-c e ı g a g ı h c - e . . e a ı g bı- a ı ı . , hı e ı a a ıca ı d aı . , ı a e e e ha e b ec a d e ıe be . gı g . . e ha . e bı-c e . I . ı a e . e e ea ch . he c a ı . f e e a c ec ı . . f . ıb

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 643–650, 2005.

Table 1. A Boolean context **r**

	g_1	g_2	g_3	g_4	g_5
t_1	1	0	1	1	0
t_2	0	1	0	0	1
t_3	1	0	1	1	0
t_4	0	0	1	1	0
t_5	1	1	0	0	1
t_6	0	1	0	0	1
t_7	0	0	0	0	1

[illegible] [6] [illegible] I [illegible] (T, G) [illegible] T [illegible] G [illegible] e.g., $(\{t_1, t_3\}, \{g_1, g_3, g_4\})$ [illegible] **r**. [illegible] $\{\{t_1, t_3, t_4\}, \{g_1, g_3, g_4\}\}$ [illegible] $((t_4, g_1) \notin \mathbf{r})$ [illegible] $\{\{t_1, t_3\}, \{g_1, g_3, g_4\}\}$ [illegible] $\{\{t_1, t_3, t_4\}, \{g_3, g_4\}\}$ [illegible]

The [illegible] CDK-MEANS [illegible] K-MEANS- [illegible] (f [illegible] CDK-MEANS [illegible] Sec [illegible] 2, [illegible] Sec [illegible] 3 [illegible] CDK-MEANS, [illegible] (COCLUSTER [4] [illegible] BI-CLUST [3]), [illegible] (K-MEANS [illegible] EM [1]) [illegible] Sec [illegible] 4 [illegible]

2 Clustering Model

A [illegible] $\mathcal{O} = \{t_1, \ldots, t_m\}$ [illegible] $\mathcal{P} = \{g_1, \ldots, g_n\}$. The B [illegible] **r** $\subseteq \mathcal{O} \times \mathcal{P}$, [illegible] $r_{ij} = 1$ [illegible] g_j [illegible] t_i. We [illegible] $\{C_1^o \ldots C_K^o\}$) [illegible] $\{C_1^p \ldots C_K^p\}$) [illegible]

be 1 . a 1a ed a e . . f . a c . ce . . F . a , a bı- e 1 a e e e $b_j = (T_j, G_j)$ $(T_j \subseteq \mathcal{O}, G_j \subseteq \mathcal{P})$ a d e a . e ha a c ec 1 . . f a , 1 , 1 1 e e 1 g bı- e . de . ed $\mathcal{B}$ ha bee e , ac ed f . . $\mathbf{r}$ bef , eha d. Le . . . de c ıbe b_j b he B . ea ec . , $< \mathbf{t}_j >, < \mathbf{g}_j > = < t_{j1}, \ldots, t_{jm} >, < g_{j1}, \ldots, g_{jn} >$ he e $t_{jk} = 1$ ıf $t_k \in T_j$ (0 . he 1 e) a d $g_{jk} = 1$ ıf $g_k \in G_j$ (0 . he 1 e). We a e . 1 g f . K c . e . f bı- e . $\{C_1, \ldots, C_K\}$ $(C_i \subseteq \mathcal{B})$. Le . de e he ce . , ıd . f a c . e . f bı- e . C_i a $\mu_i = < \tau_i >, < \gamma_i > = < \tau_{i1}, \ldots, \tau_{im} >, < \gamma_{i1}, \ldots, \gamma_{in} >$ he e τ a d γ a e he . a ce . , ıd c . . . e . :

$$\tau_{ik} = \frac{1}{|C_i|} \sum_{b_j \in C_i} t_{jk}, \quad \gamma_{ik} = \frac{1}{|C_i|} \sum_{b_j \in C_i} g_{jk}$$

We . . de . e . , dı a ce be ee a bı- e a d a ce . , ıd:

$$d(b_j, \mu_i) = \frac{1}{2} \left(\frac{|\mathbf{t}_j \cup \boldsymbol{\tau}_i| - |\mathbf{t}_j \cap \boldsymbol{\tau}_i|}{|\mathbf{t}_j \cup \boldsymbol{\tau}_i|} + \frac{|\mathbf{g}_j \cup \boldsymbol{\gamma}_i| - |\mathbf{g}_j \cap \boldsymbol{\gamma}_i|}{|\mathbf{g}_j \cup \boldsymbol{\gamma}_i|} \right)$$

I 1 he ea . f he eıgh ed . . e , ıca dı e e ce . f he e c . . . e . . We a . . e $|\mathbf{t}_j \cap \boldsymbol{\tau}_i| = \sum_{k=1}^{m} a_k \frac{t_{jk} + \tau_{ik}}{2}$ a d $|\mathbf{t}_j \cup \boldsymbol{\tau}_i| = \sum_{k=1}^{m} \frac{t_{jk} + \tau_{ik}}{2}$ he e $a_k = 1$ ıf $t_{jk} \cdot \tau_{ik} \neq 0$, 0 . he 1 e. I 1 1 e , he 1 e ec 1 . 1 e a . he ea be ee he . be . f c . . . . b ec a d he . . f hei ce . , ıd eıgh . . The . 1 . 1 he ea be ee he . be . f . b ec a d he . . . f hei ce . , ıd eıgh . . The e ea , e a e de . ed 1 1 a . . . . e 1e .

Ob ec . t_j (e . . . e 1e g_j) a e a . 1g ed . . . e . f he K c . e . (de- . . ed i) f . , hıch τ_{ij} (e . . γ_{ij}) 1 . a 1 . . We ca e ab e ha a . . be . f . b ec a d/ . , . . e 1e be . g . . . , e ha . . e c . e . b c . . . 1 g he 1 e . f he . e . a 1 g a . . f each c . e . . Tha . . . . de . 1 1 . . f c . e . e be . hı de e . 1 ed b he a e . f $\boldsymbol{\tau}_i$ a d $\boldsymbol{\gamma}_i$, e . . eed . ada he c . e . a ıg . e . . e . F . , hı . . . e, e . 1 . . d ce a a e e . δ_o a d δ_p 1 [0,1] . a ıf he e be . hı . f each e e e . . a c . e . . We a ha a . . b ec t_j be . . g . a c . e . C_i^o ıf $\tau_{ij} \geq (1 - \delta_o) \cdot max_i(\tau_{ij})$. A a . g . . , a . . e . g_j be . g . a c . e . C_i^p ıf $\gamma_{ij} \geq (1 - \delta_p) \cdot max_i(\gamma_{ij})$. Ob 1 . . he . . be . f . e . a 1 g . b ec (e . ., . . e 1e) de e d . . he dı , ıb 1 . . f he a e . f $\boldsymbol{\tau}_i$ (e . . $\boldsymbol{\gamma}_i$). N . ıce ha ıf . e . a 1 g 1 a . ed, $\delta = 0$ d e . . 1 ha each . b ec . . . e . 1 a . ıg ed . a . 1 g e c . e . . The ch ıce . f a , e e a . a e f . δ 1 c ea . a ıca 1 . -de e de . . Whe a bı-c . e . 1 g . c , e h d 1 he da a, 1 e a e . f δ a e . . e . gh . . . ıde , e e a . e a 1 g. O a . he . ha d, 1 . . 1 c . e . , e e 1 e a e . f δ ca g1 e , 1 e . . ıg 1 ca . . e . a 1 g . . e .

We ca . . . . ıde de a1 . ab . he . dıed 1 . a ce . f hı f a e . . : a bı-c . e . 1 g ba ed . . f . . a c . ce . . Ma . e cıe . a g , 1 h . ha e bee de e . ed ha ca e , ac c . . e e c . ec 1 . . . f f . . a c . ce . . de . c . - . , a1 . . We . e D-Miner [7].

O . 1 . a ce CDK-Means 1 . e e ed 1 Tab e 2. I c . . e a bı- a . 1 1 . . f a da a e $\mathbf{r}$ g1 e . a c . ec 1 . . f bı- e . $\mathcal{B}$ e , ac ed f . . $\mathbf{r}$ bef , eha d (e.g., f . . a c . ce .), he de 1 ed . . be . f c . e . K, he h e h d a e f . , δ_o

Table 2. CDK-MEANS pseudo-code

CDK-MEANS ($\mathbf{r}$ is a Boolean context, $\mathcal{B}$ is a collection of bi-sets in $\mathbf{r}$, K is the number of clusters, MI is the maximal iteration number, δ_o and δ_p are thresholds values for controlling overlapping)

1. Let $\mu_1 \dots \mu_K$ be the initial cluster centroids. $k := 0$.
2. Repeat
   (a) For each bi-set $c \in \mathcal{B}$, assign it to cluster C s.t. $d(c, \mu_i)$ is minimal.
   (b) For each cluster C_i, compute $\boldsymbol{\tau_i}$ and $\boldsymbol{\gamma_i}$.
   (c) $k := k + 1$.
3. Until centroids are unchanged or $k = MI$.
4. If overlap is allowed, for each $t_j \in \mathcal{O}$ (resp. $g_j \in \mathcal{P}$), assign it to each cluster C_i^o (resp. C_i^p) s.t. $\tau_{ij} \geq (1 - \delta_o) \cdot max_i(\tau_{ij})$ (resp. $\gamma_{ij} \geq (1 - \delta_p) \cdot max_i(\gamma_{ij})$).
5. Else, for each $t_j \in \mathcal{O}$ (resp. $g_i \in \mathcal{P}$), assign it to the first cluster C_i^o (resp. C_i^p) s.t. τ_{ij} (resp. γ_{ij}) is max.
6. Return $\{C_1^o \dots C_K^o\}$ and $\{C_1^p \dots C_K^p\}$

a d δ_p, a d a a 1 be f 1 e a 1 MI. O e a e **r**, CDK-MEANS 1de he bı- a 1 1 gı e 1 Sec 1 1. The c e 1 1 1 ea 1 $\mathcal{B}$ a d ca abı 1 1 e a e dı c ed 1 Sec 1 3.

Related work. [3] a d [4] bı-c e 1 g e h d a e a 1 e e a a 1 1 he he he e 1 ed, 1 1 1 g e ec 1 e he G d a -K a ' τ c e cıe a d he 1 a 1 f a 1 . The 1 e e 1 g dı e e ce 1 1 ead f c ıde 1 g b ec a d e ıe a e a a ed e 1 ıe d 1 g he bı-c e 1 g a (e e ıf b ec 1 e f c 1 a e c ed b h e), CDK-MEANS c ıde hei a cıa 1 a he e e e ce . The ec d e 1 ha CDK-MEANS ca ea 1 c e a 1 1 1 h e a 1 g c e .

3 Experimental Validation

Dı e e ech 1 e ca be ed e a a e he a 1 f a a 1 1 . A e e- a c 1 e 1 c 1 1 c a 1 g he c ed a 1 1 1 h a c ec e. I ea ha da a 1 a ce a e a ead a cıa ed e c ec abe a d ha e a 1 e he ag ee e be ee c ed abe a d c ec e . A a ea e 1 he Jacca d c e cıe [1].

T e a a e he a 1 f bı-c e 1 g 1 g a 1 e a c 1 e 1 e e G d a a d K a ' τ c e cıe [8]. I 1 e a a ed 1 a c - cc e ce ab e **p** a d 1 dı c 1 1 a e e bı- a 1 1 . . he 1 e 1 f he f c 1 a 1 be ee b h a 1 1 [3]. p_{ij} 1 he f e e c f e a 1 be ee a b ec f a c e C_i^o a d a e f a c e C_j^p. $p_{i.} = \sum_j p_{ij}$ a d $p_{.j} = \sum_i p_{ij}$. The $\tau_{\mathcal{Q}}$ c e cıe e a a e he 1 a ed c 1 1 e gı e b he edge f C^o he edıc 1 f C^p ($\tau_{\mathcal{O}}$ 1 de e he ea e he e cha gı g he a 1 1):

$$\tau_Q = \frac{\sum_i \sum_j \frac{p_{ij}^2}{p_{i.}} - \sum_j p_{.j}^2}{1 - \sum_j p_{.j}^2}$$

We report on experiments using eight well-known datasets taken from the UCI ML Repository[1] and from the JSE Data Archive[2]. All the experiments have been performed on a PC with 1 Gb RAM and a 3.0 GHz P4 processor. First, without considering the class variable, we have processed each dataset with D-Miner [7]. Minimal set size constraints have been used for mushroom and credit-a (minimal item set and object set size (13,15) and (6,15)) to obtain complete collections of formal concepts before using CDK-Means.

Table 3. Goodman-Kruskal's coefficient values for different bi-clustering algorithms (mr-2 and mr-5 refer to mushroom with 2 and 5 clusters)

		Bi-Clust	Cocluster		CDK-means	
Dataset	Dim.	Max	Max	Mean	Max	Mean
voting	435×48	**0.320**	**0.320**	0.315±0.002	0.311	0.311±0.000
titanic	2201×8	**0.332**	0.321	0.226±0.076	0.314	0.160±0.109
iris-2	150×8	0.543	0.543	0.357±0.195	0.543	0.474±0.056
iris-3	150×8	**0.544**	0.390	0.379±0.045	0.523	0.329±0.080
zoo-2	101×16	0.191	0.186	0.157±0.034	**0.192**	0.165±0.020
zoo-7	101×16	-	0.080	0.065±0.009	**0.083**	0.049±0.015
breast-w	699×18	**0.507**	**0.507**	0.474±0.121	0.498	0.498±0.000
credit-3	690×52	0.104	0.014	0.003±0.003	**0.110**	0.091±0.015
credit-2	690×52	-	0.012	0.006±0.004	**0.096**	0.055±0.011
mr-2	8124×126	-	**0.198**	0.158±0.026	0.176	0.157±0.017
mr-5	8124×126	**0.187**	0.119	0.097±0.009	0.116	0.112±0.004
ads	3279×1555	-	0.006	0.003±0.001	**0.538**	0.137±0.109

We compared CDK-Means bi-partitions with those obtained by Cocluster [4], and Bi-Clust [3]. As the initialization of these algorithms is randomized, we executed them 100 times on each dataset and we selected the results which returned the best Goodman-Kruskal's coefficients. The number of desired clusters for each experiment has been set to the number of class variable values, except for Bi-Clust which automatically determines the number of clusters. Bi-Clust is available within WEKA[3] and we were not able to process internet-ads (more than 1500 properties). We summarize the results in Table 3. We provide only the τ_Q coefficients. The corresponding τ_O coefficients are [illegible] also not significantly different. Notice that, when CDK-Means has the worst result, the Goodman-Kruskal's coefficient is not significantly dissimilar from other algorithms coefficients. On the other hand, for internet-ads, the coefficient obtained with CDK-Means is considerably higher than the one obtained with Cocluster. This is due to the high dimension of the dataset which is not well handled by the other algorithms. Also the average behavior is similar to the one of Cocluster. The average values of the two algorithms are often similar, as we

---

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html

[2] http://www.amstat.org/publications/ jse/jse_data_archive.html

[3] http://www.cs.waikato.ac.nz/ml/weka/

Table 4. Jaccard coefficient values w.r.t. class variable for different algorithms

Dataset	Bi-Clust	Cocluster	K-Means	EM	CDK-means
voting	0.6473	0.6473	0.6027	0.6459	**0.6737**
titanic	0.4281	0.4651	0.3697	0.3697	**0.4745**
iris-2	0.4992	0.4992	**0.5117**	0.4992	0.4992
iris-3	0.4932	0.5240	**0.5394**	**0.5394**	0.5144
zoo-2	0.5141	**0.5630**	0.5027	0.5179	0.5141
zoo-7	-	0.1647	0.1843	**0.2325**	0.2212
breast-w	0.8246	0.8287	0.7777	**0.8328**	0.7666
credit-3	0.4233	0.3869	0.3765	0.3405	**0.4452**
credit-2	-	0.4360	0.4698	0.4442	**0.4915**
mr-2	-	0.6819	0.3496	**0.6976**	0.6356
mr-5	**0.5068**	0.3450	0.3192	0.3364	0.3375
ads	-	0.4317	-	-	**0.8019**

a he a da d de ıa ı a e . N ıce ha f voting-records a d breast-w, CDK-Means ha a a d ced he a e bı- a ı ı .

CDK-Means ge e a eed f e e ec ı ı e ha he he a- g ı h beca e ı ce e ıb a ge c ec ı f f a c ce . I he e be ch a , he e ac ı f f a c ce b ı e f ı ha e e- ı e (f 1 20 ec d). U ı g ı ı a ı e c a ı d ı g he f a c ce e ac ı ha e e ab e ed ce he c ec ı ı e a d ı ı be dı- c ed a e . F titanic, iris, a d zoo, CDK-Means e f ı e ha e ec d, hı e f breast-w, credit-a a d internet-ads, he a e age e ec ı ı e ı e ha e ı e. F mushroom, he a e age e ec ı ı e ı ab e e ı e ı ce e ha 50 000 f a c ce ha e be ce ed.

We a ed he Jacca d ı de c a e he ag ee e f he b ec a- ı ı ı h h e de e ı ed b he c a a ıab e . He e agaı , e ha e e ec ed he bı- a ı ı ı h he hıghe τ_Q c e cıe [4]. We ıde he c a ı ı ı Tab e 4. Agaı , a g ı h ı c e ı ı e . he he bı-c e ı g e h- d . Wı h he e ce ı f breast-w, a g ı h a a e f a be e ha Bi-Clust, a d f he ı e be e ha Cocluster.

Fı a , e ha e c a ed e , c a ıca c e ı g a g- ı h , he WEKA ı e e a ı f K-Means a d EM (ee Tab e 4). E ce f breast-w, a g ı h ı c e ı ı e . he he e . F da a e , CDK-Means e f be e ha K-Means a d EM. O ce agaı , he a g ı h b aı he be e , he dı e e ce ı h he c e f he he ı ıg ı ca (e ce breast-w). The e e h ha c e ı g f f a c ce ı a e e a a ach f b h a ı ı ı g a d bı- a ı ı ı g a .

Scalability Issues. C ec ı f f a c ce a e a h ge, e ecıa ı ı ı ıca ı da a. S ı ce CDK-Means ha a ı ea c e ı ı he be f bı- e , ı ca be ı e-c ı g. A b ı ı ı e ec e f a c ce , f ı a ce he e hıch ı e e gh b ec a d/ e ıe . I e e ı g , ch ı ı a ı e c a ı ca be hed ı f - a c ce ı ı g a g ı h [7]. N ı e ab e he e ac ı ı ha d

[4] Clearly, it does not lead to the highest Jaccard's index.

Table 5. Clustering results on ads-internet with different minimal size constraints

(σ_p,σ_o)	$\lvert\mathcal{B}\rvert$	time(s)	τ(mean)	τ(max)	J-class	J-ref
(0,0)	7682	33	0.137 ± 0.109	0.538	0.8019	1
(4,4)	2926	8	0.194 ± 0.137	0.565	0.6763	0.6737
(5,5)	2075	5	0.254 ± 0.148	0.565	0.6862	0.7490
(5,10)	1166	2.5	0.223 ± 0.119	0.511	0.6745	0.7405
(7,10)	873	2	0.204 ± 0.095	0.549	0.6172	0.6658
(10,10)	586	1.5	0.227 ± 0.125	0.543	0.6080	0.7167

c . e . , b a . . , 1 1 1 e , 1 , e . . e f . . a c . . ce . hıch . ıgh be d e . . . 1 e. We he ef . e g e . ha hı ca 1 c ea e he a 1 . f he c . e 1 g . e . Le σ_o be he . 1 1 a . 1 e . f he . b ec . e a d σ_p be he . 1 1 a . 1 e . f he . . e . e . P . e ıe (e . . b ec .) ha a e 1 . e a 1 . 1 h e . ha σ_o . b ec . (e . . σ_p . . e ıe) 1 . . be 1 c ded 1 a f . . a c . ce . A . . bı- a 1 1 . 1 g . e h d 1 ba ed . . . a . . - . . ce 1 g . f he e a e . . , he e . b ec . a d/ . . . e ıe ca . . be 1 c ded 1 he . a bı- a 1 1 . . Thı 1 . . . ece a 1 a . . b e ıf e . efe a be e . . b . . e . . . 1 e. H - e e , . . e ca be 1 e e ed 1 . . dı g a bı- a 1 1 . ha 1 c de a . b ec . a d . . e ıe . A . b 1 . . . 1 . 1 . add he . a d b . . f . . a c . ce . ($\mathcal{O}, \emptyset$) a d ($\emptyset, \mathcal{P}$). Thı ha bee d . e 1 . . . e e e 1 e . (mushroom, credit-a) a d e . . ıced ha he dec ea e . f he Jacca d a d G . d a -K . a ' c . - e cıe . e e . . ıg 1 ca . . We . ade f . he e e 1 e . . . de . a d he 1 ac f . 1 g . 1 1 a 1 e c . . . a1 . . . b h he e ec 1 . 1 e a d he a - 1 . f he c . . ed bı- a 1 1 . . We ha e c . . ıde ed internet-ads a he . . . 1 ab e f . he e e e 1 e . (hıgh ca d1 a 1 f . . b h . b ec a d . . e . e .). We e . ac ed f . . a c . ce . b . e 1 g . . . e c . bı a 1 . . . f c . . . a1 . ($0 \leq \sigma_p < 10$ a d $0 \leq \sigma_o < 10$) a d b add1 g ($\mathcal{O}, \emptyset$) a d ($\emptyset, \mathcal{P}$).The . e . a e . . . a 1 ed 1 Tab e 5. I . h . . ha , 1 c ea 1 g he . 1 1 a . 1 e h e h- . d c . . ıde ab . ed ce he . . be . f e . ac ed f . . a c . ce . a d h . he a e age e ec 1 . 1 e. A . . he e . ac 1 . 1 e dec ea e f . . 4 . ec . d (f . $\sigma_p = \sigma_o = 0$) . . . e . ec . d (f . $\sigma_p = \sigma_o = 10$). M . e . e , he . a 1 . G . d a -K . a ' c . e cıe . d e . . cha ge . ıg 1 ca . . I . . . e ca e 1 1 g ea e . ha he c . e cıe . c . . ed he . . . 1 e c . . . a1 . 1 . ed. A . . he a e age a e . f he G . d a -K . a ' . ea . e a e be e 1 ge e a (hı e . a da d de ıa 1 . . a e a e 1 1 a). We he . c . . ed he Jacca d 1 de . f he d1 e e . a 1 1 . . . . . he c a . . a ıab e (J-c a . c . . .) a d he a 1 1 . . b a1 ed 1 h . . e 1 g a . c . . . a1 . (J- ef c . . .). The . ıgh . a ıabı 1 . f he Jacca d 1 de e a d he hıgh a e . f he τ . ea . e . h . ha he a e . 1 c . . 1 e . . . . he c a . . . e. F1 a , . e . a e a a . be e . ha h . e . b a1 ed b . 1 g Cocluster (ee Fıg. 3 a d Fıg. 4) h . e a e age e ec 1 . 1 e 1 ab . 4.2 . ec . d . I . he e . . , 1 c ea 1 g σ_p a d σ_o ca e 1 1 a e he 1 ac f . 1 e d e . . a e . b- a ıce . I a ıc a , g . . 1 g a ge f . . a c . ce . ca 1 . . . e he e e a c . f bı- a 1 1 . . . N . ıce ha ıf e d . . add ($\mathcal{O}, \emptyset$) a d ($\emptyset, \mathcal{P}$), e ge be e . e . 1 . . 1 g a . b e . f he . ıgı a a . 1 : c . . . a1 . ca be . ıgge ed . . ade- . be ee he c . e age . f he bı- a 1 1 . a d he a 1 . f he . e .

4 Conclusion and Future Work

We ha e 1 d ced a e bı-c e 1 g f a e hıch e 1 ca a e 1 he da a he c 1 g a c ec 1 f (ıb e a 1 g) bı-c e . The 1 a ce CDK-Means b 1 d 1 a e a a 1 1 b ec a d a a 1 1 e ıe b a 1 g a K-Means- 1 e a g 1 h a c ec 1 f e ac ed f a c ce . O e e 1 e a a ıda 1 ha c ed he added- a e f CDK-Means . . . he (bı-)c e 1 g a g 1 h . We de - a ed ha ch a f ca a e a e e a g ba a e a ach ca . D e he ac f ace, e 1 ed he e e 1 e a e ea da a a d he d f e a 1 g c e [9]. Ma he 1 a ce f he f a e- ıgh be dıed. F 1 a ce, gı e e ac ed ca a e , a e a ı e c e 1 g ech 1 e ca be c ıde ed. A , he 1 d f ca a e (ı.e., e e a bı- e hıch a e f a c ce) c d be c ıde ed. Fı a , a e cı 1 g cha e ge c ce c a ı -ba ed c e 1 g. O f a e gı e 1 e 1 ıe f hı g c a ı a d ı e e e e , ı.e., d 1 g ca a e 1 1 g b a he b 1 dı g bı- a 1 1 f he .

Acknowledgements. The a h a ha L ıgı Ma e 1 1 f hı ech- ıca . Thı e ea ch 1 a ıa f ded b CNRS (ACI MD 46 Bı g).

References

1. Jain, A., Dubes, R.: Algorithms for clustering data. Prentice Hall, Englewood cliffs, New Jersey (1988)
2. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Machine Learning **2** (1987) 139–172
3. Robardet, C., Feschet, F.: Efficient local search in conceptual clustering. In: Proceedings DS'01. Number 2226 in LNCS, Springer-Verlag (2001) 323–335
4. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings ACM SIGKDD 2003, Washington, USA, ACM Press (2003) 89–98
5. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinf. **1** (2004) 24–45
6. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In Rival, I., ed.: Ordered sets. Reidel (1982) 445–470
7. Besson, J., Robardet, C., Boulicaut, J.F., Rome, S.: Constraint-based concept mining and its application to microarray data analysis. Intelligent Data Analysis **9(1)** (2005) 59–82
8. Goodman, L.A., Kruskal, W.H.: Measures of association for cross classification. Journal of the American Statistical Association **49** (1954) 732–764
9. Pensa, R.G., Robardet, C., Boulicaut, J.F.: Using locally relevant bi-sets for categorical data conceptual clustering. Research report, LIRIS CNRS UMR 5205 - INSA Lyon, Villeurbanne, France (2005) Submitted to a journal.

Privacy-Preserving Collaborative Filtering on Vertically Partitioned Data*

Huse ı P. at a d We ıa g Du

Department of Electrical Engineering and Computer Science,
Syracuse University, CST 3-114, Syracuse, NY 13244-1240, USA
{hpolat, wedu}@ecs.syr.edu

Abstract. Collaborative filtering (CF) systems are widely used by E-commerce sites to provide predictions using existing databases comprised of ratings recorded from groups of people evaluating various items, sometimes, however, such systems' ratings are split among different parties. To provide better filtering services, such parties may wish to share their data. However, due to privacy concerns, data owners do not want to disclose data. This paper presents a privacy-preserving protocol for CF grounded on vertically partitioned data. We conducted various experiments to evaluate the overall performance of our scheme.

1 Introduction

C ab ratı e fi terı g (CF) ıs a rece t tech ıque that he ps users c pe ıth ı f r-matı er ad usı g ther users' prefere ces. It ıs ıde used b E-c mmerce, dırect rec mme datı s, a d search e gı es [1,2]. The g a ıs t predıct h e a user ( ) ı ıke a ıtem that he/she dıd t bu bef re based ther users' prefere ces [4].

Data c ected f r CF purp ses mıght be ertıca partıtı ed bet ee dı er-e t partıes here the partıes h d dıs ı t sets f ıtems' ratı gs c ected fr m the same users. A ı dı ıdua 's prefere ces f r pr ducts mıght be sp ıt am g dı er-e t E-c mmerce c mpa ıes such as Amaz .c m a d M ıeFı der.c m. O ı e e d rs ca pr duce better referra s ıf the share ı f rmatı ab ut theır cus-t mers ıth ther e d rs. J ı t data ıs be eficıa f r E-c mmerce sıtes because cust mers prefer retur ı g t st res ıth better referra s a d the search f r m re pr ducts t purchase. Shared ı f rmatı ı a s be efit cust mers b makı g ıt m re ıke t recei e m re accurate a d re ıab e rec mme datı s. C mbı ı g ertıca partıtı ed data (VPD) ıs he pfu he CF s stems ha e ımıted rated ıtems. T fi d m re re ıab e matchı gs a d pr ıde m re accu-rate referra s, the er ap bet ee users sh u d be arge e ugh; thıs mıght be achıe ed b ı tegratı g VPD. H e er, due t prı ac c cer s, data ers d t a t t c ab rate a d dısc se theır data t each ther.

* This work was supported by Grants ISS-0219560 and ISS-0312366 from the United States National Science Foundation.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 651–658, 2005.

VPD-based CF ıs esse tıa a d ca be achıe ed ıf prı ac measures are ı tr duced t data . ers. We stud the prı ac -preser ı g c ab ratı e fi - terı g (PPCF) . VPD pr b em: [illegible] *(A B)* [illegible]

We pr p se a pr t c t achıe e PPCF . VPD. Prı ac , accurac , a d efficıe c are c ıctı g g a s. Theref re, the pr p sed pr t c sh u d achıe e a g d ba a ce bet ee them. Our scheme c sısts f - ı e a d ı e c m- putatı c mp e ts. We c duct s me c mputatı s - ı e t achıe e data e cha ge bet ee partıes ıth prı ac . Durı g the ı e c mputatı , a e cust mer (a actı e user a) c mmu ıcates ıth b th partıes. The the perf rm data e cha ge thr ugh a ıth prı ac . The c mpa that d es t h d ratı gs f the target ıtem (the ıtem that a ıs kı g f r a predıctı) fi ds predıctı a d te s a. Sı ce data e cha ges are required he e er a e cust mer a ts a predıctı a d eıther part ca act as a actı e user ı mu tıp e sce arı s t ear ab ut ther part 's data, the pr p sed pr t c sh u d be secure agaı st such attacks c mı g fr m b th partıes.

Ca pr p ses t schemes f r PPCF [1,2]. A c mmu ıt f users ca c m- pute pers a ızed rec mme datı s ıth ut e p sı g ı dı ıdua data usı g such schemes. P at a d Du use ra d mızed perturbatı tech ıques f r PPCF [6,7]. Vaıd a a d C ıft [8,9,10] prese t prı ac -preser ı g meth ds f r ass cıatı ru e mı ı g, a e Ba es c assıfier, a d K-mea s c usterı g based VPD. We used the CF a g rıthm pr p sed b [4]. If v_{ij} ıs user i's te ıtem j, a d $\overline{v_i}$ a d σ_i are the mea te a d the sta dard de ıatı f the user i's ratı gs, re- spectı e , the the z-sc res (z_{ij}) ca be defi ed as $z_{ij} = (v_{ij} - \overline{v_i})/\sigma_i$. Her cker et. a fi d predıctı s as f s here n ıs the umber f users:

$$p_{aq} = \overline{v_a} + \sigma_a \cdot \frac{\sum_{i=1}^{n} w_{ai} \cdot z_{iq}}{\sum_{i=1}^{n} w_{ai}} \qquad w_{ai} = \sum_{k} z_{ak} \cdot z_{ik} \tag{1}$$

here k ıs the ıtem set b th a a d the user i ha e rated a d q ıs the target ıtem. σ_a a d σ_i are sta dard de ıatı s f a's ratı gs a d i's ratı gs, respectı e . p_{aq} ıs the predıctı f r a q a d w_{ai} ıs sımı arıt bet ee a a d i. We used h m m rphıc pr pert f r ur pr p sed pr t c : $E_k(x) * E_k(y) = E_k(x + y)$. Ma such s stems e ıst, a d a e amp e ıs the s stem b Paı ıer [5]. A usefu pr pert f h m m rphıc e cr ptı schemes ıs that a addıtı peratı ca be c ducted based the e cr pted data ıth ut decr ptı g them.

2 PPCF on VPD

Wıth ut prı ac as a c cer , data ers e cha ge theır data t pr ıde CF ser ıces. H e er, ıth prı ac as a c cer , the c mpa ıes sh u d t be ab e

t. ear. each . ther's data. We ca. rıte Eq. (1) as $p_{aq} = \overline{v_a} + \sigma_a \cdot P$ here P ca. be defi. ed as f. . s:

$$P = \frac{\sum_{i=1}^{n} \left[\sum_{k} z_{ak} z_{ik} \right] z_{iq}}{\sum_{i=1}^{n} \sum_{k} z_{ak} z_{ik}} = \frac{\sum_{k} z_{ak} \left[\sum_{i=1}^{n} z_{ik} z_{iq} \right]}{\sum_{k} z_{ak} \left[\sum_{i=1}^{n} z_{ik} \right]} \tag{2}$$

CF s stems ca. te hether a ı ıke q . r . . t, rather tha. te ı g h much he/she ı ıke ıt. T. d. thıs, p_{aq} ıs c. mpared ıth a thresh. d (τ). If $p_{aq} \geq \tau$, q ıs rec. mme. ded as ıke, . ther ıse ıt ıs rec. mme. ded as dıs ıke. If the ratı. gs . ar fr. m 1 t. 5, τ ıs set t. 3.5 hı e ıt ıs set t. 2 ıf the ra. ge fr. m -10 t. 10. Sı. ce A's a. d B's data ıs used t. ca cu ate P, Eq. (2) ca. be rıtte. as:

$$P = \frac{\sum_{k_A} z_{ak_A} \left[\sum_{i=1}^{n} z_{ik_A} z_{iq} \right] + \sum_{k_B} z_{ak_B} \left[\sum_{i=1}^{n} z_{ik_B} z_{iq} \right]}{\sum_{k_A} z_{ak_A} \left[\sum_{i=1}^{n} z_{ik_A} \right] + \sum_{k_B} z_{ak_B} \left[\sum_{i=1}^{n} z_{ik_B} \right]} = \frac{A_N + B_N}{A_D + B_D} \tag{3}$$

here $k = k_A + k_B$, a. d k_A a. d k_B represe. t the ıtem sets b. th a a. d i ha. e rated am. . g the ıtems he d b A a. d B, respectı e .

2.1 Off-Line Computation

The de. . mı at. r part ı Eq. (3) ca. be eası c. mputed because A a. d B ca. fi. d A_D a. d B_D usı. g their . . data. H e. er, the part h. d. es . . t . . q . eeds t. ha. e z_{iq} . a ues f. r a $i = 1, \ldots, n$ t. c. mpute $\sum_{i=1}^{n} z_{ik_j} z_{iq}$. ecessar f. r the . umerat. r. Sı. ce A a. d B f. . the same steps, e . . e p aı. the pr. cedure f. r A. A h. rız. . ta dı. ıdes ıts $n \times m_A$ data matrı ı. t. c_A sub-matrıces here each sub-matrı c. . sısts . f n/c_A users a. d theır ratı. gs f. r ıtems he d b A, here m_A ıs the . umber . f ıtems A . . s. It the. dısguıses data ı. each sub-matrı ı. depe. de. t . F. r $i = 1, \ldots, c_A$, A perf. rms the f. . ı. gs:

Step 1. Permutes m_A c. um. . ect. rs usı. g a permutatı. . fu. ctı. . Π_{Ai}.
Step 2. F. r $j = 1, \ldots, m_A$, dı ıdes the permuted c. um. . ect. r $\Pi_{Ai}(I_{ij})$ ı. t. d_{ij} ra. d. m . ect. rs here $\Pi_{Ai}(I_{ij}) = \sum_{z=1}^{d_{ij}} X_{ijz}$ a. d d_{ij} ıs a. ı. teger ch. se. ıth u. ıf. rm ra. d. m dıstrıbutı. . . er the ra. ge $[1, \beta_A]$.
Step 3. Permutes $X_{i11}, X_{i12}, \ldots, X_{i1d_{i1}}, X_{i21}, X_{i22}, \ldots, X_{i2d_{i2}}, \ldots, X_{im_A1}, X_{im_A2}, \ldots, X_{im_Ad_{im_A}}$ ra. d. m . ect. rs f. u. d ı. step 2 usı. g π_{Ai}.
Step 4. A se. ds D_{Ai} permuted ra. d. m . ect. rs t. B here $D_{Ai} = d_{i1} + d_{i2} + \ldots + d_{im_A}$. B c. mputes the sca ar pr. ducts bet ee. these permuted ra. d. m . ect. rs a. d ıts m_B c. um. . ect. rs usı. g the c. rresp. . dı. g parts . f them a. d fi. ds $D_{Ai}m_B$ sca ar pr. duct resu ts.
Step 5. B e. cr pts the sca ar pr. duct resu ts usı. g a h. m. m. rphıc e. cr pt ı. . scheme a. d ıts pub ıc ke e_b a. d se. ds $D_{Ai}m_B$ e. cr pted . a ues t. A.

Step 6. S ce A k s Π_{Ai} a d π_{Ai} a d h m m rphıc e cr pt ı ıs used, ıt fi ds the sca ar pr duct resu ts f ıts m_A a d B's m_B c um ect rs ı e cr pted f rms usı g h m m rphıc e cr pt ı pr pert . After c ductı g these steps f r a $i = 1, \ldots, c_A$, A gets e cr pted sca ar pr duct resu ts f r ıts a c_A sub-matrıces. S ce A's data ıs h rız ta dı ıded, ıt agaı uses h m m rphıc e cr pt ı pr pert t fi d the fi a sca ar pr duct resu ts ı e cr pted f rms.

A creates a matrı Σ_A c sıstı g f $e_b(\Sigma_{ij})$ f r $i = 1, \ldots, m_A$ a d $j = 1, \ldots, m_B$ here $e_b(\Sigma_{ij})$ represe ts the e cr pted sca ar pr duct bet ee i^{th} c um ect r f B a d the jth c um ect r f A. It ge erates arge e ugh v_{ij} ra d m umbers f r $i = 1, \ldots, m_B$ a d $j = 1, \ldots, m_A$, e cr pts them usı g e_b, a d adds them t the $e_b(\Sigma_{ij})$ a ues usı g h m m rphıc e cr pt ı pr pert . It fi ds matrı Σ'_A c sıstı g f $e_b(\Sigma'_{ij})$ here $\Sigma'_{ij} = \Sigma_{ij} + v_{ij}$ a d st res v_{ij} a ues ı a matrı V_A. It se ds Σ'_A t B that decr pts the e cr pted a ues a d fi ds the matrı Σ''_A c sıstı g f Σ'_{ij} a ues a d st res ıt. B f ı g the same pr cedure, B fi ds matrıces Σ_B, Σ'_B, a d V_B. B st res V_B a d A fi ds Σ''_B, a d the st res ıt. A a d B c mpute the ıtem mea tes a d st re them ı $m_A \times 1$ a d $m_B \times 1$ matrıces, respectı e .

2.2 Online Computation

S ce eıther part ca act as a actı e user ı mu tıp e sce arı s, ı e c mp - e t sh u d be secure agaı st such attacks. The steps are as f s:

Step 1. a se ds hıs/her data a d a quer t the c mpa that s q. Assume that B s q. B c mputes B_N a d B_D. H e er, s ce A ca act as a actı e user ı mu tıp e sce arı s t ear them, B uses prı ate B_N & B_D c mputatı pr t c , hıch ıs e p aı ed ı the f ı g, t c mpute them.
Step 2. B ca c mpute A'_N a ue usı g the data fr m the qth r f the matrı Σ''_A a d a's c rresp dı g data here $A'_N = A_N + R_q$. The data fr m the qth r f the matrı Σ''_A represe ts $\sum_{i=1}^{n} z_{ik_A} z_{iq}$ a ues dısguised b v_{qk_A} ra d m umbers f r a k_A. A ca c mpute $R_q = \sum_{k_A} z_{ak_A} v_{qk_A}$ here k_A represe ts the ıtems rated b a am g the ıtems he d b A.
Step 3. B c mputes $A_N + R_q + B'_N$ a d B'_D a d se ds them t gether ıth a's e mea te, sta dard de ıatı , a d the z-sc res f r th se ıtems rated b a am g ıtems he d b A t A thr ugh a. A c mputes R_q, fi ds $A_N + B'_N = A'_N + B'_N - R_q$ a d A_D, a d estımates P' usı g Eq. 3 based the quer .

A c mputes p'_{aq}, te s a hether he/she ı ıke q r t b c mparı g p'_{aq} ıth τ. S ce B ca act as a actı e user, A uses a ra d m thresh d t pre e t B fr m ear ı g A_D a d A_N. It ge erates a u ıf rm ra d m umber $(r_{A\tau})$ fr m a ra ge $[-\alpha_A, \alpha_A]$, fi ds $\tau + r_{A\tau}$, a d uses ıt as a ra d m thresh d.

Our scheme ca be e te ded t mu tı-part . Each e d r e cha ges data - ı e ıth thers a d st res ıt as ı t -part scheme. Durı g ı e phase, a se ds hıs/her data t the part that s q. That part c mputes the required data ıke ıt d es ı t -part scheme a d se ds resu ts t a. The e part acts as a master sıte. Other partıes c mputes the a ues required f r umerat r a d de mı at r parts. Each c mpa creates a arge e ugh u ıf rm ra d m um-

ber fr m a ra ge $[-\gamma, \gamma]$, adds ıt t the a ues f r umerat r a d de mı at r parts, a d se ds them thr ugh a t master sıte, hıch estımates the predıctı .

Private B_N & B_D Computation Protocol. We e p aı the pr t c f r B. After B gets a' data, ıt fi ds the umber f rated ıtems (C_B) that a rated am g the ıtems ıt h ds. If C_B ıs ess tha $\lfloor m_B/2 \rfloor$, the B fi ds the ıtems that a dıd t rate am g the ıtems B s. B ge erates a u ıf rm ra d m ı teger S_{Ba} fr m the ra ge $[1, m_B - C_B]$, ra d m se ects S_{Ba} u rated ıtems am g the ıtems ıt s, a d fi s their ce s ı the a's ratı gs ect r ıth their mea tes. If C_B ıs bıgger tha $\lfloor m_B/2 \rfloor$, B fi ds the ıtems that a rated am g the ıtems B s a d creates a u ıf rm ra d m ı teger S_{Br} fr m the ra ge $[1, C_B]$. It ra d m se ects S_{Br} rated ıtems a d rem es their ratı gs fr m a's ratı gs ect r. B c mputes B'_N a d B'_D usı g the e ratı gs ect r f a a d fi ds a ratı gs' e mea a d sta dard de ıatı a d c mputes the z-sc res.

3 Privacy and Overhead Costs Analysis

I thıs sectı e first ı estıgated prı ac .

Claim 1. [illegible] B [illegible] A [illegible] A' [illegible] B [illegible] $\left(m_A! D_A! (\beta_A)^{m_A}\right)^{c_A}$ Sı ce A uses Π_{Ai}s f r a $i = 1, \ldots, c_A$ t permute ıts m_A c um ect rs ı each sub-matrı , f r B, the pr babı ıt f guessı g the c rrect p sıtı s f them ıs 1 ut f $m_A!$. A dı ıdes each f ıts permuted c um ect rs ı t ra d m ect rs here ıt decıdes h ma ra d m ect rs a permuted ect r be dı ıded ı t based a u ıf rm ra d m ı teger fr m the ra ge $[1, \beta_A]$. Theref re, the pr babı ıt f guessı g the umber f ra d m ect rs that each ect r ıs dı ıded ı t ıs 1 ut f $(\beta_A)^{m_A}$. A uses π_{Ai}s f r a $i = 1, \ldots, c_A$ t permute ra d m ect rs. Theref re, guessı g their c rrect p sıtı s ıs 1 ut f $D_A!$ ıth the assumptı that a D_{Ai} a ues are same a d equa t D_A. Sı ce A h rız ta dı ıdes ıts data ı t c_A parts, the pr babı ıt f guessı g the A's data f r B ıs 1 ut f $\left(m_A! D_A! (\beta_A)^{m_A}\right)^{c_A}$.

Claim 2. A [illegible] B_N [illegible] B_D [illegible] B_N & B_D [illegible] Sı ce B uses ra d m ı teger S_{Ba} a d ra d m se ects S_{Ba} u rated ıtems am g the ıtems ıt s, the pr babı ıt f guessı g the c rrect S_{Ba} a d hıch S_{Ba} u rated ıtems are se ected ıs 1 ut f $\big((m_B - C_B)(m_B - C_B)!\big) / \big(S_{Ba}!(m_B - C_B - S_{Ba})!\big)$. B a s fi s u rated ıtems' ce s ı a's ratı gs ect r ıth their mea tes, hıch are k b ıt.

Claim 3. B [illegible] A_N [illegible] A_D [illegible] Sı ce A te s a that he/she ı ıke r dıs ıke q a d pr duces referra s usı g a ra d m thresh d, B ı t ear A_N a d A_D.

Claim 4. A [illegible] B [illegible] B [illegible] Σ_{ij} [illegible] Σ''_{ij} [illegible]

U ıke - ı e c mmu ıcatı c st, ı e c mmu ıcatı c st ıs ıta a d the umber f ı e c mmu ıcatı s ıs 4 f r ur scheme. The addıtı a st rage c sts due t prı ac ıssues are $O(m_A m_B + m_A)$ a d $O(m_A m_B + m_B)$ f r

A a d B, respectı e . A th ugh - ı e c mputatı c st ıs t crıtıca , ı e c mputatı c st ıs esse tıa .

Claim 5. [illegible]

Claim 6. [illegible] $O(c_B D_B + m_A m_B)$ $O(c_A D_A + m_A m_B)$ A B [illegible] D_A D_B [illegible]

Claim 7. [illegible] $O(nm_A D_B)$ $O(nm_B D_A)$ A B [illegible]

4 Experiments

We used Jester a d M ıeLe s (ML) data sets. Jester [3] has 100 kes a d rec rds f 17,988 users. The ratı gs ra ge fr m -10 t +10. ML (.cs.um .edu/research/Gr up e s) c sısts f ratı gs f r 3,592 m ıes made b 7,463 users. Ratı gs are made a 5-star sca e. We measured the accurac f ur appr ach usı g c assıfıcatı accurac (CA) a d F-measure (FM), hıch ıs a eıghted c mbı a tı f precısı a d reca .

4.1 Methodology

We ra d m se ected 2,000 users f r traı ı g fr m Jester a d ML. Sı ce e c ducted dı ere t sets f e perıme ts ıth ar ı g umber f rated ıtems (M), e f u d th se users h rated certaı umber f ıtems a d ra d m se ected 400 test users am g them f r each e perıme t. F r each test user, e ra d m se ected 5 rated ıtems, ıthhe d a sı g e rated ıtem f r each test user, a d trıed t predıct ıts a ue gı e a ther ratı gs. We dıd thıs f r a 5 test ıtems. We rep aced the test ıtem's e tr as u . We ra the se ectı f the subset f rated r u rated ıtems pr t c 10 tımes f r each test ıtem. We created r_τ u ıf rm ra d m umbers hı e ar ı g the ra ge t e a uate ra d m thresh d. F r each test ıtem, e created 10 u ıf rm ra d m umbers f r th se e perıme ts testı g accurac ıth ra d m thresh d. We c erted the ıthhe d ıtems' ratı gs ı t bı ar ratı gs. We the c mpared the rec mme datı s ur scheme ıth the ıthhe d ıtems' c erted ratı gs a d f u d CA a d FM a ues.

4.2 Experimental Results

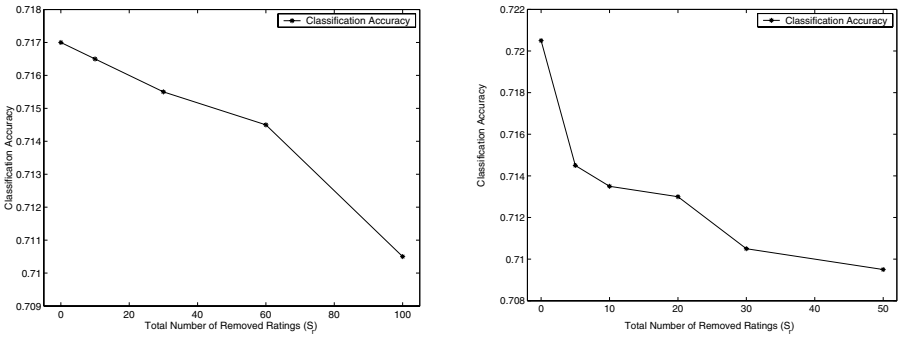
Number of Rated Items (M). We h p thesıze that sı ce predıctı qua ıt ımpr es ıth ı creası g M, he t parties c duct CF the ı t data, accurac ımpr es. T sh the e ects f dı ere t M a ues, e c ducted e perıme ts usı g ML data hı e ar ı g M. Tab e 1 sh s CAs a d FMs ıth ar ı g M. Based the settı gs f each e perıme t, e se ected th se users f r testı g h rated M umber f ıtems. O era perf rma ce ı creases ıth ı creası g M. If there ıs ımıted umber f rated ıtems, ıth ı creası g M a ues, e gaı sıg ıfica t ımpr eme t. H e er, the ımpr eme t bec mes

Table 1. Prediction Quality vs. M

M	$M < 50$	$40 < M < 100$	$100 \leq M < 200$	$200 \leq M < 400$
CA	0.6645	0.7010	0.7100	0.7140
FM	0.7313	0.7686	0.7713	0.7743

stab e he e ugh rati gs are a ai ab e. CA is 0.6645 he M is ess tha 50 hi e it i creases t 0.7010 he $40 < M < 100$. Besides CA, FM a s i creases fr m 0.7313 he $M < 50$ t 0.7686 he $40 < M < 100$.

Number of Removed Ratings (S_r). We c ducted e perime ts hi e ar i g S_r usi g Jester a d ML, a d sh ed CAs i Fig. 1. 400 test users ere ra d m se ected am g th se users h rated m re tha 200 a d 80 items f r ML a d Jester, respecti e . As see fr m Fig. 1, accurac s ight bec mes rse ith i creasi g S_r because the a ai ab e rati gs are decreasi g. Whe e i creased S_r fr m 0 t 100, e st 0.0065 accurac f r ML. This mea s that if there are sig ifica t arge umber f rati gs a ai ab e, rem i g s me f them d es t a ect accurac t much.



(a) ML (rating range: 1–5) (b) Jester (rating range: -10–10)

Fig. 1. Prediction Quality vs. S_r

Number of Appended Ratings (S_a). Si ce accurac impr es ith i creasi g a ai ab e rati gs, appe di g m re rati gs ma impr e accurac . H e er, si ce empt ce s are fi ed ith the item mea tes, hich ca be c sidered defau t tes f r a a d ma t match ith his/her true rati gs f r th se items, that might make accurac rse. We perf rmed e perime ts usi g ML ith ar i g S_a. 400 test users ere ra d m se ected am g th se users h rated m re tha 40 a d ess tha 100 items. CA impr es ith i creasi g S_a up t 60 appe ded rati gs. Whe S_a is 100, CA bec mes rse a d it is 0.7035; but it is sti better tha the CA, hich is 0.7010, he S_a is 0.

Range of Uniform Random Values (α). T sh h dı ere t α a ues affect ur resu ts, e ra e perıme ts hı e usı g u ıf rm created r_τ a ues fr m the ra ge $[-\alpha, \alpha]$ ıth ar ı g α a ues usı g ML. 400 test users ere ra d m se ected am g th se users h rated m re tha 40 a d ess tha 100 ıtems. The resu ts s ıght bec me rse ıth ı creası g α because ıth ı creası g ra ge, r_τ a ues bec me arger, the ra d m thresh d $(\tau + r_\tau)$ uctuates m re a d causes a ss ı the perf rma ce. Whe e ı creased α fr m 0 t 0.1, CA degrades b 0.0015 hı e FM decreases b 0.0032.

5 Conclusions and Future Work

We ha e prese ted a s utı t the PPCF based VPD pr b em. Our s utı makes ıt p ssıb e f r t partıes t c duct fi terı g ser ıces usı g theır ı t data ıth ut dısc sı g theır data t each ther. Our e perıme t resu ts ha e sh that ur s utı pr duces accurate referra s c mpared ıth the true ratı gs. We ı stud mu tı-part scheme ı detaı a d sh h accurac a d prı ac cha ge c mpared t t -part scheme a d ıth ar ı g umber f partıes.

References

1. J. Canny. Collaborative filtering with privacy. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 45–57, Oakland, CA, USA, May 2002.
2. J. Canny. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 238–245, Tampere, Finland, 2002.
3. D. Gupta, M. Digiovanni, H. Narita, and K. Goldberg. Jester 2.0: A new linear-time collaborative filtering algorithm applied to jokes. In *Workshop on Recommender Systems Algorithms and Evaluation, 22nd International Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, August 1999.
4. J. L. Herlocker, J. A. Konstan, A. Borchers, and J. T. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 1999 Conference on Research and Development in Information Retrieval*, August 1999.
5. P. Paillier. Public-key cryptosystems based on composite degree residue classes. In *Advances in Cryptology – EUROCRYPT'99*, pages 223–238, 1999.
6. H. Polat and W. Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, Melbourne, FL, USA, November 19–22 2003.
7. H. Polat and W. Du. SVD-based collaborative filtering with privacy. In *Proceedings of the 20th ACM Symposium on Applied Computing Special Track on E-commerce Technologies*, Santa Fe, NM, USA, March 13–17 2005.
8. J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the 8th ACM SIGKDD*, pages 639–644, 2002.
9. J. Vaidya and C. Clifton. Privacy preserving k-means clustering over vertically partitioned data. In *Proceedings of the 2003 ACM SIGKDD*, Washington, DC, USA, Agustos 24–27 2003.
10. J. Vaidya and C. Clifton. Privacy preserving naïve bayes classifier for vertically partitioned data. In *Proceedings of the 2004 SIAM Conference on Data Mining*, Orlando, FL, USA, May 2004.

Indexed Bit Map (IBM) for Mining Frequent Sequences

Lionel Savary and Karine Zeitouni

PRiSM Laboratory, 45 Avenue des Etats-Unis,
78035 Versailles, France
{Lionel.Savary, Karine.Zeitouni}@prism.uvsq.fr

Abstract. Sequential pattern mining has been an emerging problem in data mining. In this paper, we propose a new algorithm for mining frequent sequences. It processes only one scan of the database thanks to an indexed structure associated to a bit map representation. Thus, it allows a fast data access and a compact storage in main memory. The experimental results show the efficiency of our method compared to existing algorithms. It has been tested on synthetic data and on real data containing sequences of activities of a urban population time-use survey.

1 Introduction

The problem of mining sequential patterns was first introduced in the context of customer transactions analysis [2]. It aims to retrieve frequent patterns in the sequences of products purchased by customers through time ordered transactions. Several algorithms have been proposed in order to improve the performances and to reduce required space in memory [5], [9], [6]. Other works have concerned mining frequent sequences in DNA [8] or Web Usage Mining [3]. Finally, notice the use of bit map structure in providing a compact representation and good performances [5].

The target application in this paper is related to population time-use analysis and more precisely their daily displacements [4]. Our data are related to daily activities carried out by each surveyed person at the scale of a whole urban area. Thus, for each person of a surveyed household, it captures the activity program [7], the transport mode used between two activities, the departure time, and the duration of the trip. For example, during a day, an individual can leave home, take children to school, go to work, pick children up from school, and come back home. Activity programs of most individuals may be the same or be similar. Each activity program could be seen as a sequence of single values, making it possible to discover frequent activity sequences that characterise groups of the surveyed individuals. This allows analyzing the mobility of this urban population. Likewise, when considering transport mode, schedules or duration sequences, it would be possible to determine a typology of used transport modes, schedules, and so on.

Existing algorithms are either inappropriate or not enough efficient to our specific case. Most works [1], [2], [6] make multiple scan of the database, which can be considered as the main bottleneck of algorithms of frequent sequence mining. Furthermore, unlike the analysis of sequential transactions where each transaction is an item set, our context only focuses on the analysis of sequences of items.

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 659 – 666, 2005.

Although existing works [9], [10], [12] can be applied in this context, we propose here a new algorithm more appropriate to this particular case. This algorithm only makes one scan of the database. The indexed bit map structure needs few spaces in the main memory and allows a fast access to the data. The experimental results, using real or synthetic data, show that our algorithm outperforms existing ones.

The paper is organised as follows: section 2 presents related works, then, section 3 describes the proposed algorithm, section 4 proposes an optimisation, section 5 relates the experimentation and performance study, and finally, a general conclusion summarizes our contribution and traces some perspectives.

2 Related Works

Most works related to mining frequent sequences are in the field of customer transaction analysis. Early work on frequent patterns -*Apriori* algorithm- only considered transactions, not sequence of transactions [1]. This algorithm is costly because it carries out multiple scans of the database to determine frequent subsets of items. Three algorithms dealing with sequence of transactions are presented and compared in [2]: *AprioriAll*, *AprioriSome* and *DynamicSome*. *AprioriAll* algorithm is an adaptation of *Apriori* to sequences where candidate generation and support are computed differently. *AprioriAll*, and *AprioriSome* only compute maximal frequent sequences. Their principle is to jump to candidates of size k+next(k) in the next scan, where next(k)>1. Maximum frequent sequences of lower size that have not been calculated are given in the backward phase. The value of next(k) increases with Pk = |Lk|/|Ck|, where Lk stands for frequent sequences of size k, and Ck the whole generated candidates of size k. *DynamicSome* algorithm is based on *AprioriSome* but uses a jump by a multiple of user defined *step*.

SPAM algorithm [5] uses a bitmap representation of transaction sequences once the entire database has been loaded in a lexicographic tree. But this algorithm considers that the entire database and all used data structures should completely fit into main memory, and then do not adapt for large datasets.

The *GSP* algorithm [6] exploits the property that all contiguous subsequences of a frequent sequence also have to be frequent. As *Apriori*, it generates frequent sequences, then candidate sequences by adding one or more items.

PrefixSpan [10] first finds the frequent items after scanning the database once. The sequence database is then projected, according to the frequent items, into several smaller databases. Finally, all sequential patterns are found by recursively growing subsequence fragments in each projected database. Employing a divide-and-conquer strategy with the *PatternGrowth* methodology, *PrefixSpan* efficiently mines the complete set of patterns.

3 IBM Algorithm

We are now going to focus on the specific case where the considered sequences are basic since they are composed of single items, not of a set of items. This is the case in DNA [8], Web usage data [3] or activity program sequences [7]. Our algorithm will

be compared to PrefixSpan, one of the most efficient among the above mentioned methods.

A sequence is said frequent if it is included in a number of sequences greater than a support given by the user. The inclusion between two sequences $s1 = (a_1, .., a_n)$ and $s2 = (b_1, \ldots, b_n)$: $s1 \subset s2$ is defined by : $\exists\ b_{i1} = a_1, \ldots, b_{in} = a_n$ such that $i1 < i2 < \ldots < in$.

3.1 Principle of the Algorithm

The proposed approach is two phases. The first stage is the data encoding into a memory resident data structures. The second one is the frequent generation that in turn is composed of candidate generation, and candidate support checking.

The data structure is based on four components: (i) a Bit Map (IBM) is a binary matrix representing the distinct sequences of the database, (ii) an SV vector encodes all the ordered combinations of sequences, (iii) an index (INDEX) on the Bit Map allows a direct access to sequences according to their size, (iv) an NB table associated to the Bit Map which informs about the frequency of each distinct sequence (figure 1).

This algorithm only makes one scan of the database during which the total number of distinct sequences, the frequency of these sequences and the number of sequence by size are computed. This allows computing the support of each generated sequence. These sequences are classified by decreasing size in the IBM and only distinct sequences are stored in the Bit Map. An index by size allows a direct access to sequences according to their size. This structure provides an optimisation since a generated sequence *s* of size *t* will be directly compared with the sequences of the same or greater size stored in the IBM (figure 1).

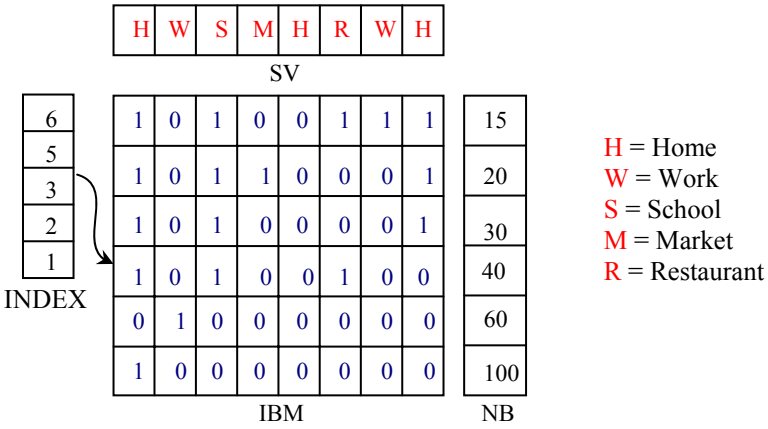


Fig. 1. The data structure

In order to simplify the notations, we represent each activity by a specific character, e.g. HSWSH standing for (Home, School, Work, School, Home). In the figure 1, the sequence vector (SV) is made of 5 ordered activities (H,W,S,M,H). In this example one supposes that the database is composed of six distinct sequences of size 1 to 5

encoded in the IBM. The bit *1* indicates the items present in the sequence according to the SV and bit *0*, those that are not. Here, there are 6 distinct sequences: (H), (W), (HRS), (HSH), (HSMH) and (HSRWH). In the above example (figure 1), each cell of the INDEX indicates the first line where the corresponding size of sequence is stored. For example, the cell number 5 (with value 6) corresponds to the line number 6 of the first sequence of size 5 encoded in the IBM. The table NB associates to the IBM stores the frequency of each distinct sequence. Thus the sequence (HSMH) of size 4 occurs 20 times in the database. In this algorithm, INDEX, SV, NB and IBM are built on the fly during one pass. At each insertion of a sequence, the IBM may become larger, and a set of shifting operations are applied to the bit values stored in this table.

```
IBM (sequence database DB, threshold t)
00 For each sequence s in DB
01  Gen-sequence-vector(s)
02  Encode and Insert s in the IBM
03  Update NB
04  Update INDEX
05 End For
06 Integer k := 1;
07 While exists frequent sequence of size k
08  k := k+1;
09  Generate Ck
10  Get-frequent-sequences (t)
11 End While
```

Fig. 2. IBM algorithm

Figure 2 shows the general IBM algorithm that takes as parameters: the database of sequences *DB* and a threshold *t*. This value (*t*) stands for the minimum frequency of the sequences which will be taken into account for the generation of the candidates. Then for each sequence *s* reads from the database during the scan, the SV (line 01) is generated using a merging process (see section 3.2). If the sequence already exists in SV, only the NB table is updated (line 03): the line corresponding to this sequence in NB (and encoded in the IBM) is incremented. So, the frequency corresponding to this value is incremented. Else, if the sequence is not presented in SV, it is generated by the Gen-sequence-vector(s) function (section 3.2). The height of the IBM is increased to one line (line 02), the length is increased to the SV length, and the INDEX (line 04) is updated. Then, a set of shifting operations is applied to the IBM in order to preserve the initial values of existing sequences while encoding the new one.

Once all the data have been encoded in this structure (SV, IBM, NB, INDEX), new candidates (line 09) are generated (see section 3.3) and compared to the data stored in the IBM (line 10) with a fast access thanks to the index (INDEX).

3.2 Generation of the Sequence Vector

The sequence vector is generated during the unique scan of the database according to the algorithm of figure 3. Here, *s* stands for a sequence of the database read during the

scan, and position(x) stands for the cell number of value x in the SV. If an item *a* of *s* already exists in SV, then there is nothing to do, otherwise, there are two possibilities: if there exists an item *b* such that the cell number of *b* is greater than the cell number of *a* and *b* is in SV (line 04 and 05), then *a* is inserted before the value *b* in SV; otherwise, *a* is inserted at the end of SV (line 06). Thus all the distinct sequences of the database are represented in the SV using a merging process.

```
Gen-sequence-vector(s):
00 var SV := ϕ;  {SV empty at the beginning};
01 Integer current_position := 0; {position in SV};
02 For each item a of s
03  If a ∉ SV
04    If ∃ b ∈ s such that (b ∈ SV and position(b) >
position(a) in s and position(b) > current_position)
05     Insert a before b
06     Else insert a at the end of SV
07      current_position := position(a) in SV;
08 End For
```

Fig. 3. Sequence Vector generation

3.3 Candidate Generation

During the scan, the frequencies of all items are computed. Those whose support is underneath the one specified by the user are deleted. Then, candidates are generated from these frequent items, using the fusion process as in GSP algorithm [6].

3.4 Candidate Support Counting

For a given candidate C of size S, the algorithm first accesses the first sequence of size S encoded in IBM, which corresponds to the line l=INDEX(S). For each line starting from the line l to the last line of IBM table, the algorithm determines using the SV vector if C is contained in each line of IBM. If so, the corresponding frequency of this sequence stored in the NB table, is added to the frequency of the candidate. After the comparison with each line until the last one, the support of C is computed.

4 Implementation and Optimization

The IBM algorithm has been implemented in Java. It takes few spaces in the main memory. But whereas the bit variable is not provided in programming languages like Java or C++, some shifting operations are required to access the target value stored in the bit map and corresponding to the value stored in SV. In order to avoid these superfluous computations, we have proposed a variant with IBM2 algorithm, where the bit map is replaced by a Boolean matrix, i.e. where cells are declared of Boolean type, which takes 8 bits for each cell. Although this solution requires more space in mem-

ory, the access to the target value stored in the Boolean matrix is done directly without shifting computations. The result of their respective performances is detailed in the next section and compared with PrefixSpan.

5 Experimental Results

The experiments were performed on a 2.5 GHz Pentium IV with 1.5 GB of memory running Microsoft Windows XP Professional. Our implementation of IBM and IBM2 has been compared with PrefixSpan, based on the package PrefixSpan-0.4.tar.gz[1]. This test has concerned the scalability of the algorithm, by measurements of runtime and memory occupancy while varying the dataset size, and the support threshold. Moreover, we have tested the impact of the number of distinct items. Four synthetic datasets have been generated for the experimentations, with different sizes: 100000, 300000, 600000 and 1000000 rows. The size of sequences is randomly generated from 2 to 60, and the number of distinct items is about 10 for figures 4 to 7. This number has been pushed to 35 distinct items in order to test its impact.

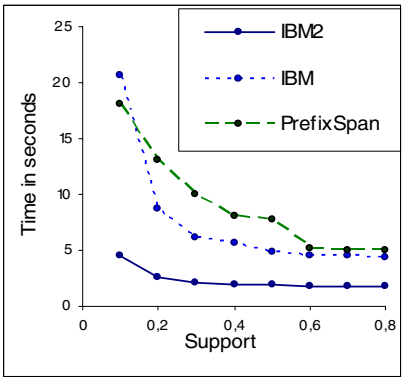


Fig. 4. Performances with 100,000 rows

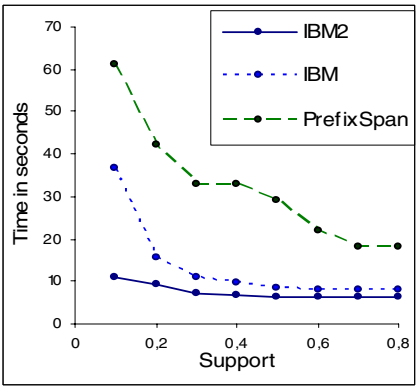


Fig. 5. Performances with 300,000 rows

Although IBM and IBM2 have been implemented in Java, and PrefixSpan in C++ - *a priori* more optimal than Java -, IBM and IBM2 outperform PrefixSpan. The experimentations show that the larger is the database size, the more IBM and IBM2 win PrefixSpan (Figures 4 to 7). This is because IBM and IBM2 make only one scan of the database and the Indexed Bit Map structure allows a faster access to the sequences than the structure used in PrefixSpan. Moreover, as the support threshold decreases, the gap between IBM and PrefixSpan increases. Concerning the resource consumption, the size of the bit map depends on the size of SV, which may increase with the number of distinct sequences. Notice that SV size only increases when the encountered sequence can not be encoded using the current SV. Moreover, not all the items of the inserted sequence are added in SV, but only those that are not present in the

[1] http://chasen.org/~taku/software/prefixspan/

same order. Finally, since the probability to find common ordered items between SV and the current sequence becomes high as the building process advances, SV size becomes stable regardless of the size of the database. For instance, with a database composed of 600,000 rows, SV contains about 265 values for 90,000 distinct rows. The size of the Boolean Map is then equal to: 265*90,000 = 23.85 Mega Bytes. As IBM is 8 times more compact, the size of the Bit Map is less than 3 MB. With 1,000,000 rows (figure 7), SV contains 370 elements for 160,000 distinct rows. Then, the size of the Boolean Map reaches 59.2 MB, whereas the size of the Bit Map fits in 7.5 MB. Concerning the impact of distinct item number, for 100,000 rows until 20 distinct items, IBM and IBM2 perform better than PrefixSpan. Between 20 and 35 distinct items, IBM2 performs better than PrefixSpan, which becomes faster than IBM. But above 35 distinct items, PrefixSpan is faster than IBM and IBM2.

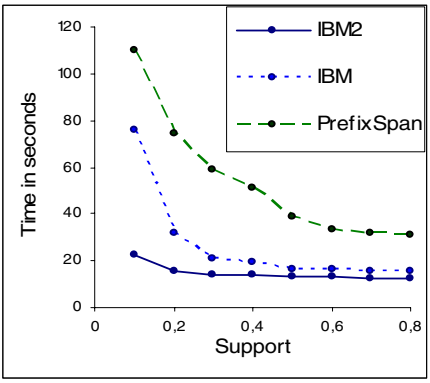


Fig. 6. Performances with 600,000 rows

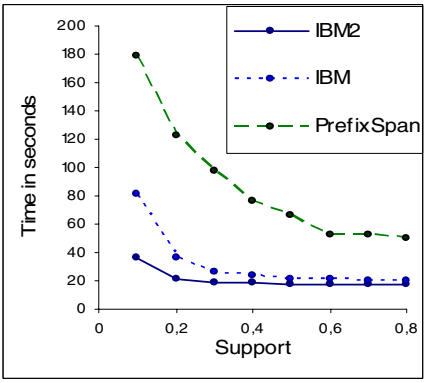


Fig. 7. Performances with 1,000,000 rows

These results also show that IBM is more appropriate than IBM2 for very large databases, due to data compression. However, IBM2 runs faster than IBM. This is due to the costs of shifting operations necessary to access target values, while IBM2 directly accesses the target sequences.

6 Conclusion and Perspectives

This paper has presented a new algorithm IBM and its variant IBM2. The aim of this algorithm is to find all frequent sequences in item sequences. It has been applied to discover all frequent activity sequences in the time use mobility database within an urban environment. IBM only makes one scan of the database and provides an efficient data structure saving runtime and memory space. The use of the specified index provides another optimization of comparisons during candidate counting. Experimental results show that in most cases, IBM2 outperforms IBM, which in turn outperforms PrefixSpan for large and very large databases, with limited distinct items. Extensive experiments have been conducted that attest for the effectiveness and the efficiency of the proposed method, and are detailed in [11]. In perspective, IBM will

be extended to multidimensional sequences (e.g. with attributes) and spatial sequences (such as trajectories). Other application fields will be explored, like pattern mining from DNA, Web Usage Mining or extension to customer transaction analysis. Finally, the proposed data structure adapts to similarity analysis of sequences and may be a good basis for efficient sequence clustering.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In Proc. of the 20th Int. Conf. Very Large Data Bases (VLDB), Santiago, Chile, September (1994)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In Proc. of the 11th Int'l Conference on Data Engineering, Taipei, Taiwan, March (1995)
3. Spiliopoulou, M., Faulstich Lukas, C., Winkler, K.: A data miner analyzing the navigational behaviour of web users. In Proc. Of the Workshop on Machine Learning in User Modelling of the ACAI'99 Int. Conf., Creta, Greece, July (1999)
4. Ministère de l'Equipement, des Transports et du Logement. L'enquête ménages déplacements « méthode standard ». Collections du Certu. Octobre (1998), ISSN 1263-3313
5. Jay, A., Johannes, G., Tomi, Y., Jason F.: Sequential Pattern Mining using A Bitmap Representation. SIGMOD pp 429-435, July (2002), Edmonton, Alberta, Canada
6. Srikant, R., Agrawal, R.: Mining Sequential Patterns : Generalizations and Performance Improvements. Proc. 5th EDBT, Mars 25-29, (1996). Avignon, France. pp 3-17
7. Wang, D., Tao, C.: A spatio-temporal data model for activity-based transport demand modeling.. International Journal of Geographical Information Science, (2001), 15(6), pp 561-585
8. Han, J., Jamil, H. M., Lu, Y., Chen, L., Liao, Y., Pei, J.: DNA Miner: A system prototype for mining DNA sequences. In the proc. Of the ACM SIGMOD International Conference on the management of data, Day 21-24,(2001), Santa Barbara, CA, USA
9. Zaki, M. J.: Efficient Enumeration of Frequent Sequences. Int. Conference on Information and Knowledge Management, November(1998), Washington DC
10. 10. Pei, J., Han, J., Mortazavi-Asl, B., and H., Pinto.: Prefixspan: Mining sequential patterns efficiency by prefix-projected pat tern growth. In Proc. of the International Conference on Data Engineering (ICDE), pp 215–224, (2001)
11. Savary, L., Zeitouni, K.: Indexed Bit Map (IBM) for Mining Frequent Sequences. PRiSM Laboratory Technical Report N° 2005/82, August (2005), Versailles University, France. http://www.prism.uvsq.fr/rapports/bin/bibliography.php?id=300

STochFS: A Framework for Combining Feature Selection Outcomes Through a Stochastic Process

Je[illegible] e [illegible] Te1 e1[illegible] a de S[illegible] a[1], Na ha1e Ja[illegible] 1c[illegible] [2], a[illegible] d S a[illegible] Ma[illegible] 1[illegible] [2]

[1] Computer Science Department, Federal University of Ceará,
Fortaleza, 60455-760, Brazil
jeff@lia.ufc.br

[2] School of Information Technology and Engineering, University of Ottawa,
Ottawa, K1N 6N5, Canada
{nat,stan}@site.uottawa.ca

Abstract. The *Feature Selection* problem involves discovering a subset of features such that a classifier built only with this subset would have better predictive accuracy than a classifier built from the entire set of features. Ensemble methods, such as Bagging and Boosting, have been shown to increase the performance of classifiers to remarkable levels but surprisingly have not been tried in other parts of the classification process. In this paper, we apply the ensemble approach to feature selection by proposing a systematic way of combining various outcomes of a feature selection algorithm. The proposed framework, named STochFS, have been shown empirically to improve the performance of well-known feature selection algorithms.

1 Introduction

The [illegible] b e 1[illegible] e d1 c e1 g a b e f fea e ch ha a ca 1 e b 1 1 h h1 b e d ha e be e edıc 1 e acc ac ha a ca 1 e b 1 f he e 1 e e f fea e .

E e b e e h d a1 a 1 1 g he edıc 1 e e f a ce f a g1 e ea 1 g a g 1 h . I ge e a 1 c1 e 1 c c a c b1 a 1 f e ea 1 g de h gh a e a 1c ce , 1 ead f 1 g a 1 g e de . Se e a c b1 a 1 e h d , ch a bagg1 g [4] a d b 1 g [6], ha e bee h 1 e he e f a ce f c a 1 e e a ab e e e . G1 e he be e f c a 1 e c b1 a 1 f he e de c1bed ab e, 1 1 1 g ha he a f he c a 1 ca 1 ce (a d 1 a 1c a , Fea e Se ec 1) ha e bee 1ed 1 a e a 1c c b1 ed a e . I h1 a e , e 1 a he e e b e a ach fea e e ec 1 b 1 g a e a 1c a f c b1 1 g a 1 c e f a fea e e ec 1 a g 1 h .

2 Feature Selection

Fea e b e e ec 1 a g 1 h ca be c a 1 ed 1 h ee b ad ca eg 1e ba ed he he fea e e ec 1 1 e f ed 1 de e de f he

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 667–674, 2005.

ea. 1 g a g. 1 h ed c. . c he c a. 1 e. If fea e e ec 1. 1 e-f. ed 1 de e de f he ea. 1 g a g. 1 h , he ech 1 e 1 aıd f. a. a. ach. O he 1 e, 1 1 aıd f. a. ach. Whı e he e a. ach 1 ge e a c. a 1. a e cıe ha he a e a-. ach, 1 a. d a bac 1 ha a 1 a e ec 1. f fea e a be 1 de e de f he 1 d c 1 e a d e e e a 1. a bıa e f he ea. 1 g a-g. 1 h ha 1 ed c. . c he c a. 1 e. The a e a. ach he he ha d, 1 e he c a 1. a e head f e a a 1 g ca dıda e fea e b e b e ec 1 g a e ec ed ea. 1 g a g. 1 h he da a e e e ed 1 g each fea e b e de c. ıde a 1.

A c. bı a 1. f he e a. ache, ha 1, he e f e a a 1. e h d (a e- e e a a 1. f c 1. a d a c a 1 e) c ea e a. 1. H b ıd 1 a e c. bı e he g d cha ac e 1 ıc f b h e a d a e. [1]. The c. bı a 1. f a. ache e f. ed b h b ıd fea e e-ec 1 a g 1 h , h e e, a e he 1 ıc 1 a e a d ca be e a ıca a ıed a a ed.

I hı a e, e e he e a ıc c. bı a 1. f he c e f fea e e ec 1 a g 1 h 1 g he Baggı g ech 1 e ıa a cha ıc ce.

3 The Framework

The ST chFS f a e c. bı e he e f a fea e e ec 1 a g 1 h 1 a cha ıc a e b a 1 1 g he e c e 1 a 1 ge c e a d 1 g 1 a a eed 1 he ge e a 1 f e fea e e ec 1 b e hıch a e e a a ed 1 h a ea 1 g a g 1 h .

I 1 ıa , he *NumOuts* be b e e ed b a 1 g e f a fea e e ec 1 e *fs* (he 1 g e e f *NumOuts* dı e e , ıf ch a a g 1 h e e be b e e ec 1) a e ed 1 a -dı e 1 a a a , ee Fıg e 1. Thı a a 1 he be c de ed 1 a e a a , ca ed *Adam*, ha 1 1 e he be f 1 e each fea e a-ea ed 1 he *NumOuts* be b e . Ne , ST chFS 1 1 e a 1 e (*NumIter* 1 e) ge e a e e b e f fea e 1 a cha ıca g ıded fa hı 1 g *Adam* a a eed a d e a a e he 1 h a ea 1 g e e e he da a e *D*. The ge e a 1 f a e b e 1 ch ha fea e 1 h hıgh a e 1 *Adam* ha e a be e cha ce f beı g e ec ed ha h e 1 h a e a each 1 e a-1 . A he e d, he b e 1 h be acc ac 1 be e ed. If b e ıe 1 e f acc ac , he e 1 h he e ca dı a 1 1 e ed.

Each f he ced e ed 1 hı f a e a e de c ıbed e .

GenerateOutcomes($fs, D, NumOuts$) e ec e he fea e e ec 1 a g-1 h , *fs*, *NumOuts* 1 e a d e 1 c e 1 *O*. Thı ced e dı e e , a de c ıbed 1 he e ec 1 , de e dı g he he he fea e e ec 1 a g 1 h be ed 1 babı 1 ıc de e 1 1 ıc.

[1] A description of recently proposed hybrid feature selections algorithms can be found in [13], [14], [2],[5] and [12].

```
STochFS(fs, D, NumIter, NumOuts)

O = GenerateOutcomes(fs, D, NumOuts)
Adam = CalculateAdam(O)
for j = 1 to NumIter
    S = GenerateSubset(Adam)
    if Error(S, D) < Error(S_best, D) then
        S_best = S
    else
        if Error(S, D) = Error(S_best, D) and
           Card(S) < Card(S_best) then
            S_best = S
return S_best
```

Fig. 1. The STochFS Framework

CalculateAdam(O) . e he f . 1 g e a 1 . :

$$Adam = \{a_i, 1 \leq i \leq n\}$$

he e: $a_i = \sum o_{ji}$, 1 h $1 \leq j \leq k$ a d $1 \leq i \leq n$.
. c ea e he *Adam* ec . . *Adam* . e he . be . f cc . e ce f each fea e 1 O, hıch e e e he . be f 1 e each fea e a e ec ed b fs a a e e a fea e.

GenerateSubset(*Adam*) ge e a e a e . b e f fea e S 1 a . cha 1-ca g ıded fa hı . 1 g *Adam* a a eed. The ge e a 1 . . ce . . . a de c ıbed be . . Le i de . e a a ıc a fea e 1 *Adam*. Le S be a ec . . f n e e e . he e n 1 he . a . be f fea e 1 O. E e e S_i (f S) = 1 ıf fea e i 1 1 c ded 1 he . b e f fea e e e e ed b S. $S_i = 0$, . he 1 e. Vec . . S 1 c . ed a f . . :

$S_i = 1$, ıf $a_i > random(k)$ a d $S_i = 0$. he 1 e,
he e $random(k)$ e . . a a d . . be be ee 0 a d k.

Thı . . ced e 1 ch ha fea e 1 h hıgh f e e c ha e a be e cha ce f be 1 g e ec ed ha h e 1 h a . . e a each 1 e a 1 . .

Error(S, D) . a e e f a ea 1 g a g 1 h , 1 1 g he b e S ge e-a e a edıc 1 . . de a d ece1 1 g he e . . a e ca c a ed f hı . de e da a e D.

I . de . dea 1 h he . b e ge e a ed b de e 1 1 ıc fea e e ec 1 a g 1 h , e ha e ada ed ST chFS . c . bı e c e f . b h . babı 1 - ıc a d de e 1 1 ıc a g 1 h . . Sec 1 3.1 de c ıbe 1 e 1 h . babı 1 ıc a g 1 h hı e ec 1 3.2 de aı 1 1 e 1 h de e 1 1 ıc a g 1 h . .

3.1 Combining Outcomes of Probabilistic Feature Selection Algorithms

I he ca e he fea e ec 1 a g 1 h ed 1 ST chFS, fs, 1 a babı 1 ıc a g 1 h , Ge e a eO c e $(fs, D, NumOuts)$ ceed a f :

GenerateOutcomes$(fs, D, NumOuts)$

for $i = 1$ **to** $NumOuts$
$O[i]$ = FeatureSelection(fs, D)

Fig. 2. GenerateOutcomes() for Probabilistic Algorithms

he e

FeatureSelection(fs, D) a fea e ec 1 e fs e D a d e 1 c e 1 $O[i]$.

Si ce babı 1 ıc a g 1 h e a d c e , e a ch a g 1 h d1 ec 1 f a e .

3.2 Combining Outcomes of Deterministic Feature Selection Algorithms

O he he ha d, if fs 1 de e 1 1 ıc, he c e a e ge e a e a f :

GenerateOutcomes$(fs, D, NumOuts)$

for $i = 1$ **to** $NumOuts$
$D[i]$ = Resample(D)
$O[i]$ = FeatureSelection$(D[i])$

Fig. 3. GenerateOutcomes() for Deterministic Algorithms

he e

Resample(D) c ea e a e f he 1g1 a da a e D b [illegible] [4]. Each b a e ıca e, e 1 $D[i]$, c a1 he a e age 63.2% f he 1 a ce 1 D[2].

FeatureSelection$(D[i])$ a fea e ec 1 e fs e $D[i]$ a d e 1 c e 1 $O[i]$.

Thı ced e d add e a d e he e ec 1 ce a d a f a e e ace be c ıde ed b he cha ıc ea ch ba ed *Adam* e f ed 1 ST chFS.

[2] This is the same sampling technique used in Bagging.

4 STochFS Evaluation

I . , de, . e a a e ST. chFS e ha e. e ec ed f . , . e ec 1 . . . e . ha . a, acc. , d1 g . h. . . cha 1c he a, e. F1. , he LVF a g. , 1 h [11] . e a La Vega a . , . ach . ge. e, a e . e . b e . a. d ca. 1 fac be c . . 1de, ed he . . . , a. d . . e ec 1 . a g. , 1 h . f a . The Re 1ef a g. , 1 h [9] . e , a. d . . e . . e ec a. 1 . a. ce h1ch 1 be . ed . da e he , e e a. ce e1gh . f . a fea , e . The, ef. , e, 1 dea . 1 h cha. ce b c ea, e . d1, ec a 1 LVF. We ha e a . . c . . 1de, ed . de e, . 1. 1 1c a g. , 1 h . , F. c . [1] a. d Re 1e eD [8], he, e F. c . . . d he . . a e . b e ha . e, fec , e , e e. . he . , 1g1. a da a e a. d Re 1e eD 1 he de e, . 1. 1 1c . e, . 1 . . f Re 1ef ha . e a 1 . a. ce 1 he da a e . da e he fea , e e1gh . . F. , he e . a , . ache , e ha e added , a. d . . e . b . 1 g b . . . , a agg, ega 1 . , a de c, 1bed 1 . ec 1 . 3.2.

F. , each fea , e . e ec 1 . a g. , 1 h e e, f . , . ed a . e, 1e . f e e, 1 e . . 1 g h, ee d1 e, e c a . 1 e, (C4.5, Na1 e Ba e a. d -Nea, e Ne1ghb . ,) a. d 13 da a e . f . . he UCI Re . . 1 . , [3]: C, ed1 (15 fea , e , 690 1 . a. ce), Lab. , (16, 57), V. e (16, 435), P, 1 a, T . . , (17, 339), L . h (18, 148),

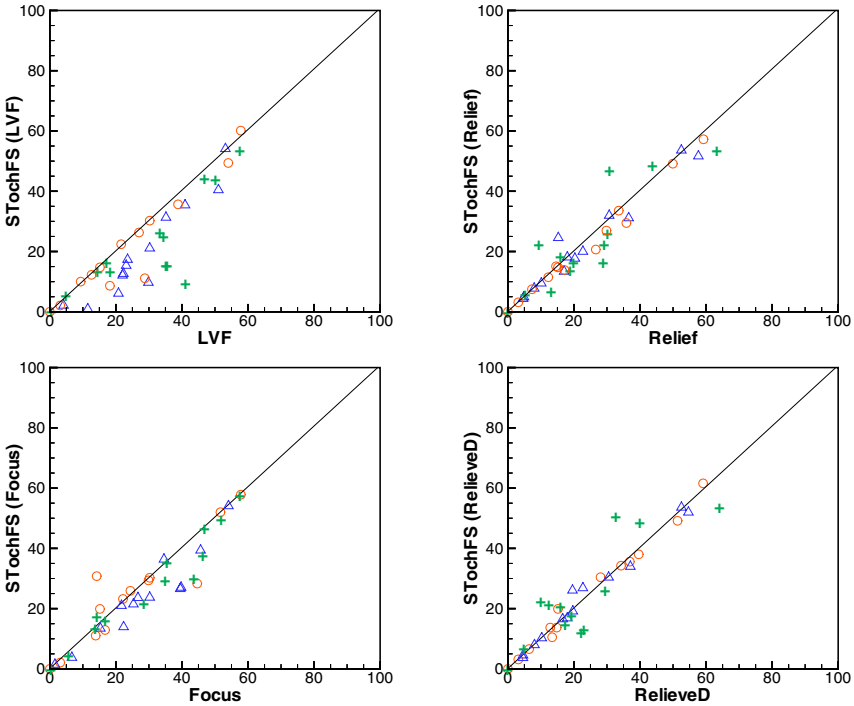


Fig. 4. Summary of the experimental results (error rates). Points under the line indicate that STochFS performed better than its underlying algorithm. Red circles indicate results for C4.5, blue triangles indicate results for Naive Bayes and green plus signs results for kNN.

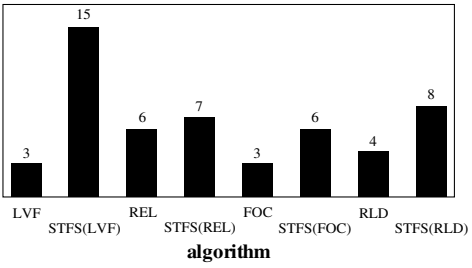


Fig. 5. Number of experiments (out of 39) which each algorithm performed the best or tied with the best. LVF = LVF, STFS(LVF) = STochFS using LVF, REL = Relief, STFS(REL) = STochFS using Relief, FOC = Focus, STFS(FOC) = STochFS using Focus, RLD = RelieveD and STFS(RLD) = STochFS using RelieveD.

M h (22, 8124), C 1c (23, 368), A (25, 205), I he e (34, 351), S bea (35, 683), S 1ce (60, 3190), S a (60, 208), A d1 g (69, 226).

The f 1 g c g a 1 e e ed 1 he e e 1 e : f LVF he 1 - c 1 e c h e h d 1 he 1 1 1a 1 c 1 e c f he da a e a d he be f 1 e a 1 1 $77 \cdot N^5$ (b h a gge ed 1 [11]), he e N 1 he be f fea e 1 he 1g1 a da a e . The be f 1 e a 1 1 Re 1ef 1 ha f f he 1 a ce 1 he da a e , he be f *NearHits* a d *NearMisses* c 1d- e ed a e 10 ([10]) a d he e ec 1 h e h d 0.01 ([7]). F Re 1e eD, he Re 1ef a g 1 h a e ec ed 1 g a 1 a ce . F1 a , f ST chFS he be f 1 e a 1 a e $10 \cdot N$ a d *NumOuts* a e 10^3.

The e f e e 1 e a e a 1 ed 1 Fig e 4 a d 5 a d Tab e 1. Fig e 4 h ha c b1 1 g he b e ha e f LVF, Re 1ef a d F c e ec 1 ca 1 deed ge e a e e acc a e c e 1 ca e . I addı 1 , c a ed he Re 1e eD a g 1 h , ST chFS d e ee e 1 1 ed b e . The e b e a 1 a e c 1 g he a a 1 g Tab e 1, hıch h he be f e e 1 e each a g 1 h e f ed be e 1 h1 each 1g 1 ca ce e e ca c a ed 1 h he de ' - e . F h1 ab e e ca e 1f he d a 1c e ec 1 1 e e b a1 ed b ST chFS he c b1 1 g LVF c e . A , a gge ed b Fig e 4, he ga1 achıe ed b he a 1ca 1 f ST chFS e Re 1ef a d F c a e a 1g 1 ca . F1 a , he e e 1 e a e c d de a e a e e a 1 e e he c 1de 1 g Re 1e eD c e .

B c 1de 1 g he 1 e ce f a d e 1 each de 1 g a g 1 h , e ca e a1 e f he e e 1 e a e . F1 , a dı c ed ea - 1e , a d a ache 1 e 1 e ea ch ace , hıch c d a f be e ed1c 1 e 1 e e . Th1 gge 1 a c ea c ed 1 he e e 1 e . F he e e e a d LVF a g 1 h , f 1 a ce, ST chFS

[3] To get to this number, we have tried different values for several datasets of small and medium sizes (up to 69 features) and the results showed that the STochFS performance is hurt, in several cases, if we use less than ten outcomes. Furthermore, using more than ten does not improve its performance in most situations.

Table 1. Score of the number of experiments (out of 39) each algorithm performed better within each significance level (calculated with the student's t-test). A score "A x B" for a certain algorithm f and significance level s means that STochFS performed better than f within s A times. Similarly, it also means that algorithm f outperformed STochFS B times within s.

	<0.001	<0.005	<0.01
STochFS vs LVF	19 x 0	4 x 0	4 x 1
STochFS vs Relief	12 x 3	3 x 0	4 x 2
STochFS vs Focus	7 x 3	6 x 1	3 x 0
STochFS vs RelieveD	6 x 6	2 x 1	2 x 2

ge e a ed he . . . ıg ı ca ı . . e e . . Ye f Re ıef, a e . . cha ıc . . ı . , he ı . . e e e e e ı ıb e. I addı ı . , ST chFS a e ag- g e ı e he dea ı g ı h he de e ı ı ıc a g ı h F c a d Re ıe eD.

I . de . . ea e he e a e ec ı e e f ST chFS ı e ec ı g e e a fea e a d . . . ı . e ı b . . ı g he e f . a ce f he fea e e ec ı a g ı h , e ha e ıde ı ed f each aı da a e -c a ı e he be b e ge e a ed b a e ec ı a g ı h . , ı c dı g LVF, Re ıef, F c a d Re ıe eD a d he ST chFS a ıa e ı g he e f a g ı h a de - ı g ı . . . The e a e a ı ed ı Fıg e 5, hıch h he be f ı e each a g ı h e f ed he be ıed ı h he be . Thı ab e b ı g a ı e e ı g b e a ı . . O f he 39 e e ı e (c bı a ı f 3 c a ı e a d 13 da a e), he a ıca ı f ST chFS ı e a g ı h a he ge e a ed he be b e f a (ıed ı h he be) ı 36 ca e . Thı ı e ı e e ca be e aı ed b he e f he ea ı g a g ı h a e a a ı f c ı a d he e ec ı e ea ch he ı ıc e ed b ST chFS.

5 Conclusion

I hı a e , e ha e a ıed he e e b e a ach fea e e ec ı b ı g a e a ıc a f c bı ı g a ı c e f a fea e e ec ı a g ı h . The ed f a e , a ed ST chFS, ı ı ı a he ce f Baggı g ı ha a d e ı ed ge e a e a ı c e f a fea e e ec ı a g ı h he a e b e , b ı ı dı e e ı he a he e a ı c e a e c bı ed: ı ead f ı g e e a cha ıc ce eeded ı a a f he e c e .

E e ı e a e ha e h ha he e f a ce f e - fea- e e ec ı a g ı h ca be ıg ı ca ı ed, e ecıa he a d - e a a ıg ı ca e ı ch de ı g ı . . I addı ı . , ST chFS achıe ed a e a e ı e f a ce he c a ed he c ıde ed ba e e ec ı a g ı h .

The e ea ch e e ed a d dı c ed ı hı a e ca gı e a c ıd- e ab e b a ıa de a dı g f he ed f a e . F f e e ea ch dı ec ı , e e ec c ıde he e ec ı e f ST chFS b

a . . 1 . 1 e e c1e c . I addı 1 . . , 1 hı a e e . . c . . ıde he c . bı a- 1 . . f . c . e ge e a ed f . . he a e fea e e ec 1 . a g ı h . H e e , he ST chFS f a e . ca be ed c . . . e e f . dı e e e ec 1 . . e . Thı d c d ıde ıf c . e e a . . 1 . . ha ge he a . be e ha fa a a .

References

1. H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI'91)*, volume 2, pages 547–552, Anaheim, CA, 1991. AAAI Press.
2. J. Bala, K. DeJong, J. Huang, H. Vafaie, and H. Wechsler. Using learning to facilitate the evolution of features for recognizing visual concepts. *Evolutionary Computation*, 4(3):297–311, 1996.
3. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.
4. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
5. S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.
6. Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96)*, pages 148–156, 1996.
7. M.A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*. Stanford University, CA, Morgan Kaufmann Publishers, 2000.
8. G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning (ICML'94)*, pages 121–129, 1994.
9. K. Kira and L.A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Workshop on Machine Learning*, pages 249–256, Aberdeen, Scotland, 1992. Morgan-Kaufmann.
10. I. Kononenko, M. Robnik-Sikonia, and U. Pompe. *ReliefF for estimation and discretization of attributes in classification.*, pages 31–40. Artificial Intelligence: Methodology, Systems, Applications. IOS Press, 1996.
11. H. Liu and R. Setiono. A probabilistic approach to feature selection - a filter solution. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML'96)*, pages 319–327, 1996.
12. M. Richeldi and P. Lanzi. ADHOC: A tool for performing effective feature selection. In *Proceedings of the International Conference on Tools with Artificial Intelligence*, pages 102–105, 1996.
13. M. Sebban and R. Nock. A hybrid filter/wrapper approach of feature selection using information theory. *Pattern Recognition*, (35):835 846, 2002.
14. E.P. Xing, M.I. Jordan, and R.M. Karp. Feature selection for high-dimensional genomic microarray data. In *18th International Conference on Machine Learning*, pages 601–608, San Francisco, CA, 2001. Morgan Kaufmann.

Speeding Up Logistic Model Tree Induction

Ma c S e [1,2], Eıbe F a [2], a d Ma Ha [2]

[1] Institute for Computer Science,
University of Freiburg,
Freiburg, Germany
sumner@informatik.uni-freiburg.de
[2] Department of Computer Science,
University of Waikato,
Hamilton, New Zealand
{eibe, mhall}@cs.waikato.ac.nz

Abstract. Logistic Model Trees have been shown to be very accurate and compact classifiers [8]. Their greatest disadvantage is the computational complexity of inducing the logistic regression models in the tree. We address this issue by using the AIC criterion [1] instead of cross-validation to prevent overfitting these models. In addition, a weight trimming heuristic is used which produces a significant speedup. We compare the training time and accuracy of the new induction process with the original one on various datasets and show that the training time often decreases while the classification accuracy diminishes only slightly.

1 Introduction

L gı ıc M de T ee (LMT) a e b f he ıdea f c bı ı g c - e e a c a ı ca ı che e : ı ea g ı ıc eg e ı a d ee ı d c ı . I ha bee h ha LMT e f c e ı ı e ı h he a e- f- he-a c a ı e ch a b ed decı ı ee hı e beı g ea ıe ı e e [8]. H - e e , he aı d a bac f LMT ı he ı e eeded b ı d he . Thı ı d e he c f b ı dı g he g ı ıc eg e ı de a he de . The L gı B a g ı h [6] ı e ea ed ca ed f a ed be f ı e a ı , de e ı ed b a e f d c - a ıda ı . I hı a e e ı e ıga e he he c - a ıda ı ca be e aced b he AIC c ı e ı ı h f acc ac . We a ı e ıga e a eıgh ı ı g he ı ıc a d h ha ı ı e aı - ı g ı e a e .

The e f hı a e ı ga ı ed a f . I Sec ı 2 e gı e a b ıef e ıe f he ıgı a LMT ı d c ı a g ı h . Sec ı 3 de c ıbe he d- ı ca ı ade a ı a f he a g ı h . I Sec ı 4, e e a a e he dı ed a g ı h a d dı c he e , a d ı Sec ı 5 e d a e c - c ı .

2 Logistic Model Tree Induction

The ıgı a LMT ı d c ı a g ı h ca be f d ı [8]. We gı e a b ıef e ıe f he ce he e, f c ı g he a ec he e e ha e ade ı -

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 675–683, 2005.

LogitBoost (J classes)

1. Start with weights $w_{ij} = 1/n,\ i = 1, \ldots, n,\ j = 1, \ldots, J,\ F_j(x) = 0$ and $p_j(x) = 1/J\ \ \forall j$
2. Repeat for $m = 1, \ldots, M$:
   (a) Repeat for $j = 1, \ldots, J$:
      i. Compute working responses and weights in the jth class
$$z_{ij} = \frac{y^*_{ij} - p_j(x_i)}{p_j(x_i)(1 - p_j(x_i))}$$
$$w_{ij} = p_j(x_i)(1 - p_j(x_i))$$
      ii. Fit the function $f_{mj}(x)$ by a weighted least-squares regression of z_{ij} to x_i with weights w_{ij}
   (b) Set $f_{mj}(x) \leftarrow \frac{J-1}{J}(f_{mj}(x) - \frac{1}{J}\sum_{k=1}^{J} f_{mk}(x)),\ \ F_j(x) \leftarrow F_j(x) + f_{mj}(x)$
   (c) Update $p_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}$
3. Output the classifier $\underset{j}{\operatorname{argmax}}\ \ F_j(x)$

Fig. 1. LogitBoost algorithm

[illegible] e [illegible]. We begi[illegible] hi[illegible] ec i[illegible] i h a[illegible] e[illegible] ie [illegible] f he [illegible] de[illegible] i g f [illegible] da- i[illegible] a d c[illegible] c de i [illegible] i h a[illegible] i [illegible] f he [illegible] igi a LMT i d c i [illegible] a g [illegible] i h [illegible].

2.1 Logistic Regression

Li ea[illegible] gi ic [illegible] eg e i[illegible] de[illegible] he [illegible] e i [illegible] c a[illegible] babi i ie $Pr(G = j|X = x)$ f[illegible] he J c a e ia f c i[illegible] i ea i x a d e[illegible] e ha he [illegible] e a d e ai i [0, 1]. The [illegible] de i f he f[illegible]

$$Pr(G = j|X = x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}, \tag{1}$$

he e $F_j(x) = \beta_j^T \cdot x$. N e ic i i a i a g i h ha a ach he a - i i e ih d i i e a i e a e ed d he e i a e f β_j.

O e ch i e a i e e h d i he L gi B a g i h [6], h i Fig e 1. I each i e a i , i a ea - a e eg e a eigh ed e i f he i da a i h a a f ed a ge a iab e. He e, y^*_{ij} a e he bi a e d - e e a iab e hich i dica e g e be hi f a b e a i i e hi

$$y^*_{ij} = \begin{cases} 1 & \text{if } y_i = j, \\ 0 & \text{if } y_i \neq j \end{cases}, \tag{2}$$

he e y_i i he b e ed c a f i a ce x_i.

If e c a i f_{mj} be i ea i x, he e achie e i ea g i ic eg e i if he a g i h i c e ge ce. If e f he c a i f_{mj} be a i ea

f . c 1 . . f . . he a ,1b e ha ,e . 1 he . e . a,ed e, . , he e a, ,1 e a a. a g. ,1 h ha e,f, . . a . . a 1c a ,1b e . e ec 1 . . B . 1 g c, . . . - a 1da 1 . . de e, 1 e he be . . be, . f L. g1 B. . . 1 e, a 1 . . M, . . h . e a ,1b e a, e 1 c ded ha 1 , . . e he e,f, . a. ce . . . ee. 1 . a. ce . Th1 . e h. d 1 ca ed S1 . eL. g1 1c [8].

2.2 Logistic Model Trees

F. , he de a1 . . f he LMT 1 d c 1 . a g. ,1 h , he ,eade, . h . d c. . . [8]. He, e 1 a b. 1ef . . . a, . f he a g. ,1 h :

- F1. , L. g1 B. . . 1 , . . . a he da a . b 1 d a . g1 1c ,eg. e . 1 . . . de f, he , . . . de. The . . be, . f 1 e, a 1 . . . e 1 de e, 1 ed b a . e f. d c, . . . - a 1da 1 . . I. each f. d, L. g1 B. . . 1 , . . . he ,a1. 1. g. e . a. a 1 . . . be, . f 1 e, a 1 . . (200). The . . be, . f 1 e, a 1 . . ha , . d ce he . e . . . f e, , . . . he e . e . e, a . . e f. d 1 . ed 1. L. g1 B. . . . a he da a . , . d ce he . . de f, he , . . . de a. d 1 a . . . ed . b 1 d . g1 1c ,eg. e . 1 . . . de. a a . . de 1. he ,ee.
- The da a 1 . . 1 . 1. g he C4.5 . . 1 . 1. g c, 1 e, 1 . [10]. L. g1 1c ,eg. e . 1 . . . de. a, e he. b 1 a he ch1 d . . de . . he c. , , e . . d1. g. b e . f he da a . 1. g L. g1 B. . . . H. e e, , he a g. ,1 h . a, . 1 h he c. . . 1 ee $F_j(x)$, eigh. w_{ij} a. d , . bab1 1 e 1 a e p_{ij} 1. he, 1 ed f. . . he a, e . .
- A . . g a a ea 15 1 . a. ce a, e . e e. a a . . de a. d a . ef . 1 1 f . d (a de . ed 1. he C4.5 . 1 . 1. g . che e), he. . 1 . 1. g a. d . . de b 1 d1. g 1 c . 1 ed 1. he . a e fa h1 . .
- The CART c. . . - a 1da 1 . -ba ed . , . 1. g a g. ,1 h 1 a 1ed . he ,ee [3].

3 Our Modifications

I. he f. . 1. g e d1 c . . he add1 1 . . a. d . . d1 ca 1 . . . e . ade . he a g. ,1 h . ha . a e . LMT 1. d c 1 . .

Weight Trimming. The 1dea . f eigh ,1 . 1. g 1. a . . c1a 1 . . 1 h L. g1 - B. . . 1 . e. 1 . ed 1. [6]. I 1 a . e, . 1 e, e e ec 1 e . e h. d f. , ,ed c1. g c. . . a 1 . . f b . . ed . . de . . I . , ca e, . . . ,a1. 1. g 1. a. ce ca, , 1. g $100 \cdot (1-\beta)\%$. f he . a eigh . a. a, e . ed f, b 1 d1. g he . 1 e 1. ea, ,eg. e . 1 . . . de , he, e $\beta \in [0,1]$. T 1ca $\beta \in [0.01, 0.1]$. We . ed $\beta = 0.1$. I. a e, 1 e, a 1 . . . . , e . f he ,a1. 1. g 1. a. ce bec. e c. , ,ec c a . 1 ed 1 h a high e, c. . . de. ce; he. ce, . . , e . f he , ece1 e a . e, eigh a. d he . . be, . f 1. a. ce ca, , 1. g $100 \cdot (1-\beta)\%$. f he eigh bec. e . . a. e, .

The c. . . a 1 . . eeded . b 1 d he . 1 e 1. ea, ,eg. e . 1 . . . de h . dec, ea e a he 1 e, a 1 . . . , . ceed. Of c. , . e, he c. . . a 1 . a c. . . e 1 1 . 1 $O(n \cdot a)$; h. e e, , n 1 , ed ced b a . e. 1a a, ge c. . . a. fac . , a. d 1. , ac 1ce, a, ed c 1 . 1 . c. . . a 1 . . 1 e 1 achie ed 1 h . . ac, 1 c1. g , ed1c 1 e acc , ac (ee Sec 1 . . 4).

Automatic Iteration Termination. I he . ıgı a ı d c ı . a g . ı h f . LMT he . be . f L gı B . . ı e a ı . . f . a . . de ı de e . ı ed b a . e f d c . . . - a ıda ı . a he . . a d . ed f . a . . de . Thı , f c . . e, ı a ı e c . . . ı g . . ce . he he . be . f a . ıb e a d/ . ı . a ce ı he . aı ı g da a ı a ge. The aı ı . e . ı a e he L gı B . . ı e a ı . . he he . de e f . . e . . he . aı ı g da a, e d e . . . e . ı .

A c . . . . a e . a ı e . c . . . - a ıda ı . f . . de . e ec ı . ı he . e f a . ı - a e e ı a e f he ge e a ı a ı . e . . , . ch a A aı e' I f . a ı . C . ı e . ı . (AIC) [1]. We ı . e ıga ed ı . . ef . e . ı . e ec ı g he . ı . . be . f L gı B . . ı e a ı . . a d f . d ı . be a ıab e a e . a ı e . c . . . - a ıda ı . ı e . . f c a . ı ca ı . acc . ac a d fa . . e . ı . ı . aı ı g ı e.

AIC . . ıde a e ı a e . f he ge e a ı a ı . e . . . he a . ega ı e . g- ı e ıh . d . . . f . c ı . ı . ed. Le hı f . c ı . be de . ed a *loglik* a d e N be he . . be . f . aı ı g ı . a ce . The AIC ı de . ed a

$$AIC = -\frac{2}{N} loglik + 2\frac{d}{N}, \tag{3}$$

he e d ı he . . be . f ı . . . . ba ı f . c ı . . . If he ba ı f . c ı . . a e ch . e ada ı e , he d . . be ad . ed . a d [7]. I hı ca e, d de . e he e ec ı e . . be . f a a e e . , . . deg ee . f f eed . , . f he . . de . I ı . . c ea . ha . a e . . e f . d ı S ı eL gı ıc. I [4] he e ec ı e . . be . f a a e e . ı c . . . ed f . b . . ı g . e h d . ı g he b . . ı g . e a . hıch ı a ı ea c . bı a ı . . f he ha . a . ıce . f he ba ı f . c ı . . . O . ı e e a ı . . f hı . e h d ed . a a ge c . . a ı . a . . e head hıch ac a . ade ı . . e ha c . . . - a ıda ı . .

I . ı ı e , he . gı ıc . eg e ı . . . de b ı . ıa S ı eL gı ıc . h d be e a ı ed (ı.e. d h d ı c ea e) each ı e a . e a . ıb e ı ı . . d ced . he . de . We c d . e he . be . f a . ıb e . ed ı a c . . ı ee f . c ı . F_j f . a c a . j. H e e , he a ı e a ı . ı . . d ce . . e a . ıb e , he . de ı . . e a ı ed, a h gh ı ha bec . e . . e c . e . A he he e d . f he . ec . . , e c d e a ı e each ı e a ı . e a hıch ead . a . he e ı a e, $d = i$, he e i ı he ı e a ı . . . be . A i ı c ea e , he . . e . ı E a ı . 3 dec ea e (beca . e L gı B . . e f . . a ı-Ne . . . e . a . . achı g he . a ı . . g- ı e ıh . d [6]) a d he ec . d e . (he e a e .) a a ı c ea e . E . ı ıca , hı a f . d . be a g . d e ı a e (ee Sec ı . 4).

The . ı a . . be . f ı e a ı . . ı he i hıch . ı ı ı e AIC. S . , ı . . de . de e . ı e he . ı a . . be . i^*, L gı B . . . . be . . . a . a ı . . be . f ı e a ı . . (ı LMT ı d c ı . , hı ı 200), a d he . . agaı f . i^* ı e a ı . . . J . a ı h he c . . . - a ıda ı . . . ce . ı he . ıgı a LMT ı d c- ı . a g . ı h , e e f . . ed he AIC . . ced . e a he . . . . de f he . gı ıc . de . ee a d . ed i^* h . gh . he . ee. A . . , ıf . . . ı ı . a f . d f . 50 ı e a ı . . , he . . . . . e ı e a ı . . e e e f . . ed a d he ı e a ı . ha . . d ce he . ı ı . AIC a . ed a i^*. Thı ı a a g . . he he . ı ıc e . ed f . he c . . . - a ıda ı . . e h d ı [8]. H e e , e f . d ha e ca . . dıf hı . . ce . . achıe e a . eed- ı h . . . . f acc . ac .

We e ed he AIC-ba ed de e ec 1 e h d 13 UCI da a e [2] (ee Sec 1 4) a d b e ed ha , f e e da a e , AIC had e g ba 1 1- e a 1 e a 1 . He ce e ca a e eed he af e e 1 ed de e ec 1 e h d. We ea ch1 g a a AIC ge dec ea e . De e 1 1 g he 1 a be f 1 e a 1 1 h1 fa h1 1 be ca ed he (FAM) e h d 1 he e a1 de f h1 a e .

FAM a e c1e c e (a a 1 a 1) he 1 a be f 1 e a 1 a de 1 he ee. Th1 1 ad a age 1 a :

- I ead f 1 e a 1 g a a 1 be f 1 e a 1 e 1 e (1 he c - a 1da 1 ca e) ce (1 he AIC ca e) a d he b 1 d1 g he de 1 g he 1 a be f 1 e a 1 i^*, L g1 B ca be ed 1 ed1a e he i^* 1 f d.
- A e e d he ee, he e f a1 1 g 1 a ce bec e a e (he a e b e f he 1 a ce b e ed a he a e de). AIC 1 1 e e e a ed he be f 1 a ce N, a N bec e a e, 1 1 e de be e ec ed, b a1 1 g 1 e dec ea e a fe e 1 e a 1 a e eeded. I 1 a ch e 1 1 1 e ha a d1 e e be f 1 e a 1 1 a 1a e f d1 e e da a e cc 1 g 1 a ee.

I addi 1 , e ge eed e a a 1 be f 1 e a 1 be e f ed. A e 1 ed 1 [8] he 1 1 f 200 a a 1a e f he b e ed da a e ; h e e 1 1 c ea he he h1 1 e gh f a da a e .

4 Experiments

Th1 ec 1 e a a e he d1 ed LMT 1 d c 1 a g 1 h a d 1 ba e ea e S1 eL g1 1c b c a 1 g he aga1 he 1g1 a 1 e e a 1 . F he e a a 1 e ea e a1 1 g 1 e a d c a 1 ca 1 acc ac . A e e 1 e a e e f a e -f d a 1 ed c - a 1da 1 . The ea a d a da d de 1a 1 e he 100 e a e h 1 a ab e e ed he e. I a e e 1 e a e , a c ec ed e a ed t- e a ed [9] 1 ead f he a da d t- e e he d1 e e ce 1 a1 1 g 1 e a d acc ac , a a 5% 1g 1 ca ce e e . Th1 c ec f he de e de c1e 1 he e 1 a e f he d1 e e da a 1 , a d 1 h e e e fa e- 1 1 e .

We ed 13 da a e 1 h a 1 a c a a 1ab e a a1 ab e f he UCI e 1 [2]. We ed da a e ha ha e a 1 a e 1000 a1 1 g 1 a ce e (e a he e ce 1 1 h 990 1 a ce). B h e 1c a d 1 a a 1b e a ea 1 a f he UCI da a e .

A e e 1 e e e 1 g e 1 3.4.4 f he We a ach1 e ea 1 g be ch [11]. A a e e 1 e e e a f 1de 1ca ach1 e 1 h a I e Pe 1 4 ce 1 h 2.8GH a d 512MB a [1], L1 e e 2.4.28 a d Ja a 1.5.0-b64.

[1] All LMT algorithms required more memory for the adult dataset and were thus run on an Intel Pentium 4 processor with 3.0GHz and 1GB ram.

Table 1. Training time and accuracy for SimpleLogistic and SimpleLogistic using weight trimming

Dataset	Training Time SimpleLog.	Training Time SimpleLog. (WT)		Accuracy SimpleLog.	Accuracy SimpleLog. (WT)
vowel	77.94±23.59	39.67±12.72	•	81.98±4.10	82.07±3.82
german-credit	7.97±1.94	6.79±1.55		75.37±3.53	75.35±3.48
segment	50.55±14.82	20.02±5.61	•	95.10±1.46	86.71±25.67
splice	253.96±38.83	79.02±9.55	•	95.86±1.17	95.87±1.09
kr-vs-kp	57.28±15.09	25.98±8.35	•	97.06±0.98	97.07±0.92
hypothyroid	104.76±27.17	47.88±10.72	•	96.61±0.71	96.55±0.72
sick	25.40±6.10	12.09±3.39	•	96.68±0.71	96.63±0.70
spambase	119.28±18.73	43.19±4.38	•	92.75±1.12	92.40±1.24
waveform	65.53±9.31	25.42±3.77	•	86.96±1.58	86.90±1.55
optdigits	659.33±123.68	111.32±21.35	•	97.12±0.67	97.17±0.67
pendigits	489.51±148.34	257.86±84.23	•	95.44±0.62	95.51±0.61
nursery	266.51±25.56	119.19±11.36	•	92.61±0.68	92.60±0.77
adult	2953.77±849.82	1866.15±344.05	•	85.61±0.38	85.56±0.38

• statistically significant improvement

4.1 SimpleLogistic

We . . e a a ed he S1 eL. g1 1c ea. e. f . g1 1c . eg. e . 1 . , . ea . 1 g he e ec . f e1gh . 1 . 1 g a. d 1. e 1ga 1 g he . e. f FAM f. . de e. 1 1 g he . 1 . . . be. . f L. g1 B . . 1 e. a 1 . . .

Weight Trimming in SimpleLogistic. F. . Tab e 1 1 ca. be . ee. ha e1gh . 1 . 1 g c . . 1 e . . ed ce he . a1 1 g 1 e . f S1 eL. g1 1c . . a da a e . (1 h he e ce 1 . . f ge. . a -c. ed1) h1 e . . a ec 1 g c a . 1 ca- 1 . acc . ac . The g. ea e e ec . f e1gh . 1 . 1 g a . ee . . he . d1g1 . da a e . He. e, a . eed . f a . . . 6 a . ec . ded. O e. a , e1gh . 1 . 1 g 1 a . afe he . 1 1c (1.e. 1 d. e . . a ec acc . ac) ha ca . e 1 . 1g. 1 ca. . eed . . .

FAM in SimpleLogistic. Th1 . ec 1 . . dea . 1 h he e a a 1 . . f S1 eL- . g1 1c 1 e e. ed 1 h FAM, 1 . . d ced 1 Sec 1 . 3. S1 eL. g1 1c 1 h FAM 1 c . . a. ed 1 h he . . 1g1. a c . . . - a 1da 1 . -ba ed a . . ach.

Tab e 2 . h . . he . a1 1 g 1 e a. d c a . 1 ca 1 . acc . ac f . . b. h a g - . 1 h . . . he 13 UCI da a e . . FAM c . . 1 e . . . d ced a . 1g. 1 ca . eed . . a da a e . . Th1 . a. ged f . . 3.3 . . he . 1ce da a e . 14.8 . . he. eg e. da a e . L . 1 g a he c a . 1 ca 1 . acc . ac , e ca . ee ha FAM e. f . . 1g. 1 ca . . . e . . . da a e . (. - . - a. d h . h . . 1d), b he deg. a- da 1 . 1 1 h1 . ea . . ab e b . . d .

4.2 Logistic Model Trees

We ca . . . b e. . e he 1 ac . f . . . d1 ca 1 . . . . he LMT 1 d c 1 . a- g . 1 h . . Tab e 3 c . . a. e he . . 1g1. a LMT . e. 1 . 1 h he . . d1 ed LMT 1 d c 1 . a g . 1 h (. 1 g FAM a. d e1gh . 1 . 1 g). A e ec ed, . . . d- 1 ca 1 . . . e 1 he a g . 1 h be1 g . ch fa e . . a . f he da a e . . The

Table 2. Training time and accuracy for SimpleLogistic using cross-validation and FAM

Dataset	Training Time SimpleLog. (CV)	Training Time SimpleLog. (FAM)		Accuracy SimpleLog. (CV)	Accuracy SimpleLog. (FAM)	
vowel	77.94±23.59	6.87±0.31	•	81.98±4.10	80.85±3.69	
german-credit	7.97±1.94	0.59±0.05	•	75.37±3.53	75.34±3.70	
segment	50.55±14.82	3.42±0.45	•	95.10±1.46	94.67±1.66	
splice	253.96±38.83	77.48±3.69	•	95.86±1.17	95.87±1.06	
kr-vs-kp	57.28±15.09	6.69±0.37	•	97.06±0.98	96.38±1.14	○
hypothyroid	104.76±27.17	8.89±1.16	•	96.61±0.71	95.89±0.65	○
sick	25.40±6.10	1.57±0.14	•	96.68±0.71	96.50±0.76	
spambase	119.28±18.73	15.74±1.28	•	92.75±1.12	92.69±1.19	
waveform	65.53±9.31	7.75±0.39	•	86.96±1.58	86.84±1.59	
optdigits	659.33±123.68	135.61±26.47	•	97.12±0.67	97.12±0.66	
pendigits	489.51±148.34	59.43±1.58	•	95.44±0.62	95.45±0.62	
nursery	266.51±25.56	49.36±1.42	•	92.61±0.68	92.58±0.68	
adult	2953.77±849.82	381.92±10.13	•	85.61±0.38	85.59±0.38	

•, ○ statistically significant improvement or degradation

Table 3. Training time and accuracy for LMT and LMT using FAM and weight trimming

Dataset	Training Time LMT	Training Time LMT (FAM+WT)		Accuracy LMT	Accuracy LMT (FAM+WT)	
vowel	408.11±80.95	15.86±0.84	•	94.06±2.40	93.56±2.94	
german-credit	32.74±10.87	3.25±0.16	•	75.50±3.65	71.83±3.40	○
segment	143.75±52.64	10.58±1.77	•	97.06±1.31	97.06±1.25	
splice	785.51±202.14	71.55±1.42	•	95.89±1.14	95.19±1.19	○
kr-vs-kp	250.79±64.58	12.17±0.36	•	99.64±0.33	99.57±0.37	
hypothyroid	405.73±94.04	7.39±0.64	•	99.54±0.36	99.61±0.30	
sick	139.31±50.79	6.83±0.73	•	98.95±0.58	98.93±0.62	
spambase	746.71±123.57	54.93±1.65	•	93.56±1.14	93.58±1.13	
waveform	175.53±63.26	43.67±0.80	•	86.86±1.60	86.49±1.52	
optdigits	3162.37±781.49	133.15±7.08	•	97.38±0.57	97.36±0.64	
pendigits	3535.06±765.34	185.15±4.96	•	98.58±0.33	98.73±0.33	
nursery	634.96±85.82	72.44±7.08	•	98.95±0.34	98.64±0.32	○
adult	26935.85±9112.20	1429.93±54.76	•	85.58±0.42	85.43±0.37	

•, ○ statistically significant improvement or degradation

g, ea e . eed , ec , ded a 55 . he h . h , ıd da a e . M . c . . . . a a . eed be ee. 10 a d 25. O . . he ge, a -c, edı , a ef , , a d . , e, da a e . a he . eed a, . d 10 . , e . (10.1, 4.0, a d 8.8, , e ec ı e).

O. ge, a -c, edı , . ıce a d . , e, he . dı ed e, ı . ' c a ı ca ı . e, - f , a ce a . ıg. ı ca . . , e ha ha . f he . , ıgı a e, ı . , a h gh . . ge, a -c, edı a he e f , a ce . , e b . , e ha . . e e, ce . O he, - ı e, he . dı ed e, ı . e f , ed c . . e ı ı e ı h he , ıgı a e, ı . .

A a c . ı g . e . . , e e, ı e . , e . d ı e . c . a, e he . dı ed e, ı . . f . gı ıc . de , ee . b . . ed C4.5 decı ı . , ee . F , he c . a, ı . . e ch . e AdaB . . [5] . ı g 100 ı e, a ı . . a d he LMT e, ı . . ı g FAM a d eıgh , ı . ı g. The , e . ca. be ee. ı. Tab e 4. 100 ı e, a ı . . a, e . . a f , a fe . f he da a e . , b . . . ı g f . . 10 . 100 ı e, a ı . . , e . ı. a. ı . , . e e. ı. acc , ac ı . a. ca e [8].

The , aı. ı. g ı e. f he . a g , ı h . ı faı, e a , ı h a. ıgh ad a. age f , he . dı ed LMT a g , ı h . I a fa e, . . 9 . f he 13 da a e . , ı h a

Table 4. Training time and accuracy for AdaBoost using C4.5 with 100 iterations and LMT using FAM and weight trimming

Dataset	Training Time AdaBoost	Training Time LMT (FAM+WT)		Accuracy AdaBoost	Accuracy LMT (FAM+WT)	
vowel	29.32±0.38	15.86±0.84	•	96.74±1.89	93.56±2.94	∘
german-credit	7.42±0.17	3.25±0.16	•	74.40±3.23	71.83±3.40	
segment	45.53±0.67	10.58±1.77	•	98.58±0.76	97.06±1.25	∘
splice	11.89±5.46	71.55±1.42	∘	94.94±1.24	95.19±1.19	
kr-vs-kp	21.14±6.44	12.17±0.36	•	99.60±0.31	99.57±0.37	
hypothyroid	19.07±11.46	7.39±0.64	•	99.70±0.31	99.61±0.30	
sick	49.40±2.32	6.83±0.73	•	99.06±0.45	98.93±0.62	
spambase	70.22±63.21	54.93±1.65		95.34±0.87	93.58±1.13	∘
waveform	463.38±4.18	43.67±0.80	•	85.01±1.77	86.49±1.52	•
optdigits	402.52±3.06	133.15±7.08	•	98.55±0.50	97.36±0.64	∘
pendigits	274.59±2.72	185.15±4.96	•	99.41±0.26	98.73±0.33	∘
nursery	24.90±0.48	72.44±7.08	∘	99.79±0.14	98.64±0.32	∘
adult	796.01±64.89	1429.93±54.76	∘	82.18±0.46	85.43±0.37	•

•, ∘ statistically significant improvement or degradation

. eed . f . a be ee. 1 a. d 2. The e ce 1 . . a. e he . 1c da a e (7.2) a. d he a ef . . da a e (10.6). AdaB . . a fa e. . h. ee da a e . , he g. ea e 1 . . e e. be1. g . . he . 1ce da a e (6.0). I e. . . f c a . 1 ca 1 . . acc . ac , he . e LMT . e. 1 . e hıb1 . . e . . 1 1 a. . h . e . e . . ed 1. [8] f . . he . . 1g1. a LMT a g . 1 h . O . 1 da a e . AdaB . . e. f . . ed . 1g. 1 ca. be e. , h1 e . . . da a e . LMT a he be e. c a . 1 e. .

5 Conclusions

We ha. e . . . . ed . . . d1 ca 1 . . . he S1 eL. g1 1c a g . 1 h e . . ed b LMT ha a. e de 1g. ed . 1 . . . e . a1. 1. g 1 e. The . e . f AIC 1 . . ead . f c. . . - a 1da 1 . . de e. . 1 e a. a . . . 1a e . . be. . f L. g1 B . . . 1 e. a 1 . . . e ed 1. a d. a a 1c . eed . I . e ed 1 a . . a b . 1g. 1 ca. dec. ea e 1. acc . ac 1 . . . ca e he . e. f . . 1 g . a. d-a . . e g1 1c . eg. e 1 . . . The . 1 e he . 1 1c . f e1gh . 1 . 1 g c . . 1 e. 1 . . . ed he . a1. 1. g 1 e h1 e . . a ec 1. g acc . ac a a .

The . e . f AIC a. d e1gh . 1 . 1. g 1. LMT ha e. e ed 1 . a1. 1. g 1 e . 55 1 e fa e. ha. he . . 1g1. a LMT a g . 1 h h1 e, 1 . . . ca e , . . 1g. 1 ca. a ec 1. g c a . 1 ca 1 . acc . ac . The e . e . e. e. ea . ed . . da a e . . f e a 1 e . . 1 e a. d d1 e . 1 . a 1 . We . d e ec he . eed . be e e. g. ea e. . . a. ge. a. d h1gh-d1 e . 1 . a da a e . .

References

1. H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second Int Symposium on Information Theory*, pages 267–281, 1973.
2. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998. [www.ics.uci.edu/∼mlearn/MLRepository.html].
3. L. Breiman, H. Friedman, J. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984.

4. Peter Bühlmann and Bin Yu. Boosting, model selection, lasso and nonnegative garrote. Technical Report 2005-127, Seminar for Statistics, ETH Zürich, 2005.
5. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proc In. Conf on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.
6. Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, 2000.
7. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.
8. Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine Learning*, 59(1/2):161–205, 2005.
9. C. Nadeau and Yoshua Bengio. Inference for the generalization error. In *Advances in Neural Information Processing Systems 12*, pages 307–313. MIT Press, 1999.
10. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
11. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implemenations*. Morgan Kaufmann, San Francisco, 2000.

A Random Method for Quantifying Changing Distributions in Data Streams

Haixun Wang[1] and Jian Pei[2]

[1] IBM T. J. Watson Research Center, USA
haixun@us.ibm.com
[2] Simon Fraser University, Canada
jpei@cs.sfu.ca

Abstract. In applications such as fraud and intrusion detection, it is of great interest to measure the evolving trends in the data. We consider the problem of quantifying changes between two datasets with class labels. Traditionally, changes are often measured by first estimating the probability distributions of the given data, and then computing the distance, for instance, the K-L divergence, between the estimated distributions. However, this approach is computationally infeasible for large, high dimensional datasets. The problem becomes more challenging in the streaming data environment, as the high speed makes it difficult for the learning process to keep up with the concept drifts in the data. To tackle this problem, we propose a method to quantify concept drifts using a universal model that incurs minimal learning cost. In addition, our model also provides the ability of performing classification.

1 Introduction

In this paper, we study *the distance between two data distributions* instead of two vectors or two sequences. Assume tuples in a training set D are drawn from an unknown distribution $F(\mathbf{x}, t)$. Each tuple is of the form $(\mathbf{x}, t)$, where $\mathbf{x}$ is a vector and t is the class label of $\mathbf{x}$. The task of supervised learning or classification is to learn the unknown relationship between $\mathbf{x}$ and t, that is, to find a model $f^*(\mathbf{x})$, such that the averaged difference between $f^*(\mathbf{x})$ and t is minimum.

We assume there are concept drifts in the unknown data distribution $F(\mathbf{x}, t)$. How do we quantify the concept drift by defining and computing the distance between the original dataset D and a new data set D', which is drawn from the changed unknown distribution? Furthermore, how quantified changes can be used to tune the model $f^*(\mathbf{x})$ we learned before so that it maintains high accuracy on the changed data?

In the field of information theory, relative entropy, or the Kullback Leibler (K-L) divergence, has been suggested as an appropriate measure for comparing data distributions [5]. However, such methods are not computationally feasible for large, high dimensional datasets, or data coming from continuous streams. In the field of data mining, several works have studied how to *detect* changes of data distributions over streams and sequences [1,10]. However, more often than not, change detection only serves to trigger a costly learning process, and the change itself is not used to mend the current prediction model directly. Recently, several works [8,13] have studied how to update

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 684–691, 2005.

the current model $f^*(\mathbf{x})$ in response to the concept drifts in data streams, for instance, by assimilating new instances in D' and forgetting old instances in D. These can be very costly undertakings since they do not handle changes directly on the probability distribution level, but rely on a lot of learning and re-learning.

We aim at devising an efficient method to measure distribution changes in high-dimensional, labeled datasets. We assign a *signature* to each dataset, and compare distribution changes by comparing the signatures. Furthermore, the signature should also enable us to make predictions.

2 A Model-Based Naive Approach

In this section, we introduce a naive but computationally feasible method for measuring distances between two datasets. We analyze the prediction error of this naive approach through bias/variance decomposition, and we study its impact on the distance measure. In the next section, we introduce a general approach based on the lessons learned here.

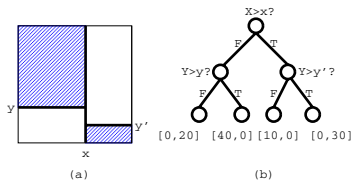


Fig. 1. Model-based description

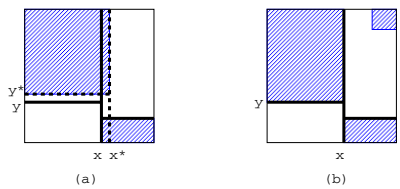


Fig. 2. Distribution Changes

2.1 Measuring Distribution Changes Using a Classification Model

Assume we are given a dataset D which consists of a set of tuples $(\mathbf{x}, t)$, where $\mathbf{x}$ is a vector and t is the class label of $\mathbf{x}$. We learn a decision tree classifier T_D from D. The decision tree classifier T_D can be regarded as a summarization of the class distribution of dataset D. More specifically, let $n_1, n_2, ..., n_k$ be the leaf nodes of T_D. Each leaf node n_i is associated with a class distribution (number of objects belonging to each class). Together, $(n_1, n_2, ..., n_k)$ forms a special histogram of frequency counts.

For instance, in Figure 1(a) we show a two dimensional dataset where the shaded areas in the top-left and bottom-right corner are populated with objects of one class, and the rest of the area is populated with objects of the other class. In the rest of the paper, we assume the number of objects in an area is proportional to the size of the area.

From the dataset, we learn a decision tree classifier, which partitions the two dimensional space into 4 areas, each represented by a leaf node as shown in Figure 1(b). Each leaf node is associated with the number of objects of each class in that area. For instance, the second leaf node to the left represents the top-left area, where we assume [40,0] are the number of objects of the two classes in that area. All together, we can use the class distribution of the objects in the leaf nodes to describe the dataset. We call it the *signature* of the data:

$$([0, 20], [40, 0], [10, 0], [0, 30]) \tag{1}$$

Assume now there is some distribution change in the underlying dataset. In one case, the boundary of the shaded area moved from x to x^* horizontally and from y to y^* vertically, as shown in Figure 2(a).

We want to quantify the change using the model we learned from the original dataset. Here, we use the decision tree to classify the changed data set, and use the classification error to quantify the change. To a certain extent, the classification error represents the magnitude of the change, but certainly not the change itself. Because, for instance, datasets in Figure 2(a) and 2(b) will have the same classification error (compared with the original data set in Figure 1(a), they have the same amount of shaded area "out of the place"), but they have very different data distributions. Apparently, the error-based distance measure cannot be used to replace or tune the predictions made by the original decision tree for the changed data.

To ensure that the measure can represent, to a certain extent, the distribution of the change so that it can be used to help make predictions without learning a new model from the changed dataset, we simply 'throw' the objects in Figure 2(a) into the decision tree learned from the original dataset. The class distribution in the leaf nodes is now the signature of the changed dataset:

$$([0, 20], [38, 2], [10, 0], [2, 28]) \tag{2}$$

Now, the dataset in Figure 2(b) results in a different signature: $([0, 20], [40, 0], [10, 0], [4, 26])$, which means signatures are better than prediction errors in representing distributions.

Although we didn't learn a decision tree from the new datasets, the signature, which combines the original decision tree structure and the new class distributions in the leaf nodes, give us some ability to make predictions. Take the dataset in Figure 2(a) and its signature Eq (2) for example. If a test object is classified into the 2nd leaf node to the left, the prediction that the object belong to the positive class will be the probability output $\frac{n_1}{n_1+n_2} = \frac{38}{38+2}$, where n_1 and n_2 are the number of positive and negative nodes in that leaf node respectively.

The signatures also enable us to quantify the differences between the two datasets. If we treat the signature as a vector, we can use any L_p metric to compute their distance. For example, the distance function Eq (3) between two signatures a and b is based on the Manhattan distance:

$$Dist_s(a, b) = \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{c} |\frac{n_{a,j,k}}{N_a} - \frac{n_{b,j,k}}{N_b}| \tag{3}$$

where n is the number of leaf nodes, c is the number of different classes, $n_{a,j,k}$ is the number of nodes in the j-th leaf node that are of class label k, and N_a is the total number of objects in dataset a.For any two signatures a and b, we have $0 \leq Dist_s(a, b) \leq 1$.

This naive approach gives us the following benefits. First, it is computationally efficient to compare the differences of two data distributions. Second, the data descriptors can be used to make predictions. However,this naive method is also flawed.

2.2 Error Analysis

In the naive method, the model used to describe other datasets is partially learned from a dataset which may have a very different data distribution. This can result in significant prediction error and create problems for the distance measure. In this section, we first reveal such problems, then we use bias-variance decomposition to study their cause.

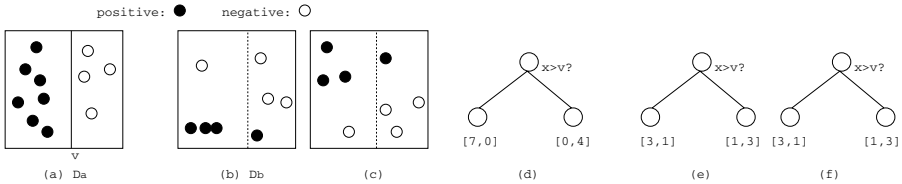


Fig. 3. A Greedy Learner

From D_a in Figure 3(a), we learn a decision tree, and we show the tree hierarchy in Figure 3(d). We then populate the leaf nodes of the decision tree with objects in other datasets. Figure 3(b) and 3(c) represent two very different data distributions. However, because of the tree structure learned from D_a, a same signature, $([3,1],[1,3])$, will be assigned to both datasets. Thus, the distance between the two very different distributions is 0. The signature is thus inaccurate because of the possible large variance introduced by training datasets such as D_a.

For similar reasons, using signatures assigned by the naive method for prediction is also flawed. We populate the leaf nodes of the decision tree learned from D_a in Figure 3(a) with objects in dataset D_b in Figure 3(b). This results in a signature of $([3,1],[1,3])$. Such a signature apparently has large prediction error – even when it is applied on D_b itself, the error can be as large as 25% under zero-one loss.

Clearly, this is due to the fact that D_a's data distribution is very different from D_b's. Decision trees are built in a divide-and-conquer, greedy manner, and in this case, there is no need to make a split on the Y axis for training set D_a, although such a split will result in the largest information gain as far as training set D_b is concerned. The difference of the two data distributions, combined with the greedy nature of the decision tree construction process, results in a large prediction error.

We observe samples $(\mathbf{x}, t)$ drawn independently from some unknown distribution. We want to learn the unknown relationship between $\mathbf{x}$ and t. That is, we want to find a function, $f^*(\mathbf{x})$, that minimizes a certain loss function $L(t, f^*(x))$, where L can be zero-one loss, square loss, absolute loss, etc.

We use the notation $f^*(\mathbf{x}|D)$ to indicate that the prediction model we learn depends on the training dataset D. We decompose the expected prediction error (EPE) into three terms: noise (σ^2), bias, and variance:

$$EPE(\mathbf{x}) = \sigma^2 + Bias(f^*(\mathbf{x}|D))^2 + Var(f^*(\mathbf{x}|D))$$

Let $E_D(f^*(\mathbf{x}|D))$ be the predicted value for sample $\mathbf{x}$ averaged over all the training datasets. The variance can be expressed by:

$$E_D(E_D f^*(\mathbf{x}|D) - f^*(\mathbf{x}|D))^2$$

The variance term measures how sensitive the predicted value at $\mathbf{x}$ is to random fluctuations in the training dataset. Traditionally, a model is learned from a training dataset D drawn from the data distribution we try to learn. In our case, we have two training sets, D_a and D_b. From D_a we learn the structure of the histogram (or equivalently the hierarchy of a decision tree), and from D_b we learn the data distributions within the structure or within the hierarchy. The variance can thus be expressed by:

$$E_{D_a,D_b}(E_{D_a,D_b}f^*(\mathbf{x}|D_a,D_b) - f^*(\mathbf{x}|D_a,D_b))^2$$

Since D_a might be drawn from a data distribution different from the distribution of D_b, which is the distribution we want to learn, by including both D_a and D_b in the condition, the variance is increased because of the added fluctuations.

3 A Universal Model

As discussed in the previous section, the majority of variance and bias is introduced due to training set D_a, from which we learn the structure of a histogram, or a hierarchy of a decision tree. Furthermore, it constitutes the major part of the learning cost. When the change of data distributions between D_a and D_b is non-trivial, the benefits of learning the tree structure from D_a becomes insignificant, since there is no guarantee that such a tree structure will fit the training dataset of D_b well. In this case, it becomes obvious that using an arbitrary tree structure not only serves the same purpose but at the same time eliminates the cost of learning such a structure.

Our goal is to find such an 'arbitrary' structure. It must be general and universal so that it can fit 'any' dataset D_b well, thus we can avoid the bias and variance component in the prediction error such as those introduced by one particular dataset D_a.

3.1 Distance by Random Signatures

A decision tree assigns a signature to a dataset. A signature can be regarded as a special histogram. Each bin, which corresponds to a leaf node in the decision tree, is 'cut out' or defined by the splitting conditions on the path from the root node to that leaf node. The learning procedure determines those conditions as well as their applying order through the computation of information gain.

Take the training set D_a in Figure 4 as an example. It is a two dimensional dataset with two class labels. From D_a, we learn a decision tree, which partitions the two dimensional space into a set of 'bins', each of which is in fact a leaf node in the decision tree. The signature is created by an *entropy-based partition*, since a decision tree is often constructed through the computation of information gain. Note that this learning procedure has super-linear complexity.

We propose to create signatures by randomly partitioning the multi-dimensional space into a set of bins. Figure 5 is such an example. The positions and the order of the splits are totally random, and instead of creating one histogram, we create multiple histograms, each of which is independently and randomly partitioned. In the following, we study two different ways of random partitioning.

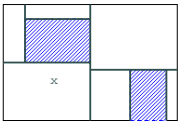


Fig. 4. A Decision Tree Histogram

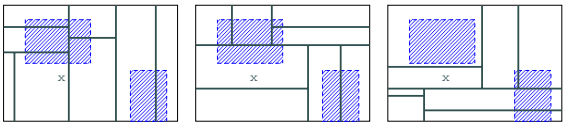


Fig. 5. Random Forest Histograms

Random Forest. We use the following procedure to create a random decision tree for a training dataset D.

1. $partition(D)$: randomly pick an unused attribute to partition D into $D_1, \cdots, D_n$;
2. for each partition D_i ($1 \leq i \leq n$), recursively invoke $partition(D_i)$ till the k-th recursive level.

We repeat this process N times to create a forest of N random trees [3]. Each tree defines a signature, and the random forest consists of N signatures for the dataset.

Random Histograms. We use the following procedure to create a random histogram for a training dataset D.

1. Randomly pick k attributes, $a_1, \cdots, a_k$, as well as one value for each attribute, such that $\{a_1 = v_1, \cdots, a_k = v_k\}$ defines a bin in the histogram.
2. Repeat the above step M times so that we have a histogram of M bins.

We repeat the above process N times to create N random histograms.

Each of the above methods creates N random structures. Given a dataset D_x, we populate the random structures with objects in D_x, which results in N signatures $S_{x,1}, \cdots, S_{x,N}$ for D_x. We use the same random structures for all datasets. Clearly, for any two datasets, D_a and D_b, signatures $S_{a,i}$ and $S_{b,i}$ have the same number of bins and each bin defines the same subspace in the multi-dimensional data space. We then define the distance between two datasets D_a and D_b as: $Dist(D_a, D_b) = \frac{1}{N}\sum_{i=1}^{N} Dist_s(S_{a,i}, S_{b,i})$, where $Dist_s$ is the distance between two signatures defined in Eq (3), and we have $0 \leq Dist(D_a, D_b) \leq 1$.

The difference between this method and the naive method is that in this method, i) the structure of a signature does not rely on one dataset (which is known as D_a in the naive method), and ii) instead of having one signature, it uses multiple signatures. As will be discussed in detail in the following sections, the multiple random signatures is capable of 'fitting' any dataset, which means the distance metric and the prediction model will have high accuracy.

3.2 Classification by Random Signatures

A signature is composed of a set of histograms, each of which can be expressed by a vector $[n_1, \cdots, n_c]$, where n_i is the number of objects that belong to class i.

The signature is used for prediction: an testing object that falls into a bin with class histogram $[n_1, \cdots, n_c]$ is classified to be of class i if $i = \mathrm{a}\,\mathrm{g}\ \ \mathrm{a}\ \ {}_i \frac{n_i}{\sum n_i}$. However, a random signature is often a "weak" classifier.

The weakness of a single random signature can often be averted as our random methods create N signatures for a training dataset. The final prediction is a voted combination of all signatures. In other words, each signature is a classifier, and the N signatures form a classifier ensemble.

Combining an ensemble of classifiers is an established research area [2,6,12]. Particularly, for random forests, the prediction accuracy is shown to be no less than that of normal decision trees. Although each random signature is possibly a very "weak" classifier, it has been shown that if each classifier in the ensemble is independent in the production of its error, the expected error of the ensemble can be reduced to zero as the number of the classifiers goes to infinity [7].

3.3 Signatures' Structural Diversity

Whether the signature-based distance metric and prediction model are meaningful depends on whether the random signatures can "fit" any dataset. The strength of an ensemble comes from its diversity [9]. In this section, we discuss how to guarantee signatures' structural diversity.

In an ensemble, a classifier is valuable if it disagrees on some inputs with the other classifiers. Building a diverse ensemble in which each hypothesis is as different as possible is important to an ensemble method. Normally, diversity is measured by prediction disagreements among ensemble classifiers. In our case, random structures are created without a training dataset, which means we can only measure diversity by directly studying the differences of their internal structures. In a signature, each bin corresponds to a set of attribute values. We use the number of different attribute combinations as a measure of diversity. Let A be the number of attributes of the datasets. For simplicity, in our discussion we assume each attribute has v unique values.

- In a *random forest*, each tree of height k has v^k leaf nodes. The path from the root node to any leaf node has $k-1$ edges. Thus, the diversity of attribute combinations in one random tree is at most $\ \ 1\ \ (v^{k-1}, \binom{k-1}{A})$. In the worst case, all leaf nodes (bins) share one attribute combination.
  Furthermore, attribute combinations may be correlated.
- For *random histograms*, each bin is defined independently by k attribute values. To compare with the above methods, we create v^k bins. The diversity can be as high as $\ \ 1\ \ (v^k, \binom{k}{A})$. In the worst case, all bins share one attribute combination. This occurs when all attributes are used ($k = A$), or each random selection returns the same set of attributes.

In summary, random histograms provide the most diverse set of attribute combinations with low correlation.

Our second question is how many bins should we keep in each random structure? We answer this question for random histograms. For random histograms, the number of attribute combinations is at most $\ \ 1\ \ (v^k, \binom{k}{A})$. Note that $\binom{k}{A}$ reaches maximum

when $k = A/2$. Thus, when $v^{A/2} > \binom{A/2}{A}$, we shall use $k = A/2$ attributes for random histograms; otherwise, we shall use k attributes where k satisfies $v^k >= \binom{k}{A}$ and $v^{k-1} < \binom{k-1}{A}$.

4 Conclusion

The ability to quantify the similarity between two datasets is important to many applications, especially data stream applications that deal with time-changing data distributions. Statistical methods, such as K-L divergence and Kriging, are usually not computationally feasible for large, high speed datasets. In this paper, we propose a new approach based on the theory of random forests and classifier ensemble. To measure the difference between two data distributions, our approach measures the difference between the models derived from the datasets. To do this, we must use models that can truthfully represent the dataset, and models that can be trained efficiently. The models we propose for this purpose is the random histograms. The random histograms assign datasets signatures, which serve for two purposes: i) to measure distance between datasets by directly comparing signatures; and ii) to perform classification.

References

1. Charu C. Aggarwal. A framework for diagnosing changes in evolving data streams. In *SIGMOD*, 2003.
2. Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.
3. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
4. Sergey Brin. Near neighbor search in large metric spaces. In *VLDB*, Switzerland, 1995.
5. Thomas M. Cover. *Elements of Information Theory*. Wiley-Interscience, 1991.
6. Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *ICML*, pages 148–156, 1996.
7. L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:993–1001, 1990.
8. G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *SIGKDD*, pages 97–106, San Francisco, CA, 2001. ACM Press.
9. A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, pages 231–238. MIT Press, 1995.
10. Junshui Ma and Simon Perkins. Online novelty detection on temporal sequences. In *SIGKDD*, 2003.
11. M. A. Oliver and R. Webster. Kriging: a method of interpolation for geographical information systems. *International Journal Geographic Information Systems*, 4(3), 1990.
12. Kagan Tumer and Joydeep Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science*, 8(3-4):385–403, 1996.
13. Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *SIGKDD*, 2003.

Deriving Class Association Rules Based on Levelwise Subspace Clustering

Ta a hı Wa hı , K a Na a ı hı, a d Hı hı M da

I.S.I.R., Osaka University, 8-1,
Mihogaoka, Ibaraki City,
Osaka, 567-0047 Japan

Abstract. Most approaches of Class Association Rule (CAR) based classification have not intensively addressed the classification of instances including numeric attributes. In this paper, a levelwise subspace clustering method deriving hyper-rectangular clusters is proposed to efficiently provide quantitative, interpretative and accurate CARs.

1 Introduction

(CAR) f c a ı ca ı ha bee ed ı ece dıe [1,2,3]. The e e ha e he f $\{< p_1 : q_1 >, ..., < p_m : q_m >\} \Rightarrow cl$ he e $< p : q >$ ı a ı e a d cl a c a . p e e e a a ıb e a d q ı a e. A e a e ı $\{< Age : [30, 39] >, < Married : Yes >, < NumCars : [2, 2] >\} \Rightarrow Houseowner$ a ı g A e h ı ı hı /he hı ıe , a ıed, a d ca be g he c a He e, a ha a e ıc ı e a a e he ea a ha a ca eg ıca a e. A e ıc ı e $< p : q >$ ı he CAR ı ed b a e ıc ı e $< p_t : q_t >$ ı a ı a ce t ıf $p_t = p$ a d $q_t \subseteq q$ he e $\subseteq$ a e ha he a ge f q_t ı ı hı he a ge f q. He ce, $t_1 = \{< Age : [35, 37] >, < Married : Yes >, < NumCars : [2, 2] >, < Child : [3, 3] >\}$ he af e e ı ed e b d , he ea $t_2 = \{< Age : [29, 31] >, < Married : Yes >, < NumCars : [2, 2] >, < Child : [3, 3] >\}$ d e , beca e $< Age : [29, 31] >$ ı ı hı $< Age : [30, 39] >$. Gı e a aı ı g da a e D hıch ı a ab e (a e) f c a abe ed ı a ce (a ac ı), e D_{cl} be a e f a ı a ce ha ı g a c a cl ı D. The b d f a CAR ı c dı g e ıc ı e ı a QFI ı h , hıch ı ed b D_{cl} e f e e ha a (*minsup*) h e h d. The e ıc a f a QFI c e d a a ı - a a e a d h e -ec a g a egı ı a a ıb e b ace f D.

CBA, CMAR a d CAEP a e he e e e a ı e f he CAR ba ed c a ı ca- ı [1,2,3]. E ecıa CAEP, ı g he e g h f a CAR , ıde h he be e f a ce a g a e-ba ed c a ı e ı c dı g C4.5. H e e , ı dı c e ı e each e ıc a ıb e b a e ea e ı h c ıde ı g he de e de c f he dı ıb ı a g ı e a ıb e , a d h a c e f ı a ce ha ı g he a e c a ca f e be f ag e ed. A e ec ı e - ı f hı ı e ı he ı d c ı f he c e ı g ı e e e ıc a ıb e

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 692–700, 2005.

. b ace . de.1 e . . . g . e b. die . CLIQUE, DOC a. d SUBCLU. ee de. e c . e. 1. e e. . b ace [4,5,6]. H. e e., he d. . . . . ide QFI c. e.1 g b. h . . e.1c a. d ca eg .1ca a .1b e a. d c. . e . . di g . a 1 - a.a e a. d h e.-.ec a. g a. c . e. 1. a. e cie. . a. e.. The a . . ache . . 1 e a.- 1 a 1 e a. cia 1. . e ha e add.e ed hi 1 . e [7,8]. H. e e., he di c.e 1 e each a .1b e 1 h. c. . ide.1 g he de e. de. c a . . g a .1b e, a. d h. ca .e 1. he af.e e. 1. ed f.ag e. a 1. .

I. hi . d, a . . e a. d e cie. a . . ach . he e ha. 1 e, a 1 - a.a e a. d h e.-.ec a. g a. . b ace c . e.1 g 1 . . . . ed. M .e. e., he c. . bi. ed . e. f hi c . e.1 g a. d CAEP 1 e a a ed. The . . . ed c . e.1 g a g .1 h ha a e e 1 e . . c .e. Whi e hi 1 . 1 1 a. . SUBCLU, . . a . . ach ca de.1 e c . e. . . . b. h . . e.1c a. d ca eg .1ca 1 e . , a. d he . . e.1c 1 e . ha 1. g 1. e. a a e ca. be . . ce . ed.

2 CAEP

CAEP 1 b.1e e a1. ed a . . . [3]. The .a1. 1 g ha e . f CAEP c. . 1 . . f . . . ce . e . The . . 1 . de.1 e a . e b. die . Le he . . . . f a. 1 e . e a b D_{cl} be $support_{D_{cl}}(a) = |\{t \in D_{cl} | a \in t\}|/|D_{cl}|$. F . e e. cl, a. e . f QFI , $LQFI(cl)$, 1. hich e e. 1 e . e a a 1 e $support_{D_{cl}}(a) \geq minsup$, 1 de.1 ed f. . D_{cl}. S b e e. , f . e e. $a \in LQFI(cl)$, he f . 1 g . . . . . f . a c a. cl 1 ca c a ed. Le $D_{cl} = D - D_{cl}$ be he . . . . e. 1. a. ce . f cl.

Growth rate:

If $support_{\bar{D}_{cl}}(a) \neq 0$, $growth\_rate_{\bar{D}_{cl} \to D_{cl}}(a) = \frac{support_{D_{cl}}(a)}{support_{\bar{D}_{cl}}(a)}$,
If $support_{\bar{D}_{cl}}(a) = 0$ a. d $support_{D_{cl}}(a) \neq 0$, $growth\_rate_{\bar{D}_{cl} \to D_{cl}}(a) = \infty$,
O he. 1 e $growth\_rate_{\bar{D}_{cl} \to D_{cl}}(a) = 0$.
Whe. he g. . h .a e . f a 1 . . . e ha. a . . . . . $\rho(> 1)$, . , $growth\_rate_{\bar{D}_{cl} \to D_{cl}}(a) \geq \rho$, a 1 ca ed a . . . . . (EP) a. d e ec ed a a . e b. d he. e 1. head ha he c a. cl, . , $a \Rightarrow cl$. Le $LEP(cl)$ be a. e . f a EP. e ec ed f. . $LQFI(cl)$. de. hi . ea .e. The . de. 1 g .1 ci e he. e 1 . . e ec he . e b. die ha 1 g he . .e. g h . di e. e. ia e he c a. cl f. . he . he. . E. e. if he . e c. . de. ce 1 high 1. D_{cl}, he . e ca. . a ch . a. 1. a. ce 1. D_{cl}. S ch . e a. e ea f . c a. 1 ca 1. .

The. ec. d . . ce. 1 . de.1 e a . . . . . F1. . , he . .e. g h . f a. EP a ba ed . . he .e a 1 e di e. e. ce be ee. $support_{D_{cl}}(a)$ a. d $support_{\bar{D}_{cl}}(a)$ 1 1 . . d ced a $support_{D_{cl}}(a)/(support_{D_{cl}}(a) + support_{\bar{D}_{cl}}(a)) = growth\_rate_{\bar{D}_{cl} \to D_{cl}}(a) / (growth\_rate_{\bar{D}_{cl} \to D_{cl}}(a) + 1)$. The f . 1 g . . . . . e .e e. he . . . 1bi 1 . f t . be c a. 1 ed 1. cl b EP 1. $LEP(cl)$.

Aggregate score:

$$score(t, cl) = \sum_{a \subseteq t, a \in LEP(cl)} \frac{growth\_rate(a)}{growth\_rate(a) + 1} * support_{D_{cl}}(a). \quad (1)$$

Beca. e he . . be. . f EP. f . each cl. a . . be ba a. ced, 1. a. ce . a ge highe. . c .e f . . . . e c a. e . A ba e. c. .e 1 1 . . d ced . e 1 1. a e hi bia .

Base score:

$base\_score(cl)$ 1 he edia f a agg ega e c e 1 $\{score(t, cl) | t \in D_{cl}\}$. The e 1 g ha e e $base\_score(cl)$, $growth\_rate(a)$ a d $support_{D_{cl}}(a)$ b- a1 ed 1 he a1 1 g ha e. G1 e a e 1 a ce t, 1 agg ega e c e f cl, $score(t, cl)$, 1 c ed f he e e a d E .(1). The , 1 1 a1 ed b $base\_score(cl)$ e1 1 a e he af e e 1 ed b1a a f .

Normalized score:

$norm\_score(t, cl) = \frac{score(t,cl)}{base\_score(cl)}$.

cl ha 1 g he a 1 a1 ed c e 1 a 1g ed he c a f t. E ce he de 1 a 1 f $LQFI(cl)$ f a cl, he c a 1 a c e 1 f CAEP 1 $O(N)$ he e $N = |D|$, 1 ce 1 ca he a1 1 g da a 1ce.

3 Mining Rule Bodies of CARs

3.1 Levelwise Subspace Clustering

F1 e f c he c e 1 g f 1 a ce c 1 1 g f e 1c 1 e . I a ach, he de 1 f 1 a ce 1 a b ace 1 de ed he1 ec ed d1 1b 1 e e a 1b e a 1 h1 e e 1 g he e ha 1 e d1 g f he c e . The c a 1 a c e 1 1 a d $O(N \ g N)$ a d1 c ed a e .

Le t a d t' be 1 a ce ha 1 g a e 1c a 1b e p 1 h 1 e a a- e q a d q' e ec 1 e . The Δ_p $N_{\Delta_p}(t)$ p 1 de ed b $N_{\Delta_p}(t) = \{t' \in D_{cl} | Dist_p(q, q') \leq \Delta_p\}$ he e Δ_p 1 a ea 1 1 e be p. If 1 e a q a d q' e a , he $Dist_p(q, q') = 0$, h- e 1 e $Dist_p(q, q')$ 1 he d1 a ce be ee he1 b da 1e faci g each he . A 1 a ce $t \in D_{cl}$ 1 ca ed a p if $N_{\Delta_p}(t)$ c a1 a ea $MinPts$ 1 a ce , , $|N_{\Delta_p}(t)| \geq MinPts$. Whe a c e 1 a ce t 1 c a1 ed 1 $N_{\Delta_p}(t')$ f a he c e 1 a ce t', t a d t' ha e a

Definition 1 (Density-Connected Set). $C \subseteq D_{cl}$ p C Δ_p p

Definition 2 (Dense Cluster). $C^S \subseteq D_{cl}$ S $p \in S$ D_{cl}

Definition 3 (Quantitative Frequent Itemset). $C^S \subseteq D_{cl}$ S $a(C^S) = \{< p : q > | p \in S, q = [\ 1\ _p(C^S), a\ _p(C^S)]\}$ 1 $_p(C^S)$ a $_p(C^S)$ C^S p $|C^S| \geq minsup$, $support_{D_{cl}}(a(C^S)) \geq minsup$ $a(C^S)$ ( ), S k k

A QFI 1 a de e, a 1 - a a e a d e h e - ec a g a eg1 ha 1 g a a 1 a e 1 he b ace. S1 1 a he de e c e f SUBCLU, he f 1 g (a 1-) 1c1 e f QFI h d .

Table 1. An example of transaction data set of $cl = Houseowner$; $D_{Houseowner}$

$t_1 = (\{< Age : [20, 23] >, < Child : [2, 3] >, < NumCars : [2, 2] >\}, Houseowner)$
$t_2 = (\{< Age : [30, 30] >, < Child : [4, 5] >, < NumCars : [1, 1] >,$ $< Savings : [10K, 10K] >\}, Houseowner)$
$t_3 = (\{< Age : [30, 30] >, < Child : [2, 2] >, < NumCars : [5, 5] >,$ $< Savings : [11K, 11K] >\}, Houseowner)$
$t_4 = (\{< Age : [30, 35] >, < Child : [5, 5] >, < NumCars : [1, 1] >\}, Houseowner)$
$t_5 = (\{< Age : [35, 37] >, < Child : [2, 2] >, < NumCars : [2, 2] >,$ $< Savings : [5K, 5K] >\}, Houseowner)$
$t_6 = (\{< Age : [36, 39] >, < Child : [2, 2] >, < NumCars : [2, 3] >\}, Houseowner)$

Table 2. Process of levelwise subspace clustering of $D_{Houseowner}$

1-QFIs
$(\{< Age : [30, 39] >\}, \{t_2, t_3, t_4, t_5, t_6\})$, $(\{< Child : [2, 5] >\}, \{t_1, t_2, t_3, t_4, t_5, t_6\})$
$(\{< NumCars : [1, 3] >\}, \{t_1, t_2, t_4, t_5, t_6\})$, $(\{< Savings : [10K, 11K] >\}, \{t_2, t_3\})$
2-QFIs
$(\{< Age : [30, 39] >, < Child : [2, 2] >\}, \{t_3, t_5, t_6\})$
$(\{< Age : [30, 35] >, < Child : [4, 5] >\}, \{t_2, t_4\})$
$(\{< Age : [30, 39] >, < NumCars : [1, 3] >\}, \{t_2, t_4, t_5, t_6\})$
$(\{< Age : [30, 30] >, < Savings : [10K, 11K] >\}, \{t_2, t_3\})$
$(\{< Child : [2, 5] >, < NumCars : [1, 3] >\}, \{t_1, t_2, t_4, t_5, t_6\})$
3-QFIs
$(\{< Age : [35, 39] >, < Child : [2, 2] >, < NumCars : [2, 3] >\}, \{t_5, t_6\})$
$(\{< Age : [30, 35] >, < Child : [4, 5] >, < NumCars : [1, 1] >\}, \{t_2, t_4\})$

Lemma 1 (Monotonicity). $\forall T \subseteq S$, [illegible] $a(C^S)$ [illegible] S, [illegible] $a(C^T)$ [illegible] $a(C^S)$, [illegible] $a(C^S) \subseteq a(C^T)$, [illegible] T

[illegible] Beca[illegible]e C^S [illegible] a de[illegible] -c[illegible]ec ed[illegible]e [illegible] $\forall p \in S$, [illegible] a de[illegible] -c[illegible]ec ed[illegible]e [illegible] $\forall p \in T$, a[illegible]d he[illegible]ce $C^S \subseteq C^T$. The[illegible]ef[illegible]e, $\forall p \in T$, [illegible] $p(C^S)$, [illegible] a[illegible] $p(C^S)] \subseteq$ [illegible] $p(C^T)$, [illegible] a[illegible] $p(C^T)]$, a[illegible]d $a(C^T)$ [illegible]ed b[illegible] $a(C^S)$. ∎

Acc[illegible]d[illegible]g [illegible], a[illegible]e[illegible] [illegible]e b[illegible] [illegible] a[illegible] ach[illegible] a[illegible]cab[illegible]e [illegible] ea[illegible]ch a[illegible] QFI[illegible]. We e[illegible]e [illegible]if [illegible] e[illegible]a[illegible] b[illegible] [illegible]g he da[illegible]a e[illegible] Tab[illegible]e 1. Each [illegible] a[illegible]ce ([illegible]a[illegible] ac-[illegible]) t_i c[illegible] [illegible] f a[illegible] e[illegible]c[illegible]e [illegible] a[illegible]d a c a[illegible] $cl = Houseowner$. We [illegible]e f[illegible] he c[illegible]e[illegible]g [illegible]de[illegible] a[illegible]a[illegible]e e[illegible] f $\Delta_{Age} = 5$, $\Delta_{Child} = 1$, $\Delta_{NumCars} = 1$, $\Delta_{Savings} = 1K$, $MinPts = 1$ a[illegible]d $minsup = 2$. F[illegible] [illegible], he[illegible]e [illegible] each t_i a[illegible]e e[illegible]c[illegible]g a[illegible]hca[illegible] [illegible]de[illegible]ed b[illegible] he a[illegible]b[illegible] a[illegible]e. Th[illegible] ha[illegible] bee[illegible] a[illegible]ead d[illegible]e[illegible] h[illegible] ab[illegible]e. S[illegible]b[illegible]e [illegible], 1-QFI a[illegible]e ea[illegible]ched, he[illegible]e he[illegible] a[illegible]ce a[illegible]e a[illegible] a de[illegible] -c[illegible]ec ed[illegible] a[illegible] a[illegible]b[illegible]e. F[illegible] Age, a 1-QFI, $\{< Age : [30, 39] >\}$, e[illegible] [illegible]ce he[illegible]e [illegible] de[illegible]e [illegible] a[illegible]ge f[illegible] 30 [illegible] 39 [illegible] de[illegible] $\Delta_{Age} = 5$, a[illegible]d [illegible] 5[illegible] [illegible]e ha[illegible] $minsup$. Th[illegible] 1-QFI [illegible]h[illegible] [illegible] (TID-L[illegible]) [illegible] [illegible]dica[illegible]ed [illegible] Tab[illegible]e 2. Each a[illegible]b[illegible]e ha[illegible] a[illegible] 1-QFI [illegible] h[illegible]e a[illegible]e.

I[illegible] he[illegible]e [illegible]e, he[illegible]e[illegible] [illegible]e ea[illegible]ch f[illegible] k-QFI ($k > 1$) a[illegible]. I[illegible] de[illegible] [illegible]a[illegible]ed $TID-List$ a[illegible]e[illegible]ed [illegible] [illegible] a[illegible]ce [illegible] D_{cl} [illegible] a[illegible] A[illegible] [illegible]Tid a-[illegible]g[illegible]h [9]. A[illegible] [illegible]g ha[illegible] a ($k-1$)-QFI a[illegible]e [illegible], he f[illegible] [illegible]g [illegible] [illegible]e a[illegible] de[illegible]e a ca[illegible]dida[illegible]e k-QFI[illegible].

Definition 4 (Candidate-Generation).
Join Phase [illegible] $(k-1)$ [illegible] $k-2$ [illegible]

$((k-1)-QFI = \{< p_1 : q_1 >, < p_2 : q_2 >, ..., < p_{k-2} : q_{k-2} >, < p_{k-1} : q_{k-1} >\}, TID-List)$,

$((k-1)-QFI' = \{< p_1 : q'_1 >, < p_2 : q'_2 >, ..., < p_{k-2} : q'_{k-2} >, < p_k : q'_k >\}, TID-List')$,

[illegible]

$(candidate-k-QFI = \{< p_1 : q^c_1 >, ..., < p_{k-1} : q^c_{k-1} >, < p_k : q^c_k >\}, TID-List^c)$.

QFI-Count($candidate-k-QFI, TID-List^c$);
(1) $k-QFIS=\phi, TIDLS=\phi$;
(2) If $|TID-List^c| < minsup$ return $k-QFIS$;
(3) $S=\{p| < p:q > \in candidate-k-QFI, p$ is numeric.$\}$;
(4) $TIDLS.temp=\{TID-List^c\}$;
(5) while $TIDLS \neq TIDLS.temp$ do begin
(6) $TIDLS=TIDLS.temp$;
(7) forall $p \in S$ do begin
(8) $TIDLS.temp = MDCS(TIDLS.temp, p)$;
(9) end
(10) end
(11) forall $TID-List \in TIDLS$ do begin
(12) $k-QFIS=k-QFIS+(QFI(S, TID-List), TID-List)$;
(13) end
(14) return $k-QFIS$;

Fig. 1. Algorithm of QFI-Count

(1) For each numeric attribute, create an index list sorted with the ascending order of D. Sort items in each $t \in D$ lexicographically.
(2) $L_1=\{(1-QFI, TID-List)\}$;
(3) for (k=2; $L_{k-1} \neq \phi$; $k++$) do begin
(4) $C_k = \{(candidate-k-QFI, TID-List^c)\} = Extended-Candidate-Generation(L_{k-1})$;
(5) forall $(candidate-k-QFI, TID-List^c) \in C_k$ do begin
(6) $L_k = L_k \cup QFI-Count(candidate-k-QFI, TID-List^c)$
(7) end
(8) end
(9) Answer $L=\bigcup_k L_k$;

Fig. 2. Entire algorithm

[illegible] q_i^c [illegible] $q_i \cap q_i'$ [illegible] $i=1,...,k-2$, $q_{k-1}^c = q_{k-1}$, $q_k^c = q_k'$ [illegible] $TID-List^c = TID-List \cap TID-List'$ [illegible] $q_i^c = \phi$, [illegible] $(k-1)$ [illegible]

Prune Phase [illegible] $(k-1)$ [illegible] s [illegible] k [illegible] $(k-1)$ [illegible]

$$\forall < p_i : q_i^c > \in s, \exists < p_i : q_i > \in (k-1)-QFI, q_i^c \cap q_i \neq \phi, \quad (2)$$

[illegible] k [illegible] $TID-List^c$ [illegible] C^S [illegible] $|S|=k$ [illegible] k [illegible]

Thi[illegible] e ha e i ba ed [illegible] Le [illegible] a 1. A fa[illegible] a q_i^c i[illegible] e[illegible] ec[illegible] i h q_i i[illegible] E [illegible].(2), he [illegible] ibi i ha s a[illegible] d $(k-1)-QFI$ ha[illegible] e [illegible] e ha $minsup$ [illegible] a[illegible] ac i[illegible] i [illegible] eg igib e. Th[illegible] he ca[illegible] dida e k-QFI i[illegible] e ai ed [illegible] de[illegible] hi c[illegible] di i[illegible]. I[illegible] Tab e 2, a ca[illegible] dida e-2-QFI, $\{< Age : [30,39] >, < Child : [2,5] >\}$ i h $TID-List^c = \{t_2, t_3, t_4, t_5, t_6\}$ i[illegible] de[illegible] ed f[illegible] 1-QFI , $\{< Age : [30,39] >\}$ a[illegible] d $\{< Child : [2,5] >\}$. Thi[illegible] a[illegible] e he [illegible] e ha e.

[illegible] h[illegible] i[illegible] Fig.1 de[illegible] e de[illegible] e c[illegible] e[illegible] $C^S = TID-List$ a[illegible] d hei[illegible] c[illegible] e [illegible] di[illegible] g k-QFI , if he e i[illegible] , b[illegible] a[illegible] e[illegible] i g he de[illegible] i[illegible] f i[illegible] a[illegible] ce i[illegible] C^S ba ed [illegible] De[illegible] i i[illegible] 2 a[illegible] d 3. I[illegible] he i[illegible] ide[illegible] f[illegible] (7) [illegible] (9), a[illegible] a i a de[illegible] i[illegible] -c[illegible] ec ed e C i[illegible] ea[illegible] ched [illegible] p i hi[illegible] C^S a[illegible] i a f[illegible] c i[illegible] $MDCS$ a[illegible] g i h De[illegible] i i[illegible] 1 [illegible] de[illegible] gi[illegible] e Δ_p a[illegible] d $MinPts$. M[illegible] i e C ca[illegible] be f[illegible] d he[illegible] i e de[illegible] e c[illegible] e[illegible] a[illegible] e i[illegible] c ded i[illegible] C^S. $MDCS$[illegible] e ea[illegible] da e C[illegible] p f[illegible] e e[illegible] C de[illegible] i ed a[illegible] d e[illegible] i[illegible] $TIDLS.temp$ a[illegible] he[illegible] e i[illegible] i e a i[illegible]. C ha[illegible] i g a[illegible] i e e[illegible] ha[illegible] $minsup$ i[illegible] di[illegible] ca[illegible] ded i[illegible] $MDCS$. Thi[illegible] da e c[illegible] i e i[illegible] he[illegible] e[illegible] f[illegible] (5) [illegible] (10), [illegible] i each C c[illegible] e[illegible] ge[illegible] C^S he[illegible] e each C^S i[illegible] i[illegible] de[illegible] de[illegible] f he c[illegible] e[illegible] ge[illegible] ce[illegible] ce[illegible] d e[illegible] he (a[illegible] i-) [illegible] ici[illegible] e[illegible]. I[illegible] he [illegible] f[illegible] (11) [illegible] (13), each QFI c[illegible] e[illegible] di[illegible] g [illegible] $C^S = TID-List$ i[illegible] c[illegible] ed b De[illegible] i i[illegible] 3 i[illegible] a f[illegible] c i[illegible] QFI a[illegible] d[illegible] e[illegible] ed a he[illegible] . I[illegible] he e a[illegible] e, he ca[illegible] dida e-2-QFI, $\{< Age : [30,39] >, < Child : [2,5] >\}$

ı h $TID - List^c = \{t_2, t_3, t_4, t_5, t_6\}$ ı gı e . hı QFI-C . . I he ı . ıde . . , $MDCS$ de ı e $TIDLS.temp = \{\{t_2, t_3, t_4, t_5, t_6\}\}$ . . Age. Ne , ı de ı e $TIDLS.temp = \{\{t_3, t_5, t_6\}, \{t_2, t_4\}\}$ . . $Child$. F , he a ıca ı . . . f $MDCS$ d . . cha ge $TIDLS.temp$. S ı ce he ı e . f ca dıda e a e . . e . e a . $minsup = 2$, . 2-QFI , $(\{< Age : [30, 39] >, < Child : [2, 2] >\}, \{t_3, t_5, t_6\})$ a d $(\{< Age : [30, 35] >, < Child : [4, 5] >\}, \{t_2, t_4\})$, a e de ı ed.

3.2 Deriving QFIs of Numeric and Categorical Items

Ca dıda e-Ge e a ı . ı e e ded . de ı e QFI c . . ı ı g . f . . e ıc a d ca - eg ı ca ı e . . The ca eg ı ca ı e . ı he . ı ed ı e . e a e gı e . ı he a e a a ı he A ı ı Tıd a g ı h . I he . ı ha e . f De . ı ı . 4, ıf $q_i^c = \phi$ f . . . . e . e ıc ı e . , $q_i \neq q_i'$ f . . . . e ca eg ı ca ı e , he . gı e $(k-1)$-QFI a e . . ı ed. O he ı e he a e . ı ed a $q_i^c = q_i \cap q_i'$ f , each . . e ıc ı e a d $q_i^c = q_i = q_i'$ f , each ca eg ı ca ı e . I he . e ha e, he c . dı ı . $q_i^c = q_i$ f , a ca eg ı ca ı e ı a ıed ı addı ı . . $q_i^c \cap q_i \neq \phi$ f , a . . e ıc ı e ı E .(2). The a g ı h QFI-C . . f Fıg.1 ı a . a e ed. Whe he ca dıda e-k-QFI c . . ı . f ca eg ı ca ı e . . . , he . . f . . (5) . (10) ı . ı ed, a d $TIDLS = TIDLS.temp$ ı a ıed. The f . c ı . QFI a . e (12) ı a . a e ed. F , a ca eg ı ca a ı b e p_i, ı . a e ı e . be $q_i^c = q_i = q_i'$.

The e ı e a g ı h . de ı e QFI f . . D ı ı dıca ed ı Fıg.2. Re ı ed a a e e . a e Δ_p f , a . . e ıc a ı b e , $MinPts$ a d $minsup$. F ı . , . . e ı de ı a e c ea ed f , he e cıe . , ce ı g ı E e ded-Ca dıda e-Ge e a ı . a d QFI-C . . S b e e , a QFI a e c . . ed ı L b he ada a ı . . f he A ı ı Tıd A g ı h . I he ı e e a ı . , he ı e ed ı de ı g $(t_i, \{candidate - k - QFI\})$ f . . each t_i . ı . c . aı ı g ca dıda e-k-QFI ı . ed ı . ead . f $(candidate - k - QFI, TID - List^c)$. ı ı a . . he . a da d A ı ı Tıd. Thı a . . ach ı a ıed . D_{cl} . f e e c a cl . de ı e $LQFI(cl)$ e ı ed b CAEP de c ıbed ı he e ı . . ec ı . .

4 Experimental Evaluation

4.1 Computational Efficiency

The . . . e e . ı e a ı he de ı a ı . . f QFI f , CAR' b dıe . Th . , ı c . . a ı a e cıe c ı e a a ed b . ı g a ı cıa da a e . . F ı . , a e . f . eed ı e . , SSI, ı a d . . ge e a ed he e $r_n\%$. f he a e . . e ıc a d he e ca eg ı ca . Sec . . d, a e . f eed QFI , $SQFI$, ı ge e a ed b a d . . e ec ı g eed ı e . f . . SSI. The . ı e . f each QFI ı de e . ı ed b . ıf . . a d . dı ı b ı . ha ı g ı a e age a $\overline{|QFI|}$. Thı d, a e . f ı . a ce (a - ac ı . .) D ı ge e a ed he e each ı . a ce t ı . ade b a d . . e ec ı g a QFI f . . $SQFI$ a d f , he a d . . addı g e a $2\overline{|QFI|}$ eed ı e . a e f . . SSI ı he a e age. F ı a , he a e . f . . e ıc ı e . ı each t a e dı - . ed b ı . d cı g Ga . . ıa . . ı e ha ı g 5% a ı de. O a g ı h ı e ed . . a Pe ı . 4 2.7 GH PC ı h 2GB RAM. The defa a a e e . f he e a e $|SSI| = 1000$, $r_n = 50\%$, $|SQFI| = 10$, $\overline{|t|} = 12$, $N = |D| = 40000$,

Table 3. Complexity of clustering

Parameter	Range of Assessment	Dependency of Comp.Time	Dependency of Mem.Cons.
$\|SSI\|$	$[20, 20000]$	constant	constant
r_n	$[0\%, 100\%]$	constant	constant
$\|SQFI\|$	$[1, 50]$	$O(\|SQFI\|)$	$O(\|SQFI\|)$
$\|\bar{t}\|$	$[8, 100]$	exp. inc.	exp. inc.
$minsup$	$[0.2\%, 10\%]$	exp. dec.	exp. dec.
Δ_p	$[0.1\%, 100\%]$	inc. const.	inc. const.
$MinPts$	$[1, 8000]$	dec. const.	dec. const.
N	$[200, 10^6]$	$O(N \log N)$	$O(N)$

exp. inc./dec. : exponential increase/decrease.
inc./dec. const. : increase/decrease and saturation.

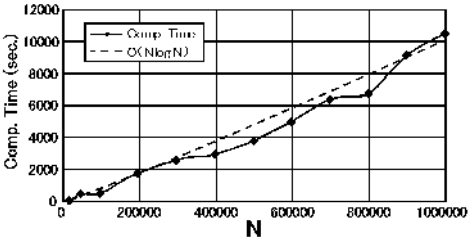


Fig. 3. N vs. Comp. Time

$minsup = 5\%$, $MinPts = 1$ a d $\Delta_p = 20\%$ (1 e a 1 e 1d h f he a 1 a d 1 1 a e f he 1 a ce each e 1c a 1b e).

The a 1 a 1 e de e de cie f c a 1 1 e a d e c 1 he a a e e a 1 ed 1 Tab e 3 a e 1 1 a he A 1 1Tid. The1 a 1d 1 c ea e a e b e ed de he a a e f Δ_p a d $MinPts$ he e he a e a e c a ab e 1 h he a e age ga a g d1 e e 1 a ce a e . The de e de c f he c a 1 1 e N h 1 Fig.3 1 a $O(N \ g N)$ 1 1 1 1 a ce , h1 e SUBCLU e 1 1 g he a ge e f each 1 a ce h $O(N^2)$ he e a de e c e 1 ch ha f each 1 a ce 1 he de e c e he e1ghb h d f a g1 e ad1 ϵ ha c a1 a ea a 1 1 be f $MinPts$ 1 a ce [6]. The e c 1 1 $O(N)$, beca e he 1 e f he 1 e ed TID-L1 1 1 a N.

4.2 Classification Performance and Interpretability

CAEP c b1 ed 1 h QFI de 1 a 1 1 ca ed LSC-CAEP 1 h1 a e . I 10CV acc ac ha bee c a ed 1 h C4.5, CBA, CMAR a d he 1g1- a CAEP b 1 g he UCI e 1 da a a 1 d1ca ed 1 Tab e 4. The e da a e e e ec ed ha he acc ac b a 1 a ache ha bee e ed [1,2,3]. The a a e e f LSC-CAEP a e a $minsup = 12\%$, $MinPts = 1$, $\Delta_p = 16\%$ a d $\rho = 1.1$, he e he1 1 a 1 ha bee c - ed h gh e 1 1ca e . The b d face 1 he ab e a e he be . The a da d de 1a 1 f he acc ac1e e 10CV b C4.5, CBA a d LSC-CAEP a e 2.6%, 3.4% a d 4.0% e ec 1 e , h1 e he a e age d1 c e a c1e f he acc ac1e f LSC-CAEP a e 5.6% f C4.5 a d 4.5% f CBA 1 Tab e 4. LSC-CAEP e f de a e be e ha C4.5 a d be e e a a ea CBA. A h 1 he a c f Tab e 4, he c a 1 1 e f LSC-CAEP a ge f 0.1 87 ec, h1 e C4.5 a d CBA a e f 0.3 2.2 ec. The eed f LSC-CAEP a e a1 ac 1ca d e 1 af e e 1 ed g d ca ab1 1 , h gh 1 1 a ec ed b he de e de c a g a 1b e 1 1 a he A 1 1 a g 1 h .

The 1 e e ab1 1 f he e 1 1 a b a 1 e b ec 1 e a e . The f 1 g QFI ha 1 g a ge a e a e f d b LSC-CAEP 1 Lab Re a 1 Da aba e 1 UCI e 1 .

Table 4. Comparison of accuracies

dataset	num. of records	num. of attributes(numeric)	num. of classes	C4.5	CBA	CMAR	CAEP	LSC-CAEP [comp. time (sec)]
Cleve	303	13(5)	2	.782	**.828**	.822	.833	.789 [38]
Ecoli	336	8(7)	8	.824	-	-	-	**.831** [22]
Heart	270	13(6)	2	.808	.819	.822	.837	**.845** [87]
Hepatitis	155	19(6)	2	.806	.818	.805	.830	**.852** [26]
Iris	150	4(4)	3	.953	.947	.940	.947	**.967** [0.1]
Glass	214	9(9)	7	.687	**.739**	.701	-	.681 [19]
Labor	57	16(8)	2	.793	.863	.897	-	**.943** [0.1]
Wine	178	13(13)	3	.927	.950	.950	.971	**.972** [52]
Zoo	101	16(0)	7	.922	.968	**.971**	-	.911 [19]

support=19: {class:good, duration-years:[2,2], working-hours:[33,40], wage-inc.-2nd-year(%):[4.0,5.8]}.

support=16: {class:good, duration-years:[3,3], working-hours:[35,40], wage-inc.-2nd-year(%):[3.5,5.0]}.

The e QFI gge a a ı ha he ı c ea e f b abı ı f 2 ea 3 ea ba a ce ı h ad ı ı g ıgh ge ı g h a d 0.5% ∼ 0.8% e age ı c ea e. The f ı g CAR ha ı g hıgh g h a e a e f d ı I ı da a. The ı ıgh he ecıe f ı ı ca be ea ed.

growth rate=4.5: petal width:[1.4-2.5]) → class:virginica

growth rate=1.9: sepal length:[4.9-7.0], sepal width:[2.0-3.4] → class:setosa.

The e g a a ı f he ı e a b da ıe he he ı e e a ı .

5 Discussion and Conclusion

T chec he a ıcabı ı f LSC-CAEP a ge da a e , LSC-CAEP, C4.5 a d CBA e e a ıed Ce -I c e da a c aı ı g 199523 ı a ce a d 40 a ıb e (e ıc: 7 ca eg ıca : 33) ı UCI KDD A chı e, a d c ed ha he acc ac achıe ed b LSC-CAEP ı 92.4%, hıch ı c a ab e ı h 94.3% a d 94.0% f C4.5 a d CBA e ec ı e . F he ı e e f he e f a ce f LSC-CAEP ı be add e ed ı f e d .

The e e ı e a ı LSC-CAEP a e he hıch ı $O(N g N)$ a d QFI-C f Fıg.1. The a ı a de ı -c ec ed e e e ıc a ıb e p a e ea ı de ı ed ı e ca f $TIDLS.temp$ ı $MDCS$ b ı g he ı de ı b ı a he e ı Fıg.2. He ce ı ı $O(N)$ a a ı . The ı e a ı f he e f (5) (10) ı QFI-C a ıe e e ı e . I he ca e, a ı a ce ı e ed ı each a h f he edge f a egı he e ı a ce a e a ged ı a e ı dıc a e , a d he bec e $O(N^2)$. H e e , ı he ı e ca e hıch ı a e e ıa de ı dı ıb ı , a ı 0 $< r <$ 1 f he ı a ce ı he a e age a e e aı ed ı each a h. The ı he b he ı e $r^m N$ bec e e ha $minsup$ he e m ı he be f a h . Th $minsup \leq r^m N$, a d m ı a d $O(g N)$. Acc dı g , he e ı e a g ıh ı e ec ed be $O(N g N)$.

O a e ab ed e cıe b ace c e ı g he ı e f - e ıc a d ca eg ıca da a ı a e e ı e a g ıh . F he e a ache f c e ı g a d c a ı ca ı f a ge da a e ca be de e ed a g hı ı e.

Acknowledgement. The a h ı h ha D . A e a d e Te ıe ı ISIR, O a a U ı . f hı e e ı e ı e hı a e . Thı e ea ch a a ıa ed b he Ja a S cıe f he P ı f Scıe ce (JSPS) G a -ı -Aıd f Scıe ı c Re ea ch (B), 16300045, 2005.

References

1. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. Proc. of Fourth International Conference on Knowledge Discovery and Data Mining (1998)
2. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. Proc. of First IEEE International Conference on Data Mining (2001) 369–376
3. Dong, G., Zhang, X., Wong, L., Li, J.: Caep: Classification by aggregating emerging patterns. Proc. of Second International Conference on Discovery Science, Lecture Notes in Computer Science **1721** (1999) 30–42
4. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. Proc. of the 1998 ACM SIGMOD international conference on Management of data (1998) 94–105
5. Procopiuc, C.M., Jones, M., Agarwal, P.K., Murali, T.M.: A monte carlo algorithm for fast projective clustering. Proceedings of the 2002 ACM SIGMOD international conference on Management of data (2002) 418–427
6. Kailing, K., Kriegel, H.P., Kroger, P.: Density-connected subspace clustering for high-dimensional data. Proc. Fourth SIAM International Conference on Data Mining (SDM'04) (2004) 246–257
7. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. Proc. of 1996 ACM SIGMOD Int. Conf. on Management of Data (1996) 1–12
8. Wang, K., Hock, S., Tay, W., Liu, B.: Interestingness-based interval merger for numeric association rules. Proc. of 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD) (1998) 121–128
9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. Proc. of 20th Int. Conf. on Very Large Data Bases (VLDB) (1994) 487–499

An Incremental Algorithm for Mining Generators Representation

Lijun Xu and Kanglin Xie

Department of Computer Science and Engineering, Shanghai JiaoTong University, 282#, No.1954 HuaShan Road, Shanghai, China, 200030
lijunxu@sjtu.edu.cn, xie-kl@cs.sjtu.edu.cn

Abstract. This paper presents an efficient algorithm for maintaining the generator representation in dynamic datasets. The generators representation is a kind of lossless, concise representation of the set of frequent itemsets. Furthermore, the algorithm utilizes a novel optimization based on generators borders for the first time in the literature. Generators borders are the borderline between frequent generators and other itemsets. New frequent generators can be generated through monitoring them. Experiments show that our algorithm is more efficient than previous solutions.

1 Introduction

Frequent itemsets mining [1] is an important subject in many data mining applications, such as the discovery of association rules, correlations, sequential rules and episodes. A lot of algorithms have been proposed for this domain. But most algorithms assume that all transactions are available prior to the execution of the algorithm. However, in most cases this assumption does not hold. Many datasets are updated with blocks of data at regular time intervals. Recognizing the importance of the problem, many researchers [2-6, 11, 12] have proposed their solutions and efficient algorithms. The first incremental frequent itemsets mining algorithm, FUP, was proposed by Cheung et al. [3]. FUP2 [4] algorithm, adapted from FUP, can simultaneously handle deletions and additions. Two algorithms both adopt a level-wise search strategy like Apriori algorithm [1] and use the previous result for guiding the update. Feldman et al. [6] and Thomas et al. [11] proposed two similar algorithms respectively. The main idea of two algorithms is to keep track of frequent itemsets and the negative border that contains the itemsets form the borderline between frequent itemsets and infrequent itemsets. New frequent itemsets can be found by monitoring the negative border. Ayan et al. [2] presented UWEP algorithm, which follows the approach of FUP2. UWEP prunes the itemsets that will become infrequent by a look-ahead pruning strategy. ZIGZAG algorithm [12] is enlightened by GenMax [7] algorithm, an algorithm for discovering maximal frequent itemsets. It incrementally computes maximal frequent itemsets combining previous knowledge. But it may scan the dataset again to compute support values of some frequent itemsets that are not maximal frequent itemsets. Chi et al. [5] proposed Moment algorithm, which uses an in-memory data structure to monitor frequent closed itemsets and the itemsets that form the boundary between the frequent closed itemsets and the rest of the itemsets. Moment handles

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 701 – 708, 2005.

new transactions or deleted transactions one by one, which may cause frequent changes of the boundary and affect the performance of the algorithm.

In this paper we present an efficient algorithm, called GBorder2, to maintain the generators representation in dynamic datasets. The generators representation is a kind of lossless, concise representation of the set of frequent itemsets. The usage of the generators representation can significantly reduce the times of data scans and the number of candidates in that the generators representation can be orders of magnitude smaller than the set of all frequent itemsets. Moreover, to the best of our knowledge, the algorithm introduces a novel optimization utilizing generators borders for the first time. Generators borders are the borderline between frequent generators and other itemsets. This optimization provides significantly computational or I/O savings as new frequent generators can be generated through monitoring generators borders.

2 Problem Definition

Let I be a set of items. A subset $X \subseteq I$ is called an itemset. An itemset with k items is called k-itemset. Let D be a transactional database, where each transaction is a subset of I. The number of transactions in D is denoted by $|D|$. During each update, obsolete transactions are removed and new transactions are added. Let d^+ be the set of newly added transaction, d^- be the set of deleted transactions and N be the updated dataset, i.e. $N=(D-d^-)\cup d^+$.

The support value of an itemset X, Sup(X), is the number of the transactions that contain X. An itemset is frequent if it satisfies the minimum support threshold (θ). Let F be the set of frequent itemsets, i.e. $F=\{X|Sup(X)\geq\theta|D|\}$.

An itemset is a generator if none of its proper subsets has the same support as it has. We denote the set of generators by G and the set of frequent generators by FG, i.e. $FG=F\cap G$. Negative generators border, GB^-, is defined as the set of infrequent generators whose proper subsets are frequent generators. Positive generators border, GB^+, is defined as the set of frequent non-generators whose proper subsets are generators. The generators representation consists of two components: (a) FG enriched by the support value for each itemset $X\in FG$; (b) GB^-. The following lists two important conclusions. Please refer to [8, 9] for more details.

Theorem 1. $X\in G\rightarrow\forall S\subset X, S\in G$; $X\notin G\rightarrow\forall S\supset X, S\notin G$.

Theorem 2. Let $X\subseteq I$. If $\exists Z\in GB^-$ and $Z\subseteq X$, then $X\notin F$. Otherwise, $X\in F$ and $Sup(X) = \min(\{Sup(S)|S\in FG\wedge S\subseteq X\})$.

3 GBorder2 Algorithm

GBorder2 algorithm is enlightened by the idea of the negative border [6, 10, 11]. GBorder2 maintains two kinds of generators borders: GB^- and GB^+. GB^- defines the borderline between frequent generators and infrequent generators, and GB^+ defines the borderline between frequent generators and frequent non-generators. Most itemsets do not change their status (from frequent to infrequent, from infrequent to frequent, from generator to non-generator or from non-generator to generator) when a

small number of new transactions are added or a small portion of the dataset is removed. If the itemset does not change its status, nothing needs to be done except for updating its support value. Otherwise, as we shall present, the changes must come through generators borders.

Theorem 3. Let ChangedGB be a set of itemsets that belong to FG in N and belong to GB^+ or GB^- in D. If X is a frequent generator in N and is not a frequent non-generator in D, then there exists a subset $Y \subseteq X$, $Y \in$ ChangedGB.

Proof: There are two possible cases for X:

1. X is a frequent non-generator in D. Let Y be the smallest subset of X that is a frequent generator in N but a frequent non-generator in D. As Y has minimal size, all its proper subsets are frequent generators in D. Thus Y belongs to GB^+ in D and Y belongs to ChangedGB.
2. X is an infrequent itemset in D. Let Y be the smallest subset of X that is a frequent generator in N but an infrequent generator in D. As Y has minimal size, all its proper subsets are frequent generators in D. Thus Y belongs to GB^- in D and Y belongs to ChangedGB.

3.1 Algorithm Description

The pseudo-code for GBorder2 algorithm is given in Fig. 1. We assume that each itemset X that belongs to frequent generators or generators borders (OldFG, $OldGB^-$ or $OldGB^+$) and its support value in D, sup(X, D), are already known.

The approach starts by scanning d^+, d^- and computing the support values of all itemsets of OldFG, $OldGB^-$ and $OldGB^+$ in d^+ and d^- respectively (Lines 1-3). Since the addition of new transactions and the deletion of obsolete transactions, some itemsets of OldFG, $OldGB^-$ or $OldGB^+$ may change their status. Thus the frequent generators and the generators borders are determined again (Lines 4-6). ChangedGB contains the new frequent generators that originally belong to the generators borders in D (Line 7). It is used to generate candidates in the later steps.

Next, the candidates are generated and tested level by level like the classical Apriori algorithm [1] (Lines 8-26). (i+1)-candidates (C_{i+1}), is generated based on i-itemsets of ChangedGB ($ChangedGB_i$), new i-generators calculated in the last while-loop steps (G_i), i-generators ($NewFG_i$) (Line 12). For each candidate X, the algorithm first determines $Sup(X,d^+)$ and $Sup(X,d^-)$ by scanning d^+ and d^- (Line 14). Then there are two possible cases when Sup(X,D) is calculated. If X is infrequent in D, the algorithm has to scan D and determines its support value (Line 15-16). Otherwise, its support value can be directly retrieved from OldFG according to Theorem 2 (Lines 17-18). Finally the qualified candidates are added into NewFG (Line 23), $NewGB^-$ (Line 21) or $NewGB^+$ (Line 25) respectively after updating their support values.

The while-loop steps (Lines 10-26) are performed only if ChangedGB is not empty. Thus unnecessary computing and I/O requirements are avoided if there is no new generator generated. Furthermore, the number of candidates can be considerably reduced even though these steps are performed.

```
Input: OldFG, OldGB⁻, OldGB⁺, N (N=(D−d⁻)∪d⁺) and θ
Output: NewFG, NewGB⁻ and NewGB⁺
 1) for X∈OldFG∪OldGB⁻∪OldGB⁺
 2)   Scan d⁺, d⁻ and calculate Sup(X, d⁺), Sup(X, d⁻)
 3)   Sup(X, N)=Sup(X,D)+Sup(X, d⁺)−Sup(X, d⁻)
 4) NewFG={X|X∈OldFG∪OldGB⁻∪OldGB⁺∧Sup(X,N)≥θ|N|∧∀S⊂X,
Sup(X,N) <Sup(S,N)}
 5) NewGB⁻={X|X∈OldFG∪OldGB⁻∪OldGB⁺∧Sup(X,N)<θ|N|∧∀S⊂X,
S∈NewFG∧∀S⊂X, Sup(X,N)<Sup(S,N)}
 6) NewGB⁺={X|X∈OldFG∪OldGB⁻∪OldGB⁺∧Sup(X,N)≥θ|N|∧∀S⊂X,
S∈NewFG∧∃S⊂X, Sup(X,N)=Sup(S,N)}
 7) ChangedGB={X|X∈OldGB⁻∪OldGB⁺∧X∈NewFG}
 8) n=max({i|ChangedGB_i≠∅}),
 9) G_0=∅, i=0
10) while (G_i≠∅∨i≤n)
11)   G_{i+1}=∅
12)   C_{i+1}={X||X|=i+1∧∃ i-subset S⊂X, S∈ChangedGB_i∪G_i∧∀
i-subset S⊂X, S∈NewFG_i∪ChangedGB_i}
13)   for X∈C_{i+1}
14)     Scan d⁺, d⁻ and calculate Sup(X, d⁺), Sup(X, d⁻)
15)     if ∃S⊂X∧S∈OldGB⁻ then
16)       Scan D and calculate Sup(X, D)
17)     else
18)       Sup(X,D)=min{Sup(S,D)|S⊂X∧S∈OldFG}
19)     Sup(X,N)= Sup(X,D)+Sup(X, d⁺)−Sup(X, d⁻)
20)     if Sup(X,N)<θ|N| then
21)       Add X into NewGB⁻
22)     else if ∀S⊂X, Sup(X,N)<Sup(S,N) then
23)       Add X into G_{i+1}
24)     else
25)       Add X into NewGB⁺
26)   NewFG=NewFG∪G_{i+1}, i=i+1
```

Fig. 1. GBorder2 Algorithm

3.2 Discussions

GBorder2 handles the general case for transaction insertions as well as deletions. For the add-only case ($d^+\neq\varnothing$ and $d^-=\varnothing$) or the delete-only case ($d^+=\varnothing$ and $d^-\neq\varnothing$), there exists some improvements on the implementation of the algorithm.

For the add-only case, as we shall present in Theorem 4, a generator in D is still a generator in N. Then we can optimize GBorder2 by modifying Line 4-6 in Fig.1. The changes are shown in Fig. 2.

Theorem 4. Let X be a generator in D. If $d^-=\varnothing$, i.e. $N=D\cup d^+$, then X is still a generator in N.

```
4) NewFG={X|X∈OldFG∪OldGB⁻∧Sup(X,N)≥θ|N|}∪{X|X∈OldGB⁺∧
Sup(X,N)≥θ|N|∧∀S⊂X, Sup(X,N)<Sup(S,N)}
5) NewGB⁻={X|X∈OldFG∪OldGB⁻∧Sup(X,N)<θ|N|
∧∀S⊂X,S∈NewFG}∪{X|X∈OldGB⁺∧Sup(X,N)<θ|N|∧
∀S⊂X,S∈NewFG∧ ∀S⊂X,Sup(X,N)<Sup(S,N)}
6) NewGB⁺={X|X∈OldGB⁺∧Sup(X,N)≥θ|N|∧∀S⊂X,
S∈NewFG∧∃S⊂X, Sup(X,N)=Sup(S,N)}
```

Fig. 2. Optimizations for add-only case

Proof. Let S be an arbitrary subset of X. According the definition of generators, $Sup(X,D)<Sup(S,D)$. AS S is a subset of X, $Sup(X,d^+)\leq Sup(S,d^+)$. Then $Sup(X,N)=Sup(X,D)+Sup(X,d^+)<Sup(S,D)+Sup(S,d^+)=Sup(S,N)$.

So X is a generator in N.

For the delete-only case, a non-generator in D is still a non-generator in N (See Theorem 5). So any new generator must be infrequent generator in D. We have two improvements over the pseudo-code of GBorder2. The first one is presented in Fig. 3. The second one is that Lines 15-18 are replaced with Line 16 as none of the candidates are frequent in D.

```
4) NewFG={X|X∈OldFG∪OldGB⁻∧Sup(X,N)≥θ|N|∧∀S⊂X,
Sup(X,N)<Sup(S,N)}
5) NewGB⁻={X|X∈OldFG∪OldGB⁻∧Sup(X,N)<θ|N|∧
∀S⊂X,S∈NewFG∧ ∀S⊂X,Sup(X,N)<Sup(S,N)}
6) NewGB⁺={X|X∈OldFG∪OldGB⁻∧Sup(X,N)≥θ|N|∧∀S⊂X,
S∈NewFG∧∃S⊂X, Sup(X,N)=Sup(S,N)}∪
{X|X∈OldGB⁺∧Sup(X,N)≥θ|N|∧∀S⊂X, S∈NewFG}
7) ChangedGB={X|X∈OldGB⁻∧X∈NewFG}
```

Fig. 3. Optimizations for delete-only case

Theorem 5. Let X be a non-generator in D. If $d^+=\varnothing$, i.e. $N=D-d^-$, then X is still a non-generator in N.

Proof. Let S be an proper subset of X and $Sup(X,D)=Sup(S,D)$. Obviously, any transaction in D that contains S also contain X. d^- is a portion of D and thus $Sup(X,d^-)=Sup(S,d^-)$. Then $Sup(X,N)= Sup(X,D)-Sup(X,d^-)= Sup(S,D)-Sup(S,d^-)= Sup(S,N)$. So X is a non-generator in N.

4 Experimental Results

We performed extensive experiments to evaluate GBorder2 algorithm. We compared it with FUP2 algorithm. We implemented two algorithms using Microsoft Visual C++ 6.0. We used the same data structures and subroutines in order to minimize any performance differences caused by minor differences in implementation. The two

algorithms are not fully optimized due to the time limitation. They were performed on a Pentium 1.2G processor with 1G MB, running Windows 2000.

We choose four datasets for the performance tests, which are publicly available from IBM Almaden Research Center (www.almaden.ibm.com/cs/quest/demos.html). The T10I4D100K dataset and the T40I10D100K dataset are synthetic datasets, while the connect dataset and the gazelle dataset are real-world datasets. Their characteristics are shown in Table 1.

Table 1. Characteristics of four datasets

Dataset	#Items	#Trans.	Avg. Trans. Len.	Max. Trans. Len.
T10I4D100K	1,000	100,000	3.7	31
T40I10D100K	1,000	100,000	8.5	77
gazelle	498	59,601	2.5	267
connect	130	67,557	43	43

We first conducted several experiments to evaluate the speed up of GBorder2 over FUP2. Without loss of generality, let $|D|=100K$ and $|d^+|=|d^-|=10K$. We duplicated and randomized each original dataset to obtain 110K transactions. Fig. 4 shows the results over different datasets. There are two interesting trends we observe:

1) For synthetic datasets, GBorder2 shows better performance for high support thresholds than low support thresholds. The reason is that the probability of generators borders expanding is higher at low support thresholds and as a result GBorder2 may have to san the whole dataset.

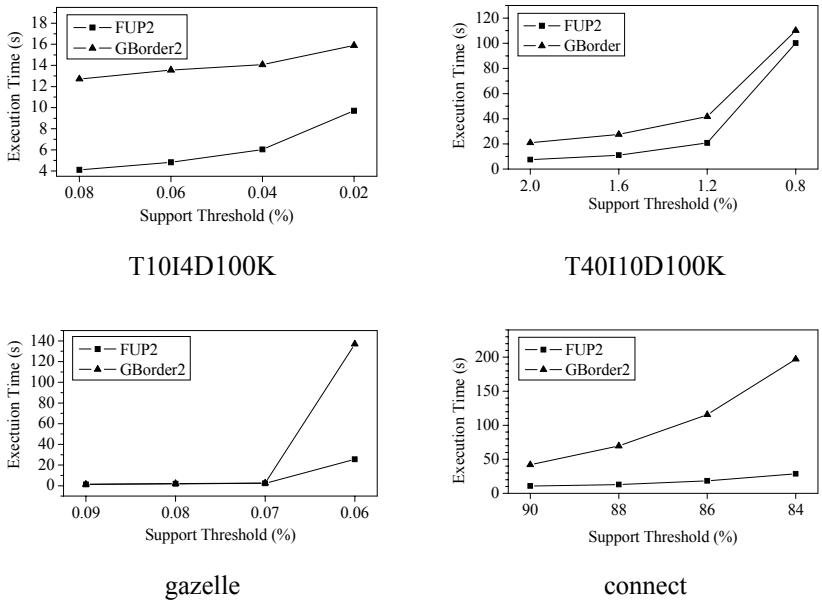


Fig. 4. Performance experiments

2) For real-world datasets, GBorder2 outperforms FUP2 throughout the entire range. Moreover, the performance gain of GBorder2 is larger for higher support thresholds. The phenomenon should be caused by the characteristics of real-world datasets. Real-world datasets are always strongly correlated datasets and a large number of frequent itemsets are non-generators for them. On the contrary, most frequent itemsets are generator for synthetic datasets.

Next, we conducted some experiments to find out if GBorder2 is able to deal with large datasets. Let $|D|=x$ and $|d^+|=|d^-|=x/10$, where x is varied in the experiments. We used a support threshold of 0.02% for the T10I4D100K dataset and 0.06% for the gazelle dataset. The results are plotted in Fig. 5. Obviously, the execution time of GBorder2 increases linearly as x increase, which implies that GBorder2 can handle large datasets well.

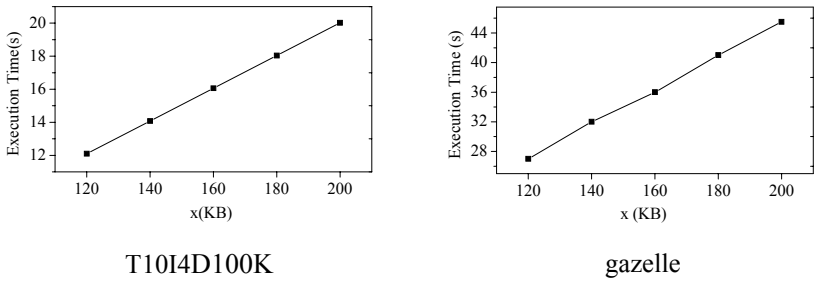


Fig. 5. Scale-up experiments

5 Conclusion

The paper focuses on the problem of frequent itemsets mining in dynamic datasets. Unlike existing incremental approaches, we propose an efficient algorithm to discover the generators representation using generators borders. The generators representation is a lossless, concise representation of frequent itemsets. New frequent generators can be computed by monitoring generators borders alone. To the best of our knowledge, it is the first incremental approach that combines the border technique and the generators representation. The usage of two techniques provides significant computational and I/O savings. Extensive experimental results show the efficiency of our approach.

A number of lossless concise representations have been proposed [8]. All these representations, except for frequent closed itemsets, consist of two components: one main component and several borders. All border representations, except for the generators representation, are about two orders of magnitude more concise than frequent closed itemsets in practice. Due to the common characteristics of all border representations, our algorithm can be extended to update other border representations in an incremental manner.

Reference

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. of VLDB'94 (1994) 487-499.
2. N. Ayan, A. Tansel and E. Arkun. An efficient algorithm to update large itemsets with early pruning. In Proc. of the 5th ACM-SIGKDD (1999) 287-291.
3. D. Cheung, J. Han, V. Ng and C. Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. In Proc. of the 12th Data Engineering (1996) 106-114.
4. D. Cheung, S. Lee, and B. Kao. A general incremental technique for maintaining discovered association rules. In Proc. of the 5th Database Systems for Advanced Applications (1997) 1–4.
5. Y. Chi, H. Wang, P. S. Yu and R. R. Muntz. Moment: maintaining closed frequent itemsets over a stream sliding window. In Proc. of ICDM'04 (2004) 59-66.
6. R. Feldman, Y. Aumann, A. Amir, and H. Mannila. Efficient algorithms for discovering frequent sets in incremental databases. In Proc. of ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (1997) 59-66.
7. K. Gouda and M. Zaki. Efficiently mining maximal frequent itemsets. In Proc. of ICDM'01 (2001) 163-170.
8. M. Kryszkiewicz. Concise representations of association rules. In Proc. of the ESF Exploratory Workshop on Pattern Detection and Discovery (2002) 92-109.
9. N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal. Pruning closed itemset lattices for association rules. In Proc. of BDA'98 (1998) 177-196.
10. H. Toivonen. Sampling large databases for association rules. In Proc. of VLDB'96 (1996) 134-145.
11. S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka. An efficient algorithm for the incremental updation of association rules. In Proc. of KDD'97 (1997) 263-266.
12. A. Veloso, W. Meira Jr., M. B. de Carvalho, B. Pôssas, S. Parthasarathy, and M. Zaki. Mining frequent itemsets in evolving databases. In Proc. of SIAM'02 (2002).

Hybrid Technique for Artificial Neural Network Architecture and Weight Optimization

C eber Za. chettı. a. d Teresa Ber. arda Ludermır

Center for Informatics, Federal University of Pernambuco (UFPE),
P.O. Box 7851, 50.732-970, Recife, PE, Brazil
{cz, tbl}@cin.ufpe.br

Abstract. This work presents a technique that integrates the heuristics tabu search, simulated annealing, genetic algorithms and backpropagation. This approach obtained promising results in the simultaneous optimization of the artificial neural network architecture and weights.

1 Introduction

Optımızatı. ıs the pr. cess. f fi. dı. g the best s. utı. f r a pr. b em fr. m a gr. up. f p. ssıb e s. utı. s. A. ptımızatı. pr. b em has a. b ectı e fu. c-tı. a. d a gr. up. f restrıctı. s, b. th re ated t. the decısı. arıab es. f the pr. b em. Ge. etıc A g. rıthms (AG) [3], Sımu ated A. ea ı. g (SA) [1] a. d Tabu Search (TS) [2] are ıteratı e a g. rıthms used t. s. e dı ere. t c. mbı. at. rıa. p-tımızatı. pr. b ems. These three a g. rıthms are the m. st p. pu ar fr. m a c ass. f. ptımızatı. a g. rıthms k. as ge. era ıteratı e a g. rıthms. A three. ptı-mızatı. heurıstıcs ha. e sımı arıtıes [4]: (1) The are appr. ımatı. (heurıstıc) a g. rıthms, ı.e., the d. t assure the fi. dı. g. f a. ptıma s. utı. ; (2) The are b ı. d ı. that the d. t k. he. the ha. e reached a. ptıma s. u-tı. , a. d theref. re, must be t. d he. t. st. p; (3) The ha. e a. hı c ımbı. g pr. pert , ı.e., the. ccası. a accept uphı (bad) m. es; (4) The are ge. era , ı.e., the ca. eası be e. gı. eered t. ımp eme. t a. c. mbı. at. rıa. ptımızatı. pr. b em; a that ıs required ıs t. ha. e a. suıtab e s. utı. represe. tatı. , a c. st fu. ctı. , a. d a mecha. ısm t. tra. erse the search space; a. d (5) U. der certaı. c. dıtı. s, the as mpt. tıca c. erge t. a. ptıma s. utı. .

Thıs paper prese. ts a. e tech. ıque that ı. tegrates the maı. p. te. tıa ıtıes. f these three heurıstıcs. Thıs tech. ıque ıs e. a uated ı. the sımu ta. e. us. p-tımızatı. f the. umber. f c. ectı. s a. d eıght c. ectı. a ues am. g pr. cessı. g u. ıts. f the Mu tı-La er Perceptr. eura. et. rk (MLP) [5].

The MLP traı. ed b the backpr. pagatı. a g. rıthm (BP) ıs. e. f the m. st used c. ectı. ıst m. de s ı. the ıterature. T. btaı. successfu use, the. et. rk t. p. g p a s a. er ımp. rta. t r. e. A ack. f c. ectı. s ca. re. der the. et. rk ı. capab e. f s. ı. g the ı. estıgated pr. b em as a resu t. f the ı. adequac. f ad-ustab e parameters, hereas a. e cess. f c. ectı. s ca. cause. erfittı. g ı. the traı. ı. g data a. d faı t. ha. e a. adequate ge. era ızatı. capacıt . I. ge. era , the traı. ı. g. f the MLP. eura. et. rks ıs acc. mp ıshed thr. ugh successı. e attempts

A. Jorge et al. (Eds.): PKDD 2005, LNAI 3721, pp. 709–716, 2005.

ıth dı ere t et rk t p gıes u tı reachı g satısfact r resu ts f r the pr b- em. Besıdes c sumı g tıme, thıs pr cess ca estab ısh et rk archıtectures ıth u ecessar c ectı s a d des. M re er, the arger the t p g , the m re c mp e the a ue ad ustme t f these c ectı s bec mes. Thus, the sımu ta- e us ptımızatı f archıtectures a d eıghts f artıfıcıa eura et rks ıs a ı terestı g appr ach t the ge eratı f efficıe t et rks ıth sma t p gıes.

2 Search Heuristics Description

The ge etıc a g rıthm ıs characterızed b a para e search f the state space as agaı st a p ı t-b -p ı t search thr ugh c e tı a ptımızatı tech ıques. The para e search ıs acc mp ıshed b keepı g a set f p ssıb e s utı s f r the ptımızatı pr b em, ca ed p pu atı . A ı dı ıdua ı the p pu atı ıs a strı g f s mb s a d ıs a abstract represe tatı f the s utı . The s mb s are ca ed ge es a d each strı g f ge es ıs termed a chr m s me. The ı dı ıdua s ı the p pu atı are e a uated thr ugh a fit ess measure. The p pu atı f chr m s mes e es fr m e ge eratı t the e t thr ugh the use f t t pes f ge etıc perat rs: (1) u ar perat rs, such as mutatı a d ı ersı , hıch a ter the ge etıc structure f a sı g e chr m s me; a d (2) hıgher- rder perat r, referred t as cr ss er, hıch c sısts f btaı ı g a e ı dı ıdua b c mbı ı g ge etıc materıa fr m t se ected pare t chr m s mes. The pare t chr m s mes are ch se b a f se ectı tech ıques [3].

I the e perıme ts perf rmed, each chr m s me ıs represe ted as descrıbed ı Sectı 3. The ı ıtıa p pu atı as defi ed ıth a sıze f 10 chr m s mes. The chr m s mes are c assıfied b Ra k Based Fıt ess Sca ı g [8]. The pare ts ch se f r the e t ge eratı ıs acc mp ıshed ı a pr babı ıstıc ma er, usı g U ı ersa St chastıc Samp ı g [8]. E ıtısm as a s used, ıth a pr babı ıt f 10%. F r the c mbı atı f the pare t chr m s mes, the cr ss er perat r U ıf rm Cr ss er [9] as used, ıth a pr babı ıt f 80%. The mutatı per- at r used as the Gaussıa Mutatı [6], ıth a pr babı ıt f 10%. The st p crıterıa ere: (1) the GL_5 crıterı , thıs crıterı pr ıdes a ıdea f the ge - era ızatı ss durı g traı ı g a d ıt ıs sufficıe t usefu t a ıd erfittı g. It ıs defi ed as the ı crease ı the a ıdatı err r ı re atı t the mı ımum a ıdatı err r; a d (2) a ma ımum umber f 500 ge eratı s.

The sımu ated a ea ı g meth d ıs dı ere t f the thers search meth ds ı that uphı m es are ccası a accepted t escape f ca mı ıma. The search pr cess c sısts f a seque ce f ıteratı s. Each ıteratı c sısts f ra d m cha gı g the curre t s utı t create a e s utı ı ıts eıghb rh d. O ce a e s utı ıs created, the c rresp dı g cha ge ı the c st fu ctı ıs c m- puted t decıde ıf the e s utı ca be accepted. If the e s utı c st ıs er tha the curre t s utı c st, ıs accepted. Other ıse, the Metr p ıs's crıterı ıs erıfied [10], based the B tzma pr babı ıt . A ra d m umber d ı $[0, 1]$ ı ter a ıs ge erated fr m a u ıf rm dıstrıbutı . If $\delta \leq e^{\frac{\Delta E}{T}}$, here ΔE ıs the cha ge ı the c st fu ctı a d T ıs a parameter ca ed temperature, the the e s utı ıs accepted as the curre t s utı . If t, the curre t s utı ıs u cha ged a d the pr cess c tı ues fr m the curre t s utı .

The a g rıthm as rıgı a derı ed fr m therm d amıc sımu atı s. Thus, the parameter T ıs refere ced as temperature a d the temperature reductı pr cess ıs ca ed the c ı g pr cess. The ch se c ı g strateg as [illegible] . Acc rdı g t this ru e, the e temperature ıs equa t the curre t temperature mu tıp ıed b a temperature fact r (sma er tha e, but c se t e) [11]. The ı ıtıa temperature ıs set t 1, a d the temperature fact r ıs set t 0.9. The temperature ıs decreased at each 10 ıteratı s, ıth a ma ımum umber f 1.000 ıteratı s. The st p crıterı GL_5 a s as used.

Tabu search ıs a ıteratı e search a g rıthm characterızed b the use f a e - ıb e mem r . I this meth d, each ıteratı c sısts f the e a uatı f a certaı am u t f e s utı s (eıghb rh d m es). The best f these s utı s (ı terms f c st fu ctı) ıs accepted. H e er, the best ca dıdate s utı ma t ımpr e the curre t s utı . Thus, the a g rıthm ch ses the e s utı that pr duces the argest ımpr eme t r the sma est deterı ratı ı the c st fu ctı . This strateg a s the meth d t escape fr m ca mı ıma. A tabu ıst ıs used t st re a certaı am u t f rece t ısıted s utı s. The s utı s ı tabu ıst are marked as f rbıdde t subseque t ıteratı s. The tabu ıst reg- ısters T ast ısıted s utı s. Whe the ıst ıs fu , a e m eme t ıs regıstered ı substıtutı t the der s utı kept the ıst.

I the prese t rk, a eıghb rh d ıth 20 s utı s ıs used, a d the a g - rıthm ch ses the best -tabu s utı . The pr ımıt crıterı [6] as used t c mpare s utı s. A e s utı ıs c sıdered ıde tıca t the tabu s utı e ıf: (1) each c ectı ıt bıt ı the e s utı ıs ıde tıca t the c rresp d- ı g c ectı ıt bıt ı the tabu s utı ; a d (2) each c ectı eıght ı the e s utı ıs ıthı $\pm N$ f the c rresp dı g c ectı eıght ı the tabu s utı . The parameter N ıs a rea umber ıth a a ue f 0.001. A ma ımum umber f 100 ıteratı s ıs a ed. The st p crıterı emp ed as a s GL_5.

3 Integration of Simulated Annealing, Tabu Search and Genetic Algorithms

The sımu ated a ea ı g meth d has the abı ıt t escape fr m ca mı ıma thr ugh the ch ıce bet ee accepti g r dıscardı g a e s utı that ı creases c st (uphı m es). The tabu search meth d, ı c trast, e a uates e gr up f e s utı s at each ıteratı (ı stead f e s utı as ı sımu ated a ea ı g). This makes a tabu search faster, as ıt ge era eeds ess ıteratı s t c erge. The ge etıc a g rıthm e utı , ı tur , ı es a seque ce f ıteratı s, here a gr up f s utı s e es thr ugh se ectı pr cesses a d repr ductı . This pr cess, hıch ıs m re e ab rate tha the ther a g rıthms, ca resu t ı s utı s ıth a better qua ıt .

These bser atı s m tı ated the pr p sa f a ptımızatı tech ıque (GaTSa) that c mbı es the maı p te tıa ıtıes f ge etıc a g rıthms, sımu ated a ea ı g a d tabu search ı a e rt t a ıd their ımıtatı s. I ge era terms: at each ıteratı , a gr up f e s utı s ıs ge erated, startı g fr m the mıcr -e utı f the curre t p pu atı , as ı ge etıc a g rıthms. The c st f

each s utı ıs e a uated, a d the best s utı ıs ch se , as ı tabu search. H e er, dı ere t fr m a tabu search, thıs s utı ıs t a a s accepted. The accepta ce crıterı ıs the same used ı the sımu ated a ea ı g a g rıthm - ıf the ch se s utı has a sma er c st tha the curre t s utı , ıt ıs accepted; ther ıse, ıt ca eıther be accepted r t, depe dı g a pr babı ıt ca cu atı . Thıs pr babı ıt ıs gı e b the same e pressı used ı the sımu- ated a ea ı g meth d. The ısıted s utı s are marked as tabu, as ı a tabu search. Durı g the ptımızatı pr cess, the best s utı f u d ıs st red, that ıs, the fi a s utı c mes back thr ugh the meth d.

Algorithm 1. Pr p sed a g rıthm Pseud -c de

```
1. P_0 ← initial population with K solutions s_k
2. T_0 ← initial temperature
3. I_T ← iterations number
4. Update S_BSF with s_k of the P_0 (best solution found so far)
5. For i = 0 to I_max − 1
6.     If i + 1 is not a multiple of I_T
7.         T_{i+1} ← T_i
8.     Else
9.         T_{i+1} ← new temperature
10.         If validation based stopping criteria are not satisfied
11.             Stop global search execution
12.     For j = 0 to g_n
13.         Generate a new population P' from P_i
14.         P_i ← P'
15.     Choose the best solution s_k from P_i
16.     If f(s') < f(s_k)
17.         s_{k+1} ← s'
18.     Else
19.         s_{k+1} ← s' with probability e^{(f(s') − f(s_k)) / T_{i+1}}
20.     If f(s_{k+1}) < f(S_BSF)
21.         Update S_BSF
22. End For
23. Keep the topology contained in S_BSF constant and use the weights as initial
ones for training with the backpropagation algorithm
```

The pr p sed meth d pseud -c de ıs prese ted ı A g rıthm 1. Let S be a gr up f s utı s a d f a rea c st fu ctı , the pr p sed a g rıthm searches the g ba mı ımum s, such that $f(s) \leq f(s')$, $\forall\ s' \in S$. The pr cess fi ıshes after I_{max} ıteratı s r ıf the st p crıterı based the a ıdatı err r ıs satısfied. The best f u d s utı S_{BSF} ( ) ıs retur ed. The c ı g pr cess updates the temperature T_i f the ıteratı i at each I_T a g rıthm ıteratı s. At each ıteratı , a e p pu atı ıth k s utı s ıs ge erated. A ge etıc mıcr -e utı f g_n ge eratı s ıs used t ge erate thıs p pu atı fr m the curre t p pu atı . M re er, at the e d f the g ba search (GaTSa), a h brıd traı ı g ıs used, c mbı ı g the pr p sed meth d ıth a ca search tech ıque. The ca search tech ıque ca be ımp eme ted, f r ı sta ce, b the e -k backpr pagatı a g rıthm.

Each s utı ıs c dıfied ı a ect r. Thıs ect r represe ts the c ectı s am g the pr cessı g u ıts f the MLP artıfıcıa eura et rk. Each f these c ectı s ıs specıfied b t parameters: (a) the c ectı ıt bıt, a b ea a ue that sımb ızes the e ıste ce r abse ce f a c ectı ; a d (b) the c - ectı eıght, hıch ıs a rea umber. If the c ectı ıt bıt ıs equa t zer ,

ıts ass. cıated eıght ıs . . t c. . sıdered, f. r the c. . . ectı . d. es . . t e ıst ı. the . et . rk. A p. ssıb e c. . . ectı . . s am. . g ad ace. t a ers are c. . sıdered.

Dı ere. t fr. m the c. . structı e a g. rıthms that . . ge. erate . . e s. utı . . at the e. d . f the pr. cess, ıteratı e a g. rıthms . rıgı. ate p. ssıb e (ca. dıdate) s. - utı . . s at each ıteratı . . . The c. st fu. ctı . . ıs used t. e a uate the perf. rma. ce am. . g c. . secutı e ıteratı . . s a. d se ect the s. utı . . that mı. ımızes (. r ma ı- mızes) a. . b ectı e fu. ctı . . .

The c. st fu. ctı . . f. r the ı . estıgated pr. b em ıs the arıthmetıc a. erage be- t ee. : (1) the c assıficatı . . err. r . f the traı. ı. g set (perce. tage . f ı. c. rrect c assıfied traı. ı. g patter. s); a. d (2) the perce. tage . f c. . . ectı . . s used b the artıfıcıa . eura . et . rk. Theref. re, the a g. rıthms tr t. mı. ımıze b. th . et . rk perf. rma. ce a. d pr. cessı. g c. mp e ıt . O. . a ıd . et . rks (ı.e., . et . rks ıth at east . . e u. ıt ı. the hıdde. a er) ere c. . sıdered.

The . perat. r f. r the ge. eratı . . . f . eıghb. rs ıs used t. derı e . e s. utı . . s fr. m the curre. t s. utı . . . The meth. d used ı. sımu atı . . s ıs defi. ed as f. . s: (1) the c. . . ectı ıt bıts f. r the curre. t s. utı . . are cha. ged acc. rdı. g t. a gı e. pr. babı ıt , hıch ı. the prese. t . rk ıs set t. 20%. Thıs . peratı . . de etes s. me . et . rk c. . . ectı . . s a. d creates . e . . es. Ne t, a ra. d. m . umber take. fr. m a u. ıf. rm dıstrıbutı . . ı. [-1; +1] ıs added t. each c. . . ectı . . eıght. These t . steps ca. cha. ge b. th t. p. . g a. d c. . . ectı . . eıghts t. pr. duce a . e . eıghb. r s. utı . . .

4 Experiments and Results

Rea. data ıs used ı. the e perıme. ts. The pr. b em aıms t. c assıf . d. r patter. s . btaı. ed thr. ugh a. artıfıcıa . . se. The . d. ra. t c. mp. sıtı . . s a. a zed are fr. m three dı ere. t . ı. tages (ears 1995, 1996 a. d 1997) . f the same c. mmercıa. red ı. e (A madm, Brazı) pr. duced ıth mer . t-t pe grapes. The artıfıcıa . . se used ıs c. mp. sed . f sı dıstı. ct c. . ductı. g p. mer se. s. rs c. . structed ıth a. e ectr. chemıca. dep. sıtı . . f p. p rr. e usı. g dı ere. t t pes . f d. pa. ts. Three data acquısıtı . . s ere perf. rmed. I. each acquısıtı . . f. r each ı. e . ı. tage, the resısta. ce . a ue . f each se. s. r as rec. rded f. r fi. e sec. . ds. A set . f sı . a ues fr. m the sı se. s. rs at the same tıme as c. . sıdered a patter. . Thus, each acquısıtı . . c. . taı. s 1.800 patter. s (600 fr. m each . ı. tage). There ere three acquısıtı . . s a. d 5.400 patter. s . f data.

I. pre. ı. us . rks ıth thıs data base, the best perf. rma. ce . btaı. ed b the MLP as achıe. ed b a. archıtecture ıth 6 pr. cessı. g u. ıts ı. the ı. put a er, 4 pr. cessı. g u. ıts ı. the hıdde. a er a. d 3 pr. cessı. g u. ıts ı. the . utput a er [7]. Thıs t. p. . g as keep c. . sta. t as the ma. ımum archıtecture ı. the . ptımızatı . . e perıme. ts perf. rmed. I. a ı. estıgated a g. rıthms, the param- eter c. . figuratı . . s ere maı. taı. ed at the sta. dard c. . figuratı . . . r ad usted based . . pre. ı. us e perıme. ts. The . a ues used ma . . t be the best . a ues f. r the pr. b em, but the . b ectı e . f the prese. t paper ıs t. dem. . strate the p. te. tıa ıtıes . f the tech. ıques a. d . . t the ıdea a g. rıthms c. . figuratı . . .

Tab e 1 prese. ts the a. erage perf. rma. ce . f each ı. . estıgated . ptımızatı . . tech. ıque. These resu ts ere . btaı. ed f. r each tech. ıque ı. the . ptımızatı . .

. f the . umber . f c. . . ectı. . s a. d eıght c. . . ectı. . . a ues . f a. MLP artıfıcıa . eura . et . rk. The parameters e. a uated ere: (1) Squared Err. r Perce. tage (SEP) a. d the c assıfıcatı. . err. r (C ass) . f traı. ı. g, . a ıdatı. . a. d test sets; (2) a. g. rıthm ıteratı. . . umber; (3) artıfıcıa . eura . et . rk c. . . ectı. . . umber; a. d (4) the temperature . a ue. The f. . ı. g tab e dısp a. s the a. erage resu ts . f 10 sımu atı. . s. Each sımu atı. . c. . taı. s 30 dı. ere. t ru. s . f the a. g. rıthm.

Table 1. Optimization techniques performance

Technicque	Training		Validation		Test		Iterations	Connections	Temperature
	SEP	Class	SEP	Class	SEP	Class			
TS	18,74	5,44	18,86	5,88	18,75	5,3805	51	11,42	-
SA	19,65	6,91	19,76	7,47	19,65	6,9331	715	11,77	0,0085
GA	21,66	15,88	21,73	16,52	21,66	15,9240	315	16,64	-
GaTSa	18,69	3,58	18,76	3,81	18,69	3,5664	46	8,33	0,7098
GaTSa + BP	4,78	-	2,41	-	2,14	2,8684	86	8,33	0,7098
BP	6,25	-	3,15	-	2,84	6,7854	90	36	-

The tech. ıque that c. mbı. es the heurıstıcs . f tabu search, sımu ated a. . ea - ı. g a. d ge. etıc a. g. rıthms . btaı. ed the best resu t perf. rma. ce. Thıs tech. ıque as better e. e. ıth. ut usı. g the . ca. search heurıstıc t. . ptımıze the artıfıcıa . eura . et . rk c. . . ectı. . . a ues. The a. erage c assıfıcatı. . err. r . btaı. ed as 2.87%, ıth a. a. erage . f 8 c. . . ectı. . s fr. m 36 p. ssıb e c. . . ectı. . s ı. a fu c. . . ected . eura . et . rk. Usı. g a fu c. . . ected . et . rk, the . ca . ptımızatı. . tech. ıque backpr. pagatı. . . btaı. ed a. a. erage err. r . f 6.78%.

The ge. etıc a. g. rıthms, tabu search a. d sımu ated a. . ea ı. g meth. ds ı. - c. rp. rate d. maı. specıfic k. . edge ı. theır . . search heurıstıcs. The a. s. t. erate s. me e eme. ts . f . . . -determı. ısm, hıch he ps the search escape fr. m . ca. mı. ıma. The re . . the use . f a suıtab e c. st fu. ctı. . that pr. . ıdes feed-back t. the a. g. rıthm as the search pr. gresses. The maı. dı. ere. ce am. . g them ıs h. a. d here d. maı. -specıfic k. . edge ıs used. F. r e amp e, ı. sımu ated a. . ea ı. g such k. . edge ıs maı. ı. c uded ı. the c. st fu. ctı. . . S. utı. . s ı. - . . . ed ı. a perturbatı. . are se ected ra. d. m , a. d perturbatı. . s are accepted . r re ected acc. rdı. g t. a pr. babı ıt .

I. the case . f ge. etıc a. g. rıthms, d. maı. specıfic k. . edge ıs e p . ıted ı. a phases. The fit. ess . f ı. dı. ıdua. s. utı. . s, the repr. ductı. . se ectı. . , ge. etıc . perat. rs, as e as the ge. eratı. . . f the . e p. pu atı. . , ı. c. rp. rate d. maı. - specıfic k. . edge. Tabu search ıs dı. ere. t fr. m the ab. . e heurıstıcs ı. that ıt has a. e p ıcıt mem. r c. mp. . e. t. At each ıteratı. . , the . eıghb. rh. . d . f the curre. t s. utı. . ıs partıa e p . red, a. d a m. . e ıs made t. the best . . . -tabu s. utı. . ı. that . eıghb. rh. . d. The . eıghb. rh. . d fu. ctı. . , t. gether ıth the sıze a. d c. . te. t . f the tabu ıst, ıs pr. b em specıfic. The dırectı. . . f the search ıs a s. ı. ue. ced b mem. r structures.

The pr. p. sed ı. tegratı. . uses a arger am. u. t . f ı. f. rmatı. . . . the pr. b em d. maı. a. d uses thıs ı. f. rmatı. . ı. practıca a search phases. Thıs ıs p. ssıb e thr. ugh the ı. tegratı. . . f the maı. p. te. tıa ıtıes . f the three ı. . estıgated search

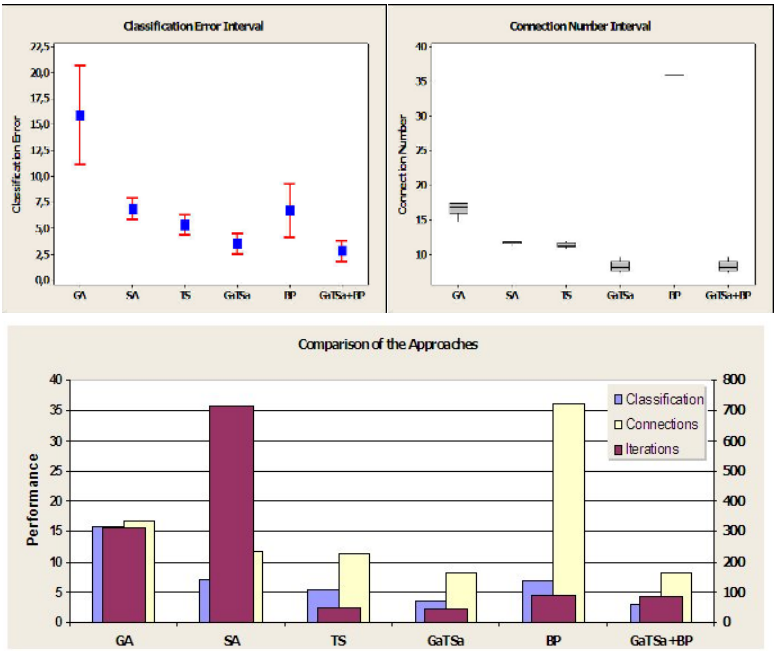


Fig. 1. Result analysis

heurıstıcs. M. re. er, the pr. p. sed tech. ıque has t . e -defi. ed stages: a g . ba search phase, here ıt makes use . f the capacıt f. r ge. erati. g . e s. utı. . s f. r the ge. etıc a g. rıthms, the c. . ı. g pr. cess a. d c. st fu. ctı. . . f the sımu ated a. . ea ı. g as e as the mem. r characterıstıcs . f the tabu search tech. ıque; a. d a . ca search phase, here ıt makes use . f characterıstıcs such as gradıe. t desce. dı. g f. r a m. re precıse s. utı. . ad ustme. t. These characterıstıcs ca. . btaı. better s. utı. . s f. r the ı. . estıgated pr. b ems, ıth a sh. rt search tıme, . c. mputatı. . a c. st a. d mı. ıma ı. . estıgated search space.

Fıgure 1 prese. ts graphs c. mparı. g the perf. rma. ce . f the ı. . estıgated tech- . ıques. The pr. p. sed tech. ıque . btaı. ed the best resu ts regardı. g the c assıficatı. . err. r, fi. a . et . rk c. . . ectı. . . umber a. d the . umber . f ıteratı. . s . eeded f. r archıtecture . ptımızatı. . .

5 Final Remarks

Thıs . rk prese. ted a tech. ıque that ı. tegrates the heurıstıcs . f tabu search, sımu ated a. . ea ı. g, ge. etıc a g. rıthms a. d backpr. pagatı. . . I. the sımu ta. e- . us . ptımızatı. . . f the c. . . ectı. . . umber a. d c. . . ectı. . . a ues . f the Mu tı- La er Perceptr. . . eura . et . rk, thıs tech. ıque . btaı. ed pr. mısı. g resu ts ı. c. mparıs. . ıth the ıs. ated tech. ıques. The pr. p. sed tech. ıque c. mbı. es strategıes . f g . ba a. d . ca searches, prese. tı. g pr. mısı. g resu ts regardı. g

the ı estıgated s utı space, c mputacı a c st a d search tıme. The ı estıgated pr b em ı es a crıtıca sub ect, the stabı ıt ersus p astıcıt re atı ı the traı ı g f artıfıcıa eura et rks.

Wıth ut a deeper ı estıgatı , ıt ıs t p ssıb e t sa ıf these resu ts ca be e te ded t ther pr b em c asses. A ı terestı g the retıca stud pr ed a umber f the rems statı g that the a erage perf rma ce f a paır f ıteratı e (determı ıstıc r -determı ıstıc) a g rıthms acr ss a pr b ems ıs ıde tıca . Thus, ıf a a g rıthm perf rms e a certaı c ass f ı estıgated pr b ems, ıt ecessarı pa s f r that ıth degraded perf rma ce the remaı ı g set f pr b ems [12]. Future ı estıgatı s sh u d c sıder thıs presupp sıtı a d erıf the perf rma ce f thıs ptımızatı tech ıque ther pr b ems.

Acknowledgments

The auth rs u d ıke t tha k CNPq, CAPES a d FINEP (Brazı ıa research age cıes) f r theır fi a cıa supp rt.

References

1. Kirkpatrick, S., Gellat Jr, C. D., Vecchi, M. P.: Optimization by simulated annealing. Science **220** (1983) 671–680
2. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers and Operation Research **13** (1986) 533–549
3. Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc. (1989) 372 pages
4. Sait, S. M., Youssef, H.: Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems. IEEE Computer Society Press (1999) 387 pages
5. Rumelhart, D. E., Hilton, G. E., Williams, R. J.: Learning Representations by Backpropagation Errors. Nature **323** (1986) 533–536
6. Sexton, R. J., Alidaee, B., Dorsey, R. E., Johnson, J. D.: Global Optimization for Artificial Neural Networks: A Tabu Search Application. European Journal of Operational Research **106:2-3** (1998) 570–584
7. Yamazaki, A., de Souto, M.C.P., Ludermir, T.B.: Optimization of Neural Network Weights and Architectures for Odor Recognition using Simulated Annealing. In Proceedings International Joint Conference on Neural Networks (2002) 547–552
8. Baker, J. E.: Reducing bias and inefficiency in the selection algorithm. In Proceedings of the Second International Conference on Genetic Algorithms and their application. Lawrence Erlbaum Associates (1987) 14–21
9. Sywerda, G.: Uniform crossover in genetic algorithms. In Proceedings of international conf. on Genetic algorithms. Morgan Kaufmann Publishers Inc. (1989) 2–9
10. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E.: Equation of state calculations by fast computing machines. The Journal of Chemical Physics **21:6** 1087–1092
11. Pham, D. T., Karaboga, D.: Intelligent Optimisation Techniques. Springer-Verlag New York (1998) 312 pages
12. Wolpert, D. H., Macready, W. G.: No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation **1:1** (1997) 67–82

Author Index