

Xue Li
Osmar R. Zaiane
Zhanhuai Li (Eds.)

LNAI 4093

Advanced Data Mining and Applications

Second International Conference, ADMA 2006
Xi'an, China, August 2006
Proceedings

 Springer

Lecture Notes in Artificial Intelligence 4093

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Xue Li Osmar R. Zaïane
Zhanhuai Li (Eds.)

Advanced Data Mining and Applications

Second International Conference, ADMA 2006
Xi'an, China, August 14-16, 2006
Proceedings

Series Editors

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

Volume Editors

Xue Li

The University of Queensland
School of Information Technology and Electronic Engineering
Queensland, Australia
E-mail: xueli@itee.uq.edu.au

Osmar R. Zaiane

University of Alberta, Canada
E-mail: zaiane@cs.ualberta.ca

Zhanhuai Li

Northwest Polytechnical University, China
E-mail: lizhh@nwpu.edu.cn

Library of Congress Control Number: 2006930269

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

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743

ISBN-10 3-540-37025-0 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-37025-3 Springer Berlin Heidelberg New York

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

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11811305 06/3142 5 4 3 2 1 0

Preface

The Second International Conference on Advanced Data Mining and Applications (ADMA) aimed at establishing its identity in the research community. The theme of ADMA is to focus on the innovative applications of data mining approaches to real-world problems that involve large data sets, incomplete and noisy data, or demand optimal solutions. Data mining is essentially a problem involving different knowledge of data, algorithms, and application domains. The first is about data that are regarded as the “first-class citizens” in application system development. Understanding data is always critical: their structures, high dimensionality, and their qualification and quantification issues. The second is about algorithms: their effectiveness, efficiency, scalability, and their applicability. Amongst a variety of applicable algorithms, selecting a right one to deal with a specific problem is always a challenge that demands contributions from the data mining research community. The third is on domain knowledge of applications. Without a good understanding of domain knowledge, data mining process is hardly able to avoid the GIGO (garbage in and garbage out) effect.

ADMA 2006 received 515 online submissions from 27 countries and regions. A screening process was applied before papers were assigned to reviewers. This eliminated approximately 5% of papers which were not relevant or not worthy for review. The rest of papers were peer reviewed by at least three reviewers consisting of international Program Committee members and external reviewers. We finally accepted 115 papers including 41 regular papers and 74 short papers, yielding a total acceptance rate of 22.3%. For quality control, we conducted a cross-check on the submitted papers for any possible duplicated submissions to four other related concurrent international conferences. Joint actions were taken by the Program Committee Co-chairs from the involved conferences to reject papers that were doubly submitted. In this case, 15 papers were rejected without review crossing PRICAI 2006, ICIC 2006, and ADMA 2006. Two papers were rejected after the acceptance notifications because they appeared in the acceptance list of ICNC-FSKD 2006. While this exercise was very time consuming, it was necessary and we hope that it will send out a message that double submissions are not tolerated in our research community.

The ADMA 2006 program highlights were three keynote speeches from outstanding researchers in advanced data mining and application areas: Usama Fayyad, Jiawei Han, and Ah Chung Tsoi. The conference also invited researchers from two international universities to report on their newest research findings.

August 2006

Xue Li
Osmar Zaïane
Zhanhuai Li

Conference Committee

ADMA 2006 was organized by the University of Queensland, Australia. The major sponsor was Xi'an Software Park, China (<http://www.xasoftpark.com/>). The other sponsor was WISE (Web Information Systems Engineering, <http://www.i-wise.org/>) Society. It was technically co-sponsored by IEEE Queensland Section. The local organization was undertaken by Xi'an Netec Network Technology Co. Ltd. (<http://www.netecweb.com/english/company.asp>).

Organization Committee

Conference Co-chairs	Kyu-Young Whang (Korea Advanced Institute of Science and Technology, Korea) Chengqi Zhang (University of Technology, Sydney, Australia)
Program Co-chairs	Xue Li (University of Queensland, Australia) Osmar Zaiane (University of Alberta, Canada) Zhanhuai Li (Northwest Polytechnical University, China)
Organizing Chairs	Ailiang Mao (Xi'an Software Park, China) Jian Chen (Xi'an Software Park-IBM Software Innovation Centre, China)
Publicity Chairs	Yonghong Peng (University of Bradford, UK) Jinyan Li (Institute for Infocomm Research, Singapore)
Web Master	Roozbeh Derakhshan (University of Queensland, Australia)

Program Committee Members

Viorel Ariton	Danubius University, Galati, Romania
Michael Bain	University of New South Wales, Australia
Elena Baralis	Politecnico di Torino, Italy
Petr Berka	University of Economics, Czech Republic
Michael R. Berthold	University of Constance, Germany
Fernando Berzal	University of Granada, Spain
Francesco Bonchi	CNR Pisa, Italy
Jean-Francois Boulicaut	INSA Lyon, France
Rui Camacho	University of Porto, Portugal
Xiaochun Cheng	Middlesex University, UK
Krzysztof Cios	University of Colorado, USA

Bruno Cremilleux	Universite de Caen, France
Luc Dehaspe	PharmaDM, Belgium
Zhaoyang Dong	University of Queensland, Australia
Hao Fan	Wuhan University, China
Hongjian Fan	University of Melbourne Australia
Joao Gama	University of Porto, Portugal
Dragan Gamberger	Rudjer Boskovic Institute, Croatia
Jean-Gabriel Ganascia	Université Pierre et Marie Curie, France
Junbin Gao	The University of New England, Australia
Christophe Giraud-Carrier	Brigham Young University, USA
Raul Giraldez Rojo	University of Seville, Spain
Bart Goethals	University of Antwerp, Belgium
Vladimir Gorodetsky	The Russian Academy of Science, Russia
Yi Hong	University of South Australia, Australia
Zhanyi Hu	Academy of China, China
Jimmy Huang	York University, Canada
Ping Jiang	Bradford University, UK
Alipio Jorge	University of Porto, Portugal
Mehmed Kantardzic	University of Louisville, USA
Eamonn Keogh	University of California, Riverside, USA
Tim Kovacs	University of Bristol, UK
Adam Krzyzak	Concordia University, Montreal, Canada
Andrew Kusiak	University of Iowa, USA
Longin Jan Latecki	Temple University Philadelphia, USA
Andre Ponce Leao	University of SãoPaulo, Brazil
Gang Li	Deakin University, Australia
Qing Li	Hong Kong City University, Hong Kong, China
Xiaoli Li	Institute for Infocomm Research, Singapore
Xuemin Lin	University of New South Wales, Australia
Wanquan Liu	Curtin University of Technology, Australia
Giuseppe Manco	National Research Council of Italy, Italy
Dunja Mladenic	Jozef Stefan Institute, Slovenia
Iveta Mrazova	Charles University, Prague, Czech Republic
Olfa Nasraoui	University of Louisville, USA
Daniel Neagu	University of Bradford, UK
Claire Nedellec	Laboratoire Mathématique, Informatique et Gènome, France
Arlindo Oliveira	Technical University of Lisbon, Portugal
Yonghong Peng	University of Bradford, UK
Jan Rauch	University of Economics, Prague, Czech Republic
Zbigniew W. Ras	University of North Carolina, USA
Cesar Rego	University of Mississippi, USA
Christophe Rigotti	INSA de Lyon, France
Joseph Roure	University of Mataro, Spain
Yucel Saygin	Sabanci University, Turkey
Marc Sebban	University Jean Monnet, France

Giovanni Semeraro	University of Bari, Italy
Seyed A. Shahrestani	University of Western Sydney, Sydney, Australia
Carlos Soares	University of Porto, Portugal
Ah-Hwee Tan	Nanyang Technological University, Singapore
Kay Chen Tan	National University of Singapore, Singapore
Kok Kiong Tan	National University of Singapore, Singapore
Arthur Tay	National University of Singapore, Singapore
Luis Torgo	University of Porto, Portugal
Shusaku Tsumoto	Shimane Medical University, Japan
Brijesh Verma	Central Queensland University, Australia
Ricardo Vilalta	University of Houston, USA
Paul Vitanyi	CWI and University of Amsterdam, The Netherlands
Dianhui Wang	La Trobe University, Melbourne, Australia
Ke Wang	Simon Fraser University, Canada
Shuliang Wang	Wuhan University, China
Wei Wang	University of North Carolina, USA
Hau San Wong	City University of Hong Kong, China
Dash Wu	University of Toronto, Canada
Dongming Xu	University of Queensland, Australia
Qiang Yang	Hong Kong University of Science and Technology, China
Zijiang Yang	York University, Canada
Mao Ye	Electronic Science and Technology, China
Jie Yin	Hong Kong University of Science and Technology, China
Gerson Zaverucha	Federal University of Rio de Janeiro, Brazil
Sarah Zelikovitz	College of Staten Island, NY
Shichao Zhang	University of Technology, Sydney, Australia
Shuigeng Zhou	Fudan University, China
Zhi-Hua Zhou	Nanjing University, China
Zhanli Zhu	Xi'an Shiyong University, China

External Reviewers

Mona Salman Hatam Al-Jiboori	Sultan Qaboos University, Oman
Domingo S. Rodríguez Baena	University of Seville, Spain
Beatriz Pontes Balanza	University of Seville, Spain
Miguel Bugalho	IST/INESC-ID, Portugal
Alexandra Carvalho	IST/INESC-ID, Portugal
Ming Chang	University of Queensland, Australia
Dingyi Chen	University of Queensland, Australia
Luís Pedro Coelho	IST/INESC-ID, Portugal
Marian Craciun	University “Dunarea de Jos” of Galati, Romania
Roozbeh Derakhshan	University of Queensland, Australia

Yi Ding	University of Queensland, Australia
Xiaoshu Hang	Deakin University, Australia
Petr Hoffmann	Charles University, Prague, Czech Republic
Pavel Jiroutek	Charles University, Prague, Czech Republic
Alexandre Francisco	IST/INESC-ID, Portugal
Ana Teresa Freitas	IST/INESC-ID, Portugal
Gongde Guo	University of Bradford, UK
Yang Lan	University of Bradford, UK
Sara C. Madeira	UBI/INESC-ID/IST, Portugal
Ladan Malazizi	University of Bradford, UK
André Luís Martins	IST/INESC-ID, Portugal
Frantisek Mraz	Charles University, Prague, Czech Republic
Juggapong Natwichai	University of Queensland, Australia
Son Nguyen Nhu	University of Queensland, Australia
Jia Rong	Deakin University, Australia
Luís Silveira Russo	IST/INESC-ID, Portugal
Sule Simsek	University of Missouri-Rolla, USA
Xingzhi Sun	University of Queensland, Australia
Osman Okyar Tahaoglu	Dokuz Eylul University, Turkey
Han (Carol) Tang	University of Missouri-Rolla, USA
Li-Shiang Tsay	Hampton University, USA
Yiqing Tu	Deakin University, Australia
Angelina Tzacheva	University of South Carolina Upstate, USA
Xin Yan	University of Queensland, Australia
Ling Zhuang	Deakin University, Australia

Table of Contents

Invited Papers

Warehousing and Mining Massive RFID Data Sets <i>Jiawei Han, Hector Gonzalez, Xiaolei Li, Diego Klabjan</i>	1
Self-organising Map Techniques for Graph Data Applications to Clustering of XML Documents <i>A.C. Tsoi, M. Hagenbuchner, A. Sperduti</i>	19
Finding Time Series Discords Based on Haar Transform <i>Ada Wai-chee Fu, Oscar Tat-Wing Leung, Eamonn Keogh, Jessica Lin</i>	31
Learning with Local Drift Detection <i>João Gama, Gladys Castillo</i>	42

Association Rules

A Fast Algorithm for Maintenance of Association Rules in Incremental Databases <i>Xin Li, Zhi-Hong Deng, Shiwei Tang</i>	56
Extending OLAP with Fuzziness for Effective Mining of Fuzzy Multidimensional Weighted Association Rules <i>Mehmet Kaya, Reda Alhajj</i>	64
Incremental Maintenance of Association Rules Based on Multiple Previously Mined Results <i>Zhuohua Duan, Zixing Cai, Yan Lv</i>	72
Mining and Validation of Localized Frequent Web Access Patterns with Dynamic Tolerance <i>Olfa Nasraoui, Suchandra Goswami</i>	80
SA-IFIM: Incrementally Mining Frequent Itemsets in Update Distorted Databases <i>Jinlong Wang, Congfu Xu, Hongwei Dan, Yunhe Pan</i>	92

Study of Positive and Negative Association Rules Based on Multi-confidence and Chi-Squared Test
Xiangjun Dong, Fengrong Sun, Xiqing Han, Ruilian Hou 100

Efficiently Mining Maximal Frequent Mutually Associated Patterns
Zhongmei Zhou, Zhaohui Wu, Chunshan Wang, Yi Feng 110

Efficiently Mining Mutually and Positively Correlated Patterns
Zhongmei Zhou, Zhaohui Wu, Chunshan Wang, Yi Feng 118

Classification

ComEnVprs: A Novel Approach for Inducing Decision Tree Classifiers
Shuqin Wang, Jinmao Wei, Junping You, Dayou Liu 126

Towards a Rough Classification of Business Travelers
Rob Law, Thomas Bauer, Karin Weber, Tony Tse 135

Feature Extraction Based on Optimal Discrimination Plane in ECG Signal Classification
Dingfei Ge, Xiao Qu 143

Music Style Classification with a Novel Bayesian Model
Yatong Zhou, Taiyi Zhang, Jiancheng Sun 150

Classification of Polarimetric SAR Data Based on Multidimensional Watershed Clustering
Wen Yang, Hao Wang, Yongfeng Cao, Haijian Zhang 157

An Effective Combination Based on Class-Wise Expertise of Diverse Classifiers for Predictive Toxicology Data Mining
Daniel Neagu, Gongde Guo, Shanshan Wang 165

Robust Collective Classification with Contextual Dependency Network Models
Yonghong Tian, Tiejun Huang, Wen Gao 173

User-Centered Image Semantics Classification
Hongli Xu, De Xu, Fangshi Wang 181

A Performance Study of Gaussian Kernel Classifiers for Data Mining Applications
Miyoung Shin 189

TTLSC – Transductive Total Least Square Model for Classification and Its Application in Medicine <i>Qun Song, Tian Min Ma, Nikola Kasabov</i>	197
Forecasting Electricity Market Price Spikes Based on Bayesian Expert with Support Vector Machines <i>Wei Wu, Jianzhong Zhou, Li Mo, Chengjun Zhu</i>	205
Integrating Local One-Class Classifiers for Image Retrieval <i>Yiqing Tu, Gang Li, Honghua Dai</i>	213
Incremental Discretization for Naïve-Bayes Classifier <i>Jingli Lu, Ying Yang, Geoffrey I. Webb</i>	223
Distance Guided Classification with Gene Expression Programming <i>Lei Duan, Changjie Tang, Tianqing Zhang, Dagang Wei, Huan Zhang</i>	239
Research on Multi-valued and Multi-labeled Decision Trees <i>Hong Li, Rui Zhao, Jianer Chen, Yao Xiang</i>	247
Clustering	
A Spatial Clustering Algorithm Based on SOFM <i>Zhong Qu, Lian Wang</i>	255
Mining Spatial-temporal Clusters from Geo-databases <i>Min Wang, Aiping Wang, Anbo Li</i>	263
A Fuzzy Subspace Algorithm for Clustering High Dimensional Data <i>Guojun Gan, Jianhong Wu, Zijiang Yang</i>	271
Robust Music Information Retrieval on Mobile Network Based on Multi-Feature Clustering <i>Won-Jung Yoon, Sanghun Oh, Kyu-Sik Park</i>	279
Joint Cluster Based Co-clustering for Clustering Ensembles <i>Tianming Hu, Liping Liu, Chao Qu, Sam Yuan Sung</i>	284
Mining Gait Pattern for Clinical Locomotion Diagnosis Based on Clustering Techniques <i>Guandong Xu, Yanchun Zhang, Rezaul Begg</i>	296
Combining Multiple Clusterings Via k-Modes Algorithm <i>Huilan Luo, Fansheng Kong, Yixiao Li</i>	308

HOV³: An Approach to Visual Cluster Analysis
Ke-Bing Zhang, Mehmet A. Orgun, Kang Zhang 316

A New Fuzzy Co-clustering Algorithm for Categorization of Datasets
 with Overlapping Clusters
William-Chandra Tjhi, Lihui Chen 328

Quantum-Behaved Particle Swarm Optimization Clustering Algorithm
Jun Sun, Wenbo Xu, Bin Ye 340

Clustering Mixed Data Based on Evidence Accumulation
Huilan Luo, Fansheng Kong, Yixiao Li 348

Mining Maximal Local Conserved Gene Clusters from Microarray Data
Yuhai Zhao, Guoren Wang, Ying Yin, Guangyu Xu 356

Novel Algorithms

A Novel P2P Information Clustering and Retrieval Mechanism
Huaxiang Zhang, Peide Liu 364

Keeping Track of Customer Life Cycle to Build Customer Relationship
Sung Ho Ha, Sung Min Bae 372

Mining of Flexible Manufacturing System Using Work Event Logs and
 Petri Nets
Hesuan Hu, Zhiwu Li, Anrong Wang 380

Improved Genetic Algorithm for Multiple Sequence Alignment Using
 Segment Profiles (GASP)
*Yanping Lv, Shaozi Li, Changle Zhou, Wenzhong Guo,
 Zhengming Xu* 388

A Novel Visual Clustering Algorithm for Finding Community in
 Complex Network
Shuzhong Yang, Siwei Luo, Jianyu Li 396

Self-Organizing Network Evolving Model for Mining Network
 Community Structure
Bo Yang 404

An Interactive Visualization Environment for Data Exploration Using
 Points of Interest
David Da Costa, Gilles Venturini 416

Forecasting the Volatility of Stock Price Index <i>Tae Hyup Roh</i>	424
ExMiner: An Efficient Algorithm for Mining Top-K Frequent Patterns <i>Tran Minh Quang, Shigeru Oyanagi, Katsuhiko Yamazaki</i>	436
Learning Bayesian Networks Structure with Continuous Variables <i>Shuang-Cheng Wang, Xiao-Lin Li, Hai-Yan Tang</i>	448
A Unified Strategy of Feature Selection <i>Peng Liu, Naijun Wu, Jiaxian Zhu, Junjie Yin, Wei Zhang</i>	457
Experimental Comparison of Feature Subset Selection Using GA and ACO Algorithm <i>Keunjoon Lee, Jinu Joo, Jihoon Yang, Vasant Honavar</i>	465
OMVD: An Optimization of MVD <i>Zhi He, Shengfeng Tian, Houkuan Huang</i>	473
ZED: Explaining Temporal Variations in Query Volume <i>Maojin Jiang, Shlomo Argamon, Abdur Chowdhury, Kush Sidhu</i>	485
An Effective Multi-level Algorithm for Bisecting Graph <i>Ming Leng, Songnian Yu</i>	493
A New Polynomial Time Algorithm for Bayesian Network Structure Learning <i>Sanghack Lee, Jihoon Yang, Sungyong Park</i>	501
Personalized Recommendation Based on Partial Similarity of Interests <i>Ming-Hua Yang, Zhi-Min Gu</i>	509
A Fast Implementation of the EM Algorithm for Mixture of Multinomials <i>Jan Peter Patist</i>	517
A Novel Approach to Pattern Recognition Based on PCA-ANN in Spectroscopy <i>Xiaoli Li, Yong He</i>	525
Semi-supervised Dynamic Counter Propagation Network <i>Yao Chen, Yuntao Qian</i>	533
The Practical Method of Fractal Dimensionality Reduction Based on Z-Ordering Technique <i>Guanghui Yan, Zhanhuai Li, Liu Yuan</i>	542

Feature Selection for Complex Patterns <i>Peter Schenkel, Wanqing Li, Wanquan Liu</i>	550
Naïve Bayesian Tree Pruning by Local Accuracy Estimation <i>Zhipeng Xie</i>	558
A New Visualization Method for Patent Map: Application to Ubiquitous Computing Technology <i>Jong Hwan Suh, Sang Chan Park</i>	566
Local Linear Logistic Discriminant Analysis with Partial Least Square Components <i>Jangsun Baek, Young Sook Son</i>	574
Activity Mining: Challenges and Prospects <i>Longbing Cao</i>	582
Finding the Optimal Cardinality Value for Information Bottleneck Method <i>Gang Li, Dong Liu, Yiqing Tu, Yangdong Ye</i>	594
A New Algorithm for Enumerating All Maximal Cliques in Complex Network <i>Li Wan, Bin Wu, Nan Du, Qi Ye, Ping Chen</i>	606
Modeling and Mining the Rule Evolution <i>Ding Pan</i>	618
Knowledge Reduction in Inconsistent Decision Tables <i>Qihe Liu, Leiting Chen, Jianzhong Zhang, Fan Min</i>	626
Text Mining	
Semantic Scoring Based on Small-World Phenomenon for Feature Selection in Text Mining <i>Chong Huang, Yonghong Tian, Tiejun Huang, Wen Gao</i>	636
A Comparative Study on Text Clustering Methods <i>Yan Zheng, Xiaochun Cheng, Ronghuai Huang, Yi Man</i>	644
Concept Based Text Classification Using Labeled and Unlabeled Data <i>Ping Gu, Qingsheng Zhu, Xiping He</i>	652
Learning Semantic User Profiles from Text <i>M. Degemmis, P. Lops, G. Semeraro</i>	661

Multimedia Mining

Audiovisual Integration for Racquet Sports Video Retrieval <i>Yaqin Zhao, Xianzhong Zhou, Guizhong Tang</i>	673
A Correlation Approach for Automatic Image Annotation <i>David R. Hardoon, Craig Saunders, Sandor Szedmak, John Shawe-Taylor</i>	681

Sequential Data Mining and Time Series Mining

Fast Discovery of Time-Constrained Sequential Patterns Using Time-Indexes <i>Ming-Yen Lin, Sue-Chen Hsueh, Chia-Wen Chang</i>	693
Multi-dimensional Sequential Pattern Mining Based on Concept Lattice <i>Yang Jin, Wanli Zuo</i>	702
Mining Time-Delayed Coherent Patterns in Time Series Gene Expression Data <i>Linjun Yin, Guoren Wang, Keming Mao, Yuhai Zhao</i>	711
Mining Delay in Streaming Time Series of Industrial Process <i>Haijie Gu, Gang Rong</i>	723
Segmental Semi-Markov Model Based Online Series Pattern Detection Under Arbitrary Time Scaling <i>Guangjie Ling, Yuntao Qian, Sen Jia</i>	731
Diagnosis of Inverter Faults in PMSM DTC Drive Using Time-Series Data Mining Technique <i>Dan Sun, Jun Meng, Zongyuan He</i>	741
Applications of Data Mining Time Series to Power Systems Disturbance Analysis <i>Jun Meng, Dan Sun, Zhiyong Li</i>	749
Mining Compressed Sequential Patterns <i>Lei Chang, Dongqing Yang, Shiwei Tang, Tengjiao Wang</i>	761
Effective Feature Preprocessing for Time Series Forecasting <i>Jun Hua Zhao, ZhaoYang Dong, Zhao Xu</i>	769
On Similarity of Financial Data Series Based on Fractal Dimension <i>Jian-rong Hou, Hui Zhao, Pei Huang</i>	782

Web Mining

A Hierarchical Model of Web Graph

*Jie Han, Yong Yu, Chenxi Lin, Dingyi Han,
Gui-Rong Xue* 790

Web Scale Competitor Discovery Using Mutual Information

Rui Li, Shenghua Bao, Jin Wang, Yuanjie Liu, Yong Yu 798

Biomedical Mining

Co-expression Gene Discovery from Microarray for Integrative Systems Biology

Yutao Ma, Yonghong Peng 809

Cardiovascular Disease Diagnosis Method by Emerging Patterns

*Heon Gyu Lee, Kiyong Noh, Bum Ju Lee, Ho-Sun Shon,
Keun Ho Ryu* 819

DNA Microarray Data Clustering by Hidden Markov Models and Bayesian Information Criterion

*Phasit Charoenkwan, Aompilai Manorat, Jeerayut Chaijaruanich,
Sukon Prasitwattanaseree, Sakarindr Bhumiratana* 827

Application of Factor Analysis on *Mycobacterium Tuberculosis* Transcriptional Responses for Drug Clustering, Drug Target, and Pathway Detections

*Jeerayut Chaijaruanich, Jamlong Khamphachua,
Sukon Prasitwattanaseree, Saradee Warit,
Prasit Palittapongarnpim* 835

First Steps to an Audio Ontology-Based Classifier for Telemedicine

Cong Phuong Nguyen, Ngoc Yen Pham, Eric Castelli 845

Obstacles and Misunderstandings Facing Medical Data Mining

Ashkan Sami 856

SVM-Based Tumor Classification with Gene Expression Data

Shulin Wang, Ji Wang, Huowang Chen, Boyun Zhang 864

GEPCLASS: A Classification Rule Discovery Tool Using Gene Expression Programming

Wagner R. Weinert, Heitor S. Lopes 871

Advanced Applications

CBR-Based Knowledge Discovery on Results of Evolutionary Design of Logic Circuits <i>Shuguang Zhao, Mingying Zhao, Jin Li, Change Wang</i>	881
Data Summarization Approach to Relational Domain Learning Based on Frequent Pattern to Support the Development of Decision Making <i>Rayner Alfred, Dimitar Kazakov</i>	889
Extreme Value Dependence in Problems with a Changing Causation Structure <i>Marlon Núñez, Rafael Morales</i>	899
A Study on Object Recognition Technology Using PCA in the Variable Illumination <i>Jong-Min Kim, Hwan-Seok Yang</i>	911
Pattern Recurring in Three-Dimensional Graph Based on Data Mining <i>Yanbing Liu, Menghao Wang</i>	919
Mining the Useful Skyline Set Based on the Acceptable Difference <i>Zhenhua Huang, Wei Wang</i>	927
Modeling Information-Sharing Behaviors in BitTorrent System Based on Real Measurement <i>Jinkang Jia, Changjia Chen</i>	934
Financial Distress Prediction Based on Similarity Weighted Voting CBR <i>Jie Sun, Xiao-Feng Hui</i>	947
Customer Churn Prediction by Hybrid Model <i>Jae Sik Lee, Jin Chun Lee</i>	959
Base Vector Selection for Kernel Matching Pursuit <i>Qing Li, Licheng Jiao</i>	967
<i>WaveSim</i> Transform for Multi-channel Signal Data Mining Through Linear Regression PCA <i>R. Pradeep Kumar, P. Nagabhushan</i>	977
Research on Query-by-Committee Method of Active Learning and Application <i>Yue Zhao, Ciwen Xu, Yongcun Cao</i>	985

Traffic Management Genetic Algorithm Supporting Data Mining and QoS in Sensor Networks
Yantao Pan, Wei Peng, Xicheng Lu 992

Comparison of Data Pre-processing in Pattern Recognition of Milk Powder Vis/NIR Spectra
Haiyan Cen, Yidan Bao, Min Huang, Yong He 1000

Semi-automatic Hot Event Detection
Tingting He, Guozhong Qu, Siwei Li, Xinhui Tu, Yong Zhang, Han Ren 1008

Security and Privacy Issues

Profile-Based Security Against Malicious Mobile Agents
Hua Li, Glena Greene, Rafael Alonso 1017

A Comprehensive Categorization of DDoS Attack and DDoS Defense Techniques
Usman Tariq, ManPyo Hong, Kyung-suk Lhee 1025

Retracted: Structural Analysis and Mathematical Methods for Destabilizing Terrorist Networks Using Investigative Data Mining
Nasrullah Memon, Henrik Legind Larsen 1037

Alert Correlation Analysis in Intrusion Detection
Moon Sun Shin, Kyeong Ja Jeong 1049

Spatial Data Mining

OSDM: Optimized Shape Distribution Method
Ashkan Sami, Ryoichi Nagatomi, Makoto Takahashi, Takeshi Tokuyama 1057

View-Angle of Spatial Data Mining
Shuliang Wang, Haning Yuan 1065

Streaming Data Mining

Maintaining Moving Sums over Data Streams
Tzu-Chiang Wu, Arbee L.P. Chen 1077

MFIS—Mining Frequent Itemsets on Data Streams
Zhi-jun Xie, Hong Chen, Cuiping Li 1085

Improving the Performance of Data Stream Classifiers by Mining
 Recurring Contexts
Yong Wang, Zhanhuai Li, Yang Zhang, Longbo Zhang, Yun Jiang 1094

Erratum

Structural Analysis and Mathematical Methods for Destabilizing
 Terrorist Networks Using Investigative Data Mining
Nasrullah Memon, Henrik Legind Larsen E1

Author Index 1107

Warehousing and Mining Massive RFID Data Sets

Jiawei Han, Hector Gonzalez, Xiaolei Li, and Diego Klabjan

University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{hanj, hagonzal, xli10, klabjan}@uiuc.edu

Abstract. Radio Frequency Identification (RFID) applications are set to play an essential role in object tracking and supply chain management systems. In the near future, it is expected that every major retailer will use RFID systems to track the movement of products from suppliers to warehouses, store backrooms and eventually to points of sale. The volume of information generated by such systems can be enormous as each individual item (a pallet, a case, or an SKU) will leave a trail of data as it moves through different locations. We propose two data models for the management of this data. The first is a *path cube* that preserves object transition information while allowing multi-dimensional analysis of path dependent aggregates. The second is a *workflow cube* that summarizes the major patterns and significant exceptions in the flow of items through the system. The design of our models is based on the following observations: (1) items usually move together in large groups through early stages in the system (e.g., distribution centers) and only in later stages (e.g., stores) do they move in smaller groups, (2) although RFID data is registered at the primitive level, data analysis usually takes place at a higher abstraction level, (3) many items have similar flow patterns and only a relatively small number of them truly deviate from the general trend, and (4) only non-redundant flow deviations with respect to previously recorded deviations are interesting. These observations facilitate the construction of highly compressed RFID data warehouses and the exploration of such data warehouses by scalable data mining. In this study we give a general overview of the principles driving the design of our framework. We believe warehousing and mining RFID data presents an interesting application for advanced data mining.

1 Introduction

Radio Frequency Identification (RFID) is a technology that allows a sensor (RFID reader) to read, from a distance and without line of sight, a unique identifier that is provided (via a radio signal) by an “inexpensive” tag attached to an item. RFID offers a possible alternative to bar code identification systems and it facilitates applications like item tracking and inventory management in the supply chain. The technology holds the promise to streamline supply chain management, facilitate routing and distribution of products, and reduce costs by improving efficiency.

Large retailers like Walmart, Target, and Albertsons have already begun implementing RFID systems in their warehouses and distribution centers, and are requiring their suppliers to tag products at the pallet and case levels. Individual tag prices are expected to fall from around 25 cents per unit to 5 cents per unit by 2007. At that price level, we

can expect tags to be placed at the individual item level for many products. The main challenge then becomes how can companies handle and interpret the enormous volume of data that an RFID application will generate. Venture Development Corporation [11], a research firm, predicts that when tags are used at the item level, Walmart will generate around 7 terabytes of data every day. Database vendors like Oracle, IBM, Teradata, and some startups are starting to provide solutions to integrate RFID information into enterprise data warehouses.

Example. Suppose a retailer with 3,000 stores sells 10,000 items a day per store. Assume that we record each item movement with a tuple of the form: $(EPC, location, time)$, where EPC is an Electronic Product Code which uniquely identifies each item¹. If each item leaves only 10 traces before leaving the store by going through different locations, this application will generate at least 300 million tuples per day. A manager may ask queries on the duration of paths like (Q_1) : “List the average shelf life of dairy products in 2003 by manufacturer”, or on the structure of the paths like (Q_2) : “What is the average time that it took coffee-makers to move from the warehouse to the shelf and finally to the checkout counter in January of 2004?”, or on the major flow characteristics like (Q_3) : “Is there a correlation between the time spent at quality control and the probability of returns for laptops manufactured in Asia?”.

Such enormous amount of low-level data and flexible high-level queries pose great challenges to traditional relational and data warehouse technologies since the processing may involve retrieval and reasoning over a large number of inter-related tuples through different stages of object movements. No matter how the objects are sorted and clustered, it is difficult to support various kinds of high-level queries in a uniform and efficient way. A nontrivial number of queries may even require a full scan of the entire RFID database.

1.1 Path Cube

The *path cube* compresses and aggregates the paths traversed by items in the system along time, location, and product related dimensions. This cube will allow a wide range of OLAP queries to be answered efficiently. Our design is based on the following key observations.

- We need to eliminate the redundancy present in RFID data. Each reader provides tuples of the form $(EPC, location, time)$ at fixed time intervals. When an item stays at the same location, for a period of time, multiple tuples will be generated. We can group these tuples into a single one of the form $(EPC, location, time_in, time_out)$.
- Items tend to move and stay together through different locations. For example, a pallet with 500 cases of CDs may arrive at the warehouse; from there cases of 50 CDs may move to the shelf; and from there packs of 5 CDs may move to the checkout counter. We can register a single *stay* tuple of the form $(EPC\ list, location, time_in, time_out)$ for the CDs that arrive in the same pallet and stay together in the warehouse, and thus generate an 80% space saving.

¹ We will use the terms EPC and RFID tag interchangeably throughout the paper.

- We can gain further compression by reducing the size of the EPC lists in the *stay* records by grouping items that move to the same locations. For example, if we have a *stay* record for the 50 CDs that stayed together at the warehouse, and that the CDs moved in two groups to shelf and truck locations. We can replace the list of 50 EPCs in the stay record for just two *generalized identifiers (gids)* which in turn point to the concrete EPCs. In addition to the compression benefits, we can gain query processing speedup by assigning path-dependent names to the gids.
- Most search or mining queries are likely to be at a high level of abstraction, and will only be interested in the low-level individual items if they are associated with some interesting patterns discovered at a high level.

1.2 Workflow Cube

The *workflow cube* aggregates item flows at different levels of abstraction of the item-related dimensions and the location-related dimensions. The measure of the *workflow cube* is a compressed probabilistic workflow, *i.e.*, each cell in the cube will contain a workflow computed on the paths aggregated in the cell. It differs from the *path cube* in two major ways: (1) it does not consider absolute time, and only relative durations, and (2) the measure of each cell is a workflow and not a scalar aggregate such as sum or count, which would be typical in the *path cube*.

Commodity flows can be analyzed from the perspective of paths (*path view*) and the abstraction level at which path stages appear or from the perspective of item related dimensions (*item view*) and their abstraction levels. Figure 1 presents a path (seen in the middle of the figure) aggregated to two different abstraction levels. The path at the top of the figure shows the individual locations inside a store, while it collapses locations that belong to transportation; this may be interesting to a store manager. The path at the bottom of the figure on the other hand, collapses locations that belong to stores, and keeps individual locations that belong to transportation; this view may be interesting to a transportation manager in the company. An orthogonal view into RFID commodity flows is related to items themselves. This is a view much closer to traditional data cubes. An item can have a set of dimension describing its characteristics, *e.g.*, product, brand, manufacturer. Each of these dimensions has an associated concept hierarchy.

The key challenge in constructing a *workflow cube* based on a set of RFID paths is to devise an efficient method to compute summaries of commodity flows for those item views and path views that are interesting to the different data analysts utilizing the application. The proposed construction method is based on the following optimizations:

- **Reduction of the size of the workflow cube by exploring two strategies.** The first is to compute only those cells that contain a minimum number of paths (iceberg condition). This makes sense as each workflow is a probabilistic model that can be used to conduct statistically significant analysis only if there is enough data to support it. The second strategy is to compute only workflows that are non-redundant given higher abstraction level workflows. For example, if the flow patterns of 2% milk are similar to those of milk (under certain threshold), then by registering just the high level workflow we can infer that one for 2% milk, *i.e.*, we expect any low level concept to behave in a similar way to its parents, and only when this behavior is truly different, we register such information.

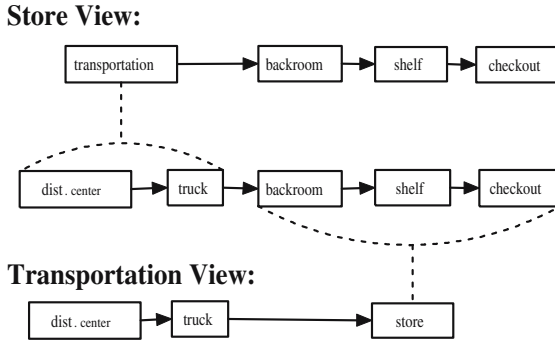


Fig. 1. Path views: The same path can be seen at two different abstraction levels

- **Shared computation.** We explore efficient computation of the *workflow cube* by sharing the computation of frequent cells and frequent path segments simultaneously. Similar to shared computation of multiple cuboids in BUC-like computation [3], we propose to compute frequent cells in the *workflow cube* and frequent path segments aggregated at every interesting abstraction level simultaneously. Shared computation minimizes the number of scans of the path database by maximizing the amount of information collected during each scan.
- **Pruning of the search space using both the path and item views.** To speed up cube computation, we use pre-counting of high abstraction level itemsets that will help us prune a large portion of the candidate space without having to collect their counts. For example if we detect that the stage shelf is not frequent in general, we know that for no particular duration it can be frequent; or if a store location is not frequent, no individual location within the store can be frequent.

The rest of the paper is organized as follows. Section 2 presents the structure of the input RFID data. Section 3 presents the architecture of the *path cube*. Section 4 Gives an overview of the architecture of the *workflow cube*. Section 5 reports experimental results. We discuss the related work in Section 6, outline some promising research directions on mining RFID data in Section 7, and conclude our study in Section 8.

2 RFID Data

Data generated from an RFID application can be seen as a stream of RFID tuples of the form $(EPC, location, time)$, where *EPC* is the unique identifier read by an RFID reader, *location* is the place where the RFID reader scanned the item, and *time* is the time when the reading took place. Tuples are usually stored according to a time sequence. A single EPC may have multiple readings at the same location, each reading is generated by the RFID reader scanning for tags at fixed time intervals or on a continuous basis.

In order to reduce the large amount of redundancy in the raw data, data cleaning should be performed. The output after data cleaning is a set of clean stay records of the form $(EPC, location, time_{in}, time_{out})$ where *time_{in}* is the time when the object enters the location, and *time_{out}* is the time when the object leaves the location.

Data cleaning of stay records can be accomplished by sorting the raw data on EPC and time, and generating *time_in* and *time_out* for each location by merging consecutive records for the same object staying at the same location.

Furthermore, for the purpose of constructing the *workflow cube* we are not interested in absolute time but only relative durations and thus we can rewrite the cleansed RFID database as a set of paths of the form $(EPC : (l_1, d_1) (l_2, d_2) \dots (l_k, d_k))$, where l_i is the i -th location in the path traveled by the item identified by *EPC*, and d_i is the total time that the item stayed at l_i .

The duration that an item spent at a location is a continuous attribute. In order to simplify the model further we can discretize all the distinct durations for each location into a fixed number of clusters. We call such a path database with discretized durations *clustered path-database*.

3 Architecture of the Path Cube

Before we describe our proposed architecture for the *path cube*, it is important to describe why a traditional data cube model would fail on such data. Suppose we view the cleansed RFID data as the fact table with dimensions $(EPC, location, time_in, time_out : measure)$. The data cube will compute all possible group-bys on this fact table by aggregating records that share the same values (or any *) at all possible combinations of dimensions. If we use count as measure, we can get for example the number of items that stayed at a given location for a given month. The problem with this form of aggregation is that it does not consider links between the records. For example, if we want to get the number of items of type “dairy product” that traveled from the distribution center in Chicago to stores in Urbana, we cannot get this information. We have the count of “dairy products” for each location but we do not know how many of those items went from the first location to the second. We need a more powerful model capable of aggregating data while preserving its path-like structure.

We propose an RFID warehouse architecture that contains a fact table, *stay*, composed of cleansed RFID records; an information table, *info*, that stores path-independent information for each item, *i.e.*, SKU information that is constant regardless of the location of the item, such as manufacturer, lot number, and color; and a *map* table that links together different records in the fact table that form a path. We call the *stay*, *info*, and *map* tables aggregated at a given abstraction level an *RFID-Cuboid*.

The main difference between the RFID warehouse and a traditional warehouse is the presence of the map table linking records from the fact table (*stay*) in order to preserve the original structure of the data.

The computation of cuboids in the *path cube* (*RFID-Cuboids*) is more complex than that of regular cuboids as we will need to aggregate the data while preserving the structure of the paths at different abstraction levels.

From the data storage and query processing point of view the RFID warehouse can be viewed as a multi-level database. The raw RFID repository resides at the lowest level, on its top are the cleansed RFID database, the minimum abstraction level *RFID-Cuboids* and a sparse subset of the full cuboid lattice composed of frequently queried (popular) *RFID-Cuboids*.

3.1 Key Ideas of RFID Data Compression

Even with the removal of data redundancy from RFID raw data, the cleansed RFID database is usually still enormous. Here we explore several general ideas for constructing a highly compact RFID data warehouse.

Taking advantage of bulky object movements

Since a large number of items travel and stay together through several stages, it is important to represent such a collective movement by a single record no matter how many items were originally collected. As an example, if 1,000 boxes of milk stayed in location loc_A between time t_1 (time_in) and t_2 (time_out), it would be advantageous if only one record is registered in the database rather than 1,000 individual RFID records. The record would have the form: $(gid, prod, loc_A, t_1, t_2, 1000)$, where 1,000 is the count, $prod$ is the product id, and gid is a generalized id which will not point to the 1,000 original EPCs but instead point to the set of new gids which the current set of objects move to. For example, if this current set of objects were split into 10 partitions, each moving to one distinct location, gid will point to 10 distinct new gids, each representing a record. The process iterates until the end of the object movement where the concrete EPCs will be registered. By doing so, no information is lost but the number of records to store such information is substantially reduced.

Taking advantage of data generalization

Since many users are only interested in data at a relatively high abstraction level, data compression can be explored to group, merge, and compress data records. For example, if the minimal granularity of time is hour, then objects moving within the same hour can be seen as moving together and be merged into one movement. Similarly, if the granularity of the location is shelf, objects moving to the different layers of a shelf can be seen as moving to the same *shelf* and be merged into one. Similar generalization can be performed for products (e.g., merging different sized milk packages) and other data as well.

Taking advantage of the merge and/or collapse of path segments

In many analysis tasks, certain path segments can be ignored or merged for simplicity of analysis. For example, some non-essential object movements (e.g., from one shelf to another in a store) can be completely ignored in certain data analysis. Some path segments can be merged without affecting the analysis results. For store managers, merging all the movements before the object reaches the store could be desirable. Such merging and collapsing of path segments may substantially reduce the total size of the data and speed-up the analysis process.

3.2 RFID-CUBOID

With the data compression principles in mind, we propose *RFID-Cuboid*, a data structure for storing aggregated data in the RFID warehouse. Our design ensures that the data are disk-resident, summarizing the contents of a cleansed RFID database in a compact yet complete manner while allowing efficient execution of both OLAP and tag-specific queries.

The *RFID-Cuboid* consists of three tables: (1) *Info*, which stores product information for each RFID tag, (2) *Stay*, which stores information on items that stay together at

a location, and (3) *Map*, which stores path information necessary to link multiple stay records.

Information Table

The information table stores path-independent dimensions, such as product name, manufacturer, product price and product category. Each dimension can have an associated concept hierarchy. All traditional OLAP operations can be performed on these dimensions in conjunction with various RFID-specific analysis. For example, one could drill-down on the product category dimension from “clothing” to “shirts” and retrieve shipment information only on shirts.

Entries in *Info* are records of the form: $\langle (EPCList), (d_1, \dots, d_m):(m_1, \dots, m_i) \rangle$, where the code list contains a set of items that share the same values for dimensions d_1, \dots, d_m , and m_1, \dots, m_i are measures of the given items, e.g., price.

Stay Table

As mentioned in the introduction, items tend to move and stay together through different locations. Compressing multiple items that stay together at a single location is vital in order to reduce the enormous size of the cleansed RFID database. In real applications items tend to move in large groups. At a distribution center there may be tens of pallets staying together, and then they are broken into individual pallets at the warehouse level. Even if products finally move at the individual item level from a shelf to the checkout counter, our stay compression will save space for all previous steps taken by the item.

Each entry in *Stay* is a record of the form: $\langle (gids, location, time_in, time_out) : (m_1, \dots, m_k) \rangle$, where *gids* is a set of generalized record ids each pointing to a list of RFID tags or lower level *gids*, *location* is the location where the items stayed together, *time_in* is the time when the items entered the location, and *time_out* the time when they left. If the items did not leave the location, *time_out* is *NULL*. Finally, m_1, \dots, m_n are the measures recorded for the stay, e.g., count, average time at *location*, and the maximal time at *location*.

Map Table

The *Map* table is an efficient structure that allows query processing to link together stages that belong to the same path in order to perform structure-aware analysis, which could not be answered by a traditional data warehouse. There are two main reasons for using a *Map* table instead of recording the complete EPC lists at each stage: (1) data compression, by reduction of the size of the EPC lists size at each location; and (2) query processing efficiency, by encoding the path of a group of items in their generalized identifier.

The map table contains mappings from higher level *gids* to lower level ones or EPCs. Each entry in *Map* is a record of the form: $\langle gid, (gid_1, \dots, gid_n) \rangle$, meaning that, *gid* is composed of all the EPCs pointed to by gid_1, \dots, gid_n . The lowest level *gids* will point directly to individual items.

In order to facilitate query processing we will assign path-dependent labels to high level *gids*. The label will contain one identifier per location traveled by the items in the *gid*.

3.3 Path Cuboid Lattice

In order to provide fast response to queries specified at various levels of abstraction, it is important to pre-compute some *RFID-Cuboids* at different levels of the concept hierarchies for the dimensions of the *Info* and *Stay* tables. It is obviously too expensive to compute all the possible generalizations, and partial materialization is a preferred choice. This problem is analogous to determining which set of cuboids in a data cube to materialize in order to answer OLAP queries efficiently given the limitations on storage space and precomputation time. This issue has been studied extensively in the data cube research [5,10] and the principles are generally applicable to the selective materialization of *RFID-Cuboids*.

In our design, we suggest to compute a set of *RFID-Cuboids* at the minimal interesting level at which users will be interested in inquiring the database, and a small set of higher level structures that are frequently requested and that can be used to quickly compute non-materialized *RFID-Cuboids*.

An *RFID-Cuboid* residing at the minimal interesting level will be computed directly from the cleansed RFID database and will be the lowest cuboid that can be queried unless one has to dig directly into the detail cleansed data in some very special cases.

3.4 Query Processing

In this section we discuss the implementation of the basic OLAP operations, *i.e.*, drill-down, roll-up, slice, and dice, applied to the *path cube*, and introduce a new operation, *path selection*, relevant to the paths traveled by items.

Given the very large size and high dimensionality of the RFID warehouse we can only materialize a small fraction of the total number of *RFID-Cuboids*. We will compute the *RFID-Cuboid* that resides at the minimum abstraction layer that is interesting to users, and those *RFID-Cuboids* that are frequently requested. When a roll-up or drill-down operation requires an *RFID-Cuboid* that has not yet been materialized, it would have to be computed on the fly from an existing *RFID-Cuboid* that is close to the required one but at a lower abstraction level. The slice and dice operations can be implemented efficiently by using relational query execution and optimization techniques.

Path Selection

Path queries, which ask about information related to the structure of object traversal paths, are unique to the RFID warehouse since the concept of object movements is not modeled in traditional data warehouses. It is essential to allow users to inquire about an aggregate measure computed based on a predefined sequence of locations (path). One such example could be: “*What is the average time for milk to go from farms to stores in Illinois?*”.

Queries on the paths traveled by items are fundamental to many RFID applications and will be the building block on top of which more complex data mining operators can be implemented. We will illustrate this point with a real example. The United States government is currently in the process of requiring the containers arriving into the country, by ship, to carry an RFID tag. The information can be used to determine if the path traveled by a given container has deviated from its historic path. This application may need to first execute a path-selection query across different time periods, and then use outlier detection and clustering to analyze the relevant paths.

In order to answer a path query we first select the *gids* for the stay records that match the conditions for the initial and final stages of the query expression. For example, g_{start} may look like $\langle 1.2, 8.3.1, 3.4 \rangle$ and g_{end} may look like $\langle 1.2.4.3, 4.3, 3.4.3 \rangle$. We then compute the pairs of *gids* from g_{start} that are a prefix of a *gid* in g_{end} . We get the pairs $\langle (1.2, 1.2.4.3), (3.4, 3.4.3) \rangle$. For each pair we then retrieve all the stay records. The pair $(1.2, 1.2.4.3)$ would require us to retrieve stay records that include *gids* 1.2, 1.2.4, and 1.2.4.3. Finally, we verify that each of these records matches the selection conditions for each $stage_i$ and for *Info*, and add those paths to the answer set.

4 Architecture of the Workflow Cube

The *workflow cube*, as in standard OLAP, will be composed of cuboids that aggregate item flows at a given abstraction level. The *workflow cube* differs from the traditional data cube in two major ways. First, the measure of each cell will not be a simple aggregate but a commodity *workflow* that captures the major movement trends and significant deviations for the subset of objects in the cell. Second, each *workflow* itself can be viewed at multiple levels by changing the level of abstraction path stages. The *workflow cube* also differs from the *path cube* in that it only considers relative duration and not absolute time in its analysis of paths. This distinction is useful in building a statistical model of object flows.

4.1 Probabilistic Workflow

A duration independent workflow is a tree where each node represents a location and edges correspond to transitions between locations. All common path prefixes appear in the same branch of the tree. Each transition has an associated probability, which is the percentage of items that took the transition represented by the edge. For every node we also record a termination probability, which is the percentage of paths that terminate at the location associated with the node.

We have several options to incorporate duration information into a duration independent workflow, the most direct way is to create nodes for every combination of location and duration. This option has the disadvantage of generating very large workflows. A second option is to annotate each node in the duration independent workflow with a distribution of possible durations at the node. This approach keeps the size of the workflow manageable and captures duration information for the case when (i) the duration distribution between locations is independent, e.g., the time that milk spends at the shelf is independent of the time it spent in the store backroom; and (ii) transitions probabilities are independent of duration, e.g., the probability of a box of milk which transits from the shelf to the checkout counter does not depend on the time it spent at the backroom.

There are cases when conditions (i) and (ii) do not hold, e.g., a product that spends a long time at a quality control station may increase its probability of moving to the return counter location at a retail store. In order to cover these cases we propose to use a new model that not only records duration and transition distributions at each node, but also stores information on significant deviations in duration and transition probabilities given frequent path prefixes to the node. A prefix to a node is a sequence of $(location, duration)$ pairs that appear in the same branch as the node but before

it. The construction of such workflow requires two parameters, ϵ that is the minimum deviation of a duration or transition probability required to record an exception, and δ the minimum support required to record a deviation. The purpose of ϵ is to record only deviations that are truly interesting in that they significantly affect the probability distribution induced by the workflow; and the purpose of δ to prevent the exceptions in the workflow to be dominated by statistical noise in the path database.

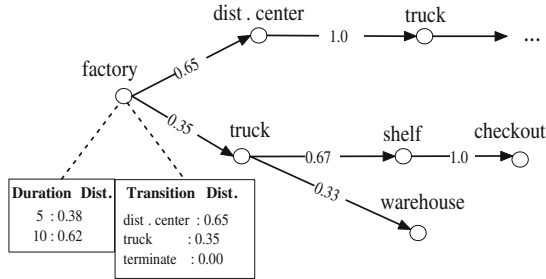


Fig. 2. Workflow

The workflow in Figure 2 also registers significant exceptions to duration and transition probabilities (not shown in the figure), e.g., the transition probability from the truck to the warehouse, coming from the factory, is in general 33%, but that probability is 50% when we stay for just 1 hour at the truck location. Similarly we can register exceptions for the distribution of durations at a location given previous durations, e.g., items in the distribution center spend 1 hour with probability 20% and 2 hours with probability 80%, but if an item spent 5 hours at the factory, the distribution changes and the probability of staying for 2 hours in the distribution center becomes 100%.

Computing a workflow can be done efficiently by (1) constructing a prefix tree for the path database, (2) annotating each node with duration and transition probabilities, and (3) mining the path database for frequent paths with minimum support δ , and checking if those paths create exceptions that deviate by more than ϵ from the general probability. Steps (1) and (2) can be done with a single scan of the path database, and for step (3) we can use any existing frequent pattern mining algorithm.

4.2 Workflow Cuboid Lattice

Each cuboid in the *workflow cube* will reside at a certain level of abstraction in the *Item Lattice* and *Path Lattice*. The *Item Lattice* represents all the possible levels of abstraction of the dimensions used to describe items, e.g., product, manufacturer, and color. This lattice is the same that one would encounter on traditional data cubes. The *Path Lattice* represents all possible levels of abstraction of the locations and durations of the paths in the database, and it is helpful in expanding only those portions of the paths that are interesting to data analysts while collapsing the rest of the locations.

Aggregation along the path abstraction lattice is unique to *workflow cubes* and is quite different from the type of aggregation performed in a regular data cube. In a data cube, an aggregated cell contains a measure on the subset of tuples from the fact

table that share the same values on every aggregated dimension. When we do path aggregation, the dimensions from the fact table remain unchanged, but it is the measure of the cell itself that changes. This distinct property requires us to develop new methods to construct a *workflow cube* that has aggregation on both item and path dimensions.

4.3 Workflow Redundancy

The workflow registered for a given cell in a *workflow cube* may not provide new information on the characteristics of the data in the cell, if the cells at a higher abstraction level on the item lattice, and the same abstraction level on the path lattice, can be used to derive the workflow in the cell. For example, if we have a workflow G_1 for milk, and a workflow G_2 from milk 2% (milk is an ancestor of milk 2% in the item abstraction lattice), and $G_1 = G_2$, we view that G_2 is redundant, as it can be inferred from G_1 . Registering only non-redundant cells not only allows significant compression but also provides important insight into the relationship of flow patterns from high to low levels of abstraction, and can facilitate the discovery of exceptions in multi-dimensional space.

4.4 Iceberg Workflow Cube

A workflow is a statistical model that describes the flow behavior of objects given a collection of paths. If the data set on which the workflow is computed is very small, the workflow may not be useful in conducting data analysis. Each probability in the model will be supported by such a small number of observations and it may not be an accurate estimate of the true probability. In order to minimize this problem, we will materialize only cells in the *workflow cube* that contain at least δ paths.

Iceberg cubes can be computed efficiently by using apriori pruning of infrequent cells. We can materialize the cube from low dimensionality to high dimensionality. If at some point a low level cell is not frequent, we do not need to check the frequency of any specialization of the cell.

5 Performance Study

In this section, we perform an experimental evaluation of the compression power and query processing performance of the *path cube*; we also report on the speedup gained by using the proposed techniques to compute the *workflow cube*.

5.1 Experimental Setup

The path databases used for our experiments were generated using a synthetic path generator that simulates the movement of items through the supply chain of a large retailer. The simulator varies the level of path bulkiness (\mathcal{B}) by changing the number of items that move together at each stage in the supply chain. In general we assume that items move in larger groups near the start of the supply chain (e.g. factories) and smaller groups near the end (e.g. shelves and checkout counters). Each path dependent dimension (i.e. locations and time), and path independent dimension (e.g. product, manufacturer, price) has an associated concept hierarchy; we vary the number of distinct

values and skew at each level of the concept hierarchies to change the distribution of frequent cells in the *workflow cube*.

For the *path cube* experiments we compare three distinct methods to represent a cuboid: (1) *clean*, which uses the cleansed path database, (2) *nomap*, which uses the stay and information tables as described in the paper but instead of using gids it directly stores EPC lists, and (3) *map* which uses the stay, information, and map tables. For the *workflow cube* experiments we compare three competing techniques used to compute the frequent cells and frequent path segments necessary to construct the *workflow cube*: (1) *shared* which is an algorithm proposed in the paper and that implements simultaneous mining of frequent cells and frequent path segments at all abstraction levels while performing apriori pruning, (2) *cubing* which is a modified version of of BUC [3] to compute the iceberg cube on the path independent dimensions and then called Apriori [2] to mine frequent path segments in each cell, and (3) *basic* is the same algorithm as *shared* except that we do not perform any candidate pruning.

As a notational convenience, we use the following symbols to denote certain data set parameters. For the *path cube* we use: $\mathcal{B} = (s_1, \dots, s_k)$ for path bulkiness, \mathcal{P} for the number of products, and k for the average path length. For the *workflow cube* we use: \mathcal{N} for the number of paths, δ for minimum support (iceberg condition), and d for the number of path independent dimensions.

5.2 Path Cube Compression

The *RFID-Cuboids* form the basis for future query processing and analysis. As mentioned previously, the advantage of these data structures is that they aggregate and collapse many records in the cleansed RFID database. Here, we examine the effects of this compression on different data sets.

Figure 3 shows the size of the cleansed RFID database (*clean*) compared with the *map* and *nomap RFID-Cuboids*. The data sets contains 1,000 distinct products, traveling in groups of 500, 150, 40, 8, and 1 through 5 path stages, and 500 thousand to 10 million cleansed RFID records. The *RFID-Cuboid* is computed at the same level of abstraction of the cleansed RFID data, and thus the compression is lossless. As it can be seen from Figure 3 the *RFID-Cuboid* that uses *map* has a compression power of around 80% while the one that uses EPC lists has a compression power of around 65%.

5.3 Path Cube Query Processing

A major contribution of the *path cube* is the ability to efficiently answer many types of queries at various levels of aggregation. We assume that for each of the scenarios we have a B+Tree on each of the dimensions. In the case of the *map cuboid* the index points to a list of gids matching the index entry. In the case of the *nomap cuboid* and the cleansed database the index points to the tuple (*RFID tag, record id*). This is necessary as each RFID tag can be present in multiple records. The query answering strategy used for the *map cuboid* is the one presented in Section 3.4. The strategy for the other two cases is to retrieve the (*RFID tag, record id*) pairs matching each component of the query, intersecting them, and finally retrieving the relevant records.

Figure 4 shows the effect of different cleansed database sizes on query processing. The *map cuboid* outperforms the cleansed database by several orders of magnitude, and

most importantly the query answer time is independent of database size. The nomap cuboid is significantly faster than the cleansed data but it suffers from having to retrieve very long RFID lists for each stage. The map cuboid benefits from using very short gid lists, and using the path-dependent gid naming scheme that facilitates determining if two stay records form a path without retrieving all intermediate stages.

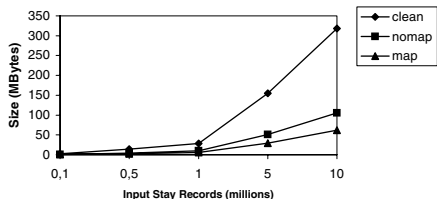


Fig. 3. Compression vs. Cleansed Data Size. $\mathcal{P} = 1000$, $\mathcal{B} = (500, 150, 40, 8, 1)$, $k = 5$.

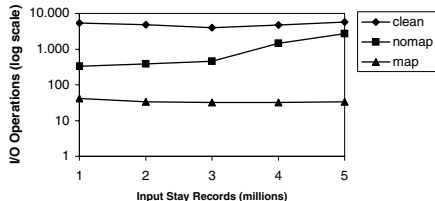


Fig. 4. Query Processing. $\mathcal{P} = 1000$, $\mathcal{B} = (500, 150, 40, 8, 1)$, $k = 5$.

5.4 Workflow Cube Construction Time vs. Database Size

In this experiment we vary the size of path database, from 100,000 paths to 1,000,000 paths. In Figure 5 we can see that the performance of shared and cubing is quite close for smaller data sets but as we increase the number of paths the runtime of shared increases with a smaller slope than that of cubing. This may be due to the fact that as we increase the number of paths the data set becomes denser BUC slows down. Another influencing factor in the difference in slopes is that as the data sets become denser cubing needs to invoke the frequent pattern mining algorithm for many more cells, each with a larger number of paths. We were able to run the basic algorithm for 100,000 and 200,000 paths, for other values the number of candidates was so large that they could not fit into memory. This highlights the importance of the candidate pruning optimizations.

5.5 Workflow Cube Construction Time vs. Minimum Support

In this experiment we constructed a path database with 100,000 paths and 5 path independent dimensions. We varied the minimum support from 0.3% to 2.0%. In Figure 6 we can see that shared outperforms cubing and basic. As we increase minimum support the performance of all the algorithms improves as expected. Basic improves faster than the other two, this is due to the fact that fewer candidates are generated at higher support levels, and thus optimizations based on candidate pruning become less critical. For every support level we can see that shared outperforms cubing, but what is more important we see that shared improves its performance faster than cubing. The reason is that as we increase support shared will quickly prune large portions of the path space, while cubing will repeatedly check this portions for every cell it finds to be frequent.

5.6 Workflow Cube Construction Time vs. Number of Dimensions

In this experiment we kept the number of paths constant at 100,000 and the support at 1%, and varied the number of dimensions from 2 to 10. The datasets used for this

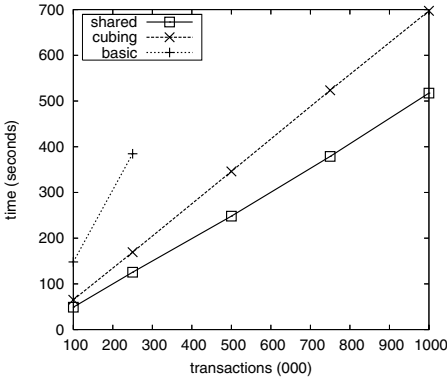


Fig. 5. Time vs. Database Size ($\delta = 0.01$, $d = 5$)

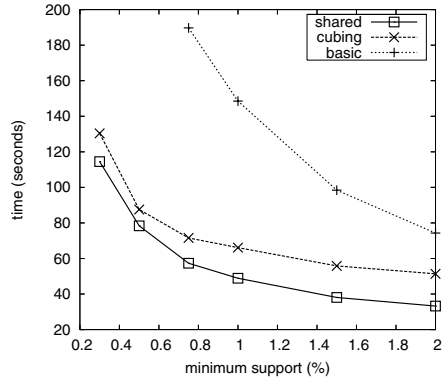


Fig. 6. Time vs. Minimum Support ($\mathcal{N} = 100,000$, $d = 5$)

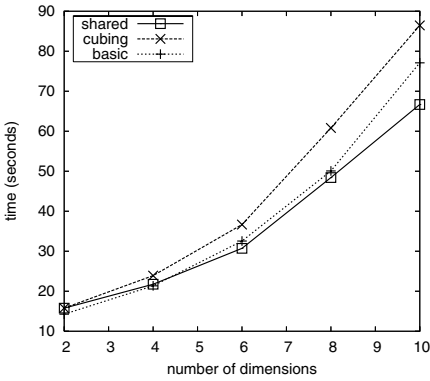


Fig. 7. Time vs. Number of Dimensions ($\mathcal{N} = 100,000$, $\delta = 0.01$)

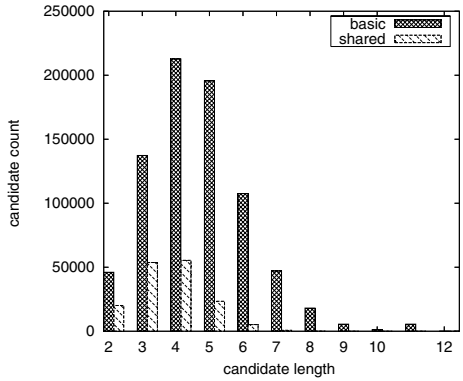


Fig. 8. Pruning Power ($\mathcal{N} = 100,000$, $\delta = 0.01$, $d = 5$)

experiment were quite sparse to prevent the number of frequent cells to explode at higher dimension cuboids. The sparse nature of the datasets makes all the algorithms achieve a similar performance level. We can see in Figure 7 that both shared and cubing are able to prune large portions of the cube space very soon, and thus performance was comparable. Similarly basic was quite efficient as the number of candidates was small and optimizations based on candidate pruning did not make a big difference given that the number of candidates was small to begin with.

5.7 Workflow Cube Construction Pruning Power

This experiment shows the effectiveness of the proposed optimizations to prune unpromising candidates from consideration in the mining process. We compare the number of candidates that the basic and shared algorithms need to count for each pattern

length. We can see in figure 8 that shared is able to prune a very significant number of candidates from consideration. Basic on the other hand has to collect counts for a very large number of patterns that end up being infrequent, this increases the memory usage and slows down the algorithm. This issue was evidenced in other experiments when the number of candidates was so large that basic could not fit them into memory. We can also see in the figure that shared considers patterns only up to length 8, while basic considers patterns all the way to length 12. This is because basic is considering long transactions that include items and their ancestors.

6 Related Work

Software research into management of RFID data is divided into three areas. The first is concerned with secure collection and management of online tag related information [14,15]. The second is cleaning of errors present in RFID data due to error inaccuracies, [13,12]. The third is related to the creation of multi-dimensional warehouses capable of providing OLAP operations over a large RFID data set [7,6].

The design of the *path cube* and the *workflow cube* shares many common principles with the traditional data cube [8]. They both aggregate data at different levels of abstraction in multi-dimensional space. Since each dimension has an associated concept hierarchy, both can be (at least partially) modeled by a *Star* schema. The problem of deciding which cuboids to construct in order to provide efficient answers to a variety of queries specified at different abstraction levels is analogous to the problem of partial data cube materialization studied in [10,16]. However, the *path cube* differs from a traditional data cube in that it also models *object transitions* in multi-dimensional space.

The *workflow cube* also shares common principles with the above lines of research but it differs from it in that our measure is a complex probabilistic model and not just an aggregate such as count or sum, and that our aggregates deal with two interrelated abstraction lattices, one for item dimensions and the other for path dimensions. Induction of workflows from RFID data sets shares many characteristics with the problem of process mining [18]. Workflow induction, the area possibly closest to our work, studies the problem of discovering the structure of a workflow from event logs. [1] first introduced the problem of process mining and proposed a method to discover workflow structure, but for the most part their methods assumes no duplicate activities in the workflow and does not take activity duration into account, which is a very important aspect of RFID data. Another area of research very closed to flowgraph construction is that of grammar induction [4,17], the idea is to take as input a set of strings and infer the probabilistic deterministic finite state automaton (PDFA) that generated the strings. This approach differs from ours in that it does not consider exceptions to transition probability distributions, or duration distributions at the nodes.

7 Towards Fruitful Research on Mining RFID Data Warehouses

The systematic data cleaning, integration, compression and aggregation in the construction of RFID data warehouses provide a valuable source of integrated and compressed data as well as a powerful infrastructure for mining RFID data. As indicated in our

scattered discussions in the previous sections, frequent pattern analysis, data flow aggregation and analysis, and exception analysis have been integrated in the process of construction of such data warehouses. With the construction and incremental maintenance of such an integrated data warehouse, it is possible to systematically develop scalable and versatile data mining functions in such an RFID data warehouse system.

Almost all kinds of data mining functions that can be associated with traditional data warehouses [9] can find their applications at mining RFID data warehouses. However, due to the tremendous amount of FRID data and its special analysis tasks, there are some particular mining tasks that may play more important roles in RFID data mining than others. Since RFID data mining is just an emerging research direction, here we only briefly outline a few such tasks. Interested researchers may likely find exciting new directions by their own research. First, due to the hardly avoidable inaccuracy or errors in RFID reading at various circumstances, RFID data likely contain noise, missing data, or erroneous readings. Data mining will help build cleansed and integrated data warehouse by finding regularities of data movements, cross-checking exceptions and outliers, and performing inference based on expert-provided or discovered rules. Second, with the bulky RFID data, one may like to know summary, statistics, and the characteristic and discriminant features of the FRID data in multi-dimensional space, especially those related to time, location, product category, and so on. A data cube-based aggregation and statistical analysis will be an important component in RFID data analysis. Third, although it is important to understand the general characteristics and trends of RFID data, it is important to detect outliers and exceptions, especially those related to data flow and object movements. Fourth, due to the incremental updates of bulky RFID data, the mining for changes of data in multi-dimensional space in an incremental manner could be another importance theme of study. Finally, due to the multi-dimensional nature of FRID data warehouse, it is critical to develop scalable and OLAP-based multi-dimensional data mining methods so that the analysis can be performed in an online analytical processing manner, either by human interaction or by automated processes.

Based on the above analysis, we believe the tremendous amount of RFID data provides an exciting research frontier in data mining as well as a fertile ground for further research across multiple disciplines.

8 Conclusions

We have proposed two novel models to warehouse RFID data that allow high-level analysis to be performed efficiently and flexibly in multi-dimensional space. The first model, the *path cube* takes advantage of bulky object movements to construct a highly compressed representation of RFID data that can be used to compute a variety of path dependent aggregates. The second model, the *workflow cube* summarizes major flows and significant flow exceptions in an RFID data set, and can be used to discover interesting patterns in the movement of commodities through an RFID application.

The *path cube* is composed of a hierarchy of compact summaries (*RFID-Cuboids*) of the RFID data aggregated at different abstraction levels where data analysis can take place. Each *RFID-Cuboid* records item movements in the *Stay*, *Info*, and *Map* tables that take advantage of the fact that individual items tend to move and stay together

(especially at higher abstraction levels) to collapse multiple movements into a single record without loss of information.

The *workflow cube* is a model useful in analyzing item flows in an RFID application by summarizing item paths along the dimensions that describe the items, and the dimensions that describe the path stages. Each cell has a probabilistic workflow as measure, this workflow is a concise representation of general flow trends and significant deviations from the trends. The *workflow cube* facilitates the discovery of trends in the movement of items at different abstraction levels. It also provides views of the data that are tailored to the needs of each user.

Notice that both of these models work well when our underlying assumption of bulky object movements, and major flow patterns with a small number of exceptions are valid. This fits a good number of RFID applications, such as supply chain management. However, there are also other applications where RFID data may not have such characteristics. We believe that further research is needed to construct efficient models for such applications. Finally, we view mining patterns, rules and outliers from RFID data in RFID data warehouses as a promising research frontier with broad applications.

Acknowledgement. The work was supported in part by the U.S. National Science Foundation NSF IIS-03-08215/05-13678 and NSF BDI-05-15813. 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.

References

1. R. Agrawal, D. Gunopulos, and F. Leymann. Mining process models from workflow logs. In *Proc. 1998 Int. Conf. Extending Database Technology (EDBT'98)*, pages 469–483, Valencia, Spain, Mar. 1998.
2. R. Agrawal and R. Srikant. Fast algorithm for mining association rules in large databases. In *Research Report RJ 9839*, IBM Almaden Research Center, San Jose, CA, June 1994.
3. K. Beyer and R. Ramakrishnan. Bottom-up computation of sparse and iceberg cubes. In *Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99)*, pages 359–370, Philadelphia, PA, June 1999.
4. R. C. Carrasco and J. Oncina. Learning stochastic regular grammars by means of a state merging method. In *Proc. 1994 Int. Col. Grammatical Inference (ICGI'94)*, pages 139–152, Alicante, Spain, Sept. 1994.
5. S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26:65–74, 1997.
6. H. Gonzalez, J. Han, and X. Li. Flowcube: Constructing RFID flowcubes for multi-dimensional analysis of commodity flows. In *Proc. 2006 Int. Conf. Very Large Data Bases (VLDB'06)*, Seoul, Korea, Sept. 2006.
7. H. Gonzalez, J. Han, X. Li, and D. Klabjan. Warehousing and analysis of massive RFID data sets. In *Proc. 2006 Int. Conf. Data Engineering (ICDE'06)*, Atlanta, Georgia, April 2006.
8. J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. *Data Mining and Knowledge Discovery*, 1:29–54, 1997.
9. J. Han and M. Kamber. *Data Mining: Concepts and Techniques* (2nd ed.). Morgan Kaufmann, 2006.

10. V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'96)*, pages 205–216, Montreal, Canada, June 1996.
11. Venture development corporation (vdc). In <http://www.vdc-corp.com/>.
12. S. R. Jeffery, G. Alonso, and M. J. Franklin. Adaptive cleaning for RFID data streams. In *Technical Report UCB/EECS-2006-29*, EECS Department, University of California, Berkeley, March 2006.
13. S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom. A pipelined framework for online cleaning of sensor data streams. In *Proc. 2006 Int. Conf. Data Engineering (ICDE'06)*, Atlanta, Georgia, April 2006.
14. S. Sarma, D. L. Brock, and K. Ashton. The networked physical world. In *White paper, MIT Auto-ID Center*, <http://archive.epcglobalinc.org/publishedresearch/MIT-AUTOID-WH-001.pdf>, 2000.
15. S. E. Sarma, S. A. Weis, and D. W. Engels. RFID systems, security & privacy implications. In *White paper, MIT Auto-ID Center*, <http://archive.epcglobalinc.org/publishedresearch/MIT-AUTOID-WH-014.pdf>, 2002.
16. A. Shukla, P. M. Deshpande, and J. F. Naughton. Materialized view selection for multidimensional datasets. In *Proc. 1998 Int. Conf. Very Large Data Bases (VLDB'98)*, pages 488–499, New York, NY, Aug. 1998.
17. F. Thollard, P. Dupont, and C. dela Higuera. Probabilistic DFA inference using kullback-leibler divergence and minimality. In *Proc. 2000 Int. Conf. Machine Learning (ICML'00)*, pages 975–982, Stanford, CA, June 2000.
18. W. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.*, 16:1128–1142, 2004.

Self-organising Map Techniques for Graph Data Applications to Clustering of XML Documents

A.C. Tsoi¹, M. Hagenbuchner², and
A. Sperduti³

¹ Monash e-Research Centre, Monash University
Victoria 3800, Australia

`ahchung.tsoi@adm.monash.edu.au`

² Faculty of Informatics, University of Wollongong,
Wollongong NSW 2522, Australia

`markus@uow.edu.au`

³ Dipartimento di Matematica Pura ed Applicata, University of Padova,
Via G.B. Belzoni, 7, 35131 Padova, Italy

`sperduti@math.unipd.it`

Abstract. In this paper, neural network techniques based on Kohonen's self-organising map method which can be trained in an unsupervised fashion and applicable to the processing of graph structured inputs are described. Then it is shown how such techniques can be applied to the problems of clustering of XML documents.

1 Introduction

Neural networks have been one of the main techniques used widely in data mining. There are a number of popular neural network architectures, e.g. multilayer perceptrons [5], self organising maps [3], support vector machines [6]. However, most of these techniques have been applied to problems in which the inputs are vectors. In other words, the inputs to these neural network architectures are expressed in the form of vectors, often in fixed dimensions. In case the inputs are not suitably expressed in the form of vectors, they are made to conform to the fixed dimension vectorial format. For example, it is known that an image may be more conveniently expressed in the form of a graph, for instance, the image of a house can be expressed as a tree, with the source node (level 0) being the house, windows, walls, and doors expressed as leaves (level 1), and details of windows, walls and doors being expressed as leaves (level 2) of those leaves located in level 1, etc. These nodes are described by attributes (features, which may express colour, texture, dimensions) and their relationships with one another are described by links. Such inputs can be made to conform to a vectorial format if we "flatten" the structure and instead represent the information in each node in the form of a vector, and obtain the aggregate vector by concatenating the vectors together. Such techniques have been prevalent in the application of neural network architectures to these problems.

Recently, there have been some attempt in preserving the graph structured data as long as we can before applying the neural network technique. For example, in support vector machines, there have been some work in expressing the graph structured data in the form of string kernels [4], or spectrum kernels [4], and then use the “kernel trick” [6] in using the support vector machine machinery to process the data. Alternatively, another way to process the data is to preserve the graph structured data format, and modify the neural network techniques to process graph structured data. In this paper, we will not consider support vector machine further, and we will concentrate only on ways to modify a classic neural network technique, self-organising maps, so that it can accept graph structured inputs.

The structure of this paper is as follows: in Section 2, we will describe ways in which the classic self-organising map idea can be extended to consider graph structured data, in what we called a self-organising map for structured data (SOM-SD) technique and contextual self-organising map for structured data (CSOM-SD) technique. In Section 3, we will describe applications of these techniques to clustering XML documents. Some conclusions will be drawn in Section 4.

2 Self-organizing Map for Structured Data

The self-organising map concept [3] was developed to help identify clusters in multidimensional, say, p -dimensional datasets. The SOM does this by effectively packing the p -dimensional dataset onto a q -dimensional display plane, where we assume for simplicity $q = 2$ throughout this paper. The SOM consists of discretising the display space into $N \times N$ grid points, each grid point is associated with a p -dimensional vector, referred to in this paper, as an artificial neuron, or simply a neuron¹. Each neuron has an associated p -dimensional vector, called a codebook vector. This codebook vector \mathbf{m} has the same dimension as the i -th input vector \mathbf{x}_i . The neurons on the map are bound together by a topology, which is often either hexagonal or rectangular. The contents of these vectors are updated with each presentation of samples from the p -dimensional original data set. The contents of these vectors encode the relationships (distances) among the p -dimensional data. The result is that data items that were “similar” or “close” to each other in the original multidimensional data space are then mapped onto nearby areas of the 2-dimensional display space. Thus SOM is a topology-preserving map as there is a topological structure imposed on the nodes in the network. A topological map is simply a mapping that preserves neighborhood relations.

In general, the SOM is a model which is trained on a set of examples in an unsupervised fashion as follows:

For every input vector \mathbf{x}_i in a training set, obtain the best matching codebook by computing

¹ This is called a neuron for historical reasons.

$$c = \arg \min_j \|\mathbf{x}_i - \mathbf{m}_j\| \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean norm.

After the best matching unit \mathbf{m}_c is found, the codebook vectors are updated. \mathbf{m}_c itself as well as its topological neighbours are moved closer to the input vector in the input space i.e. the input vector attracts them. The magnitude of the attraction is governed by a learning rate α and by a neighborhood function $f(\Delta_{jc})$, where Δ_{jc} is the topological distance between \mathbf{m}_c and \mathbf{m}_j . As the learning proceeds and new input vectors are given to the map, the learning rate gradually decreases to zero according to a specified learning rate function type. Along with the learning rate, the neighborhood radius decreases as well. The codebooks on the map are updated as follows:

$$\Delta \mathbf{m}_j = \alpha(t) f(\Delta_{jc}) (\mathbf{m}_j - \mathbf{x}_i) \quad (2)$$

where α is a learning coefficient, and $f(\cdot)$ is a neighborhood function which controls the amount by which the weights of the neighbouring neurons are updated. The neighborhood function can take the form of a Gaussian function $f(\Delta_{jc}) = \exp\left(-\frac{\|\mathbf{l}_j - \mathbf{l}_c\|^2}{2\sigma^2}\right)$ where σ is the spread, and \mathbf{l}_c and \mathbf{l}_j is the location of the winning neuron and the location of the j -th neuron respectively. Other neighborhood functions are possible. Equations (1) and (2) are computed for every input vector in the training set, and for a set number of iterations.

The SOM for Data Structures (SOM-SD) extends the SOM in its ability to encode directed tree structured graphs [1]. This is accomplished by processing individual nodes of a graph one at a time rather than by processing a graph as a whole. The network response to a given node v is a mapping of v on the display space. This mapping is called the *state* of v and contains the coordinates of the winning neuron. An input vector representation is created for every node in a graph G by concatenating a numeric data label \mathbf{l}_v which may be attached to a node v with the *state* of each of the node's immediate offsprings such that $\mathbf{x}_v = [\mathbf{l}_v \mathbf{y}_{\text{ch}[v]}]$, where $\text{ch}[v]$ denotes the children of node v , and $\mathbf{y}_{\text{ch}[v]}$ denotes the states or mappings of the children of v . The dimension of \mathbf{x} is made constant in size by assuming a maximum dimension for \mathbf{l} together with a maximum out-degree of a node. For nodes with less dimensions than the assumed, padding with a suitable value is applied. Since the initialization of \mathbf{x} depends on the availability of all the children states, this dictates the processing of nodes in an inverse topological order (i.e. from the leaf nodes towards the root nodes), and hence, this causes information to flow in a strictly causal manner (from the leaf nodes to the root nodes).

A SOM-SD is trained in a similar fashion to the standard SOM with the difference that the vector elements \mathbf{l} and \mathbf{y}_{ch} need to be weighted so as to control the influence of these components to a similarity measure. Equation (1) is altered to become:

$$c = \arg \min_j (\|\mathbf{x}_v - \mathbf{m}_j\| \mathbf{\Lambda}) \quad (3)$$

where \mathbf{x}_v is the input vector for vertex v , \mathbf{m}_i the i -th codebook, and $\mathbf{\Lambda}$ is a $m \times m$ dimensional diagonal matrix with its diagonal elements $\lambda_{1,1} \cdots \lambda_{p,p}$ set

to μ_1 , and $\lambda_{p+1,p+1} \cdots \lambda_{m,m}$ set to μ_2 . The constants μ_1 and μ_2 control the influence of \mathbf{l}_v and $\mathbf{y}_{\text{ch}[v]}$ to the Euclidean distance in (3).

The rest of the training algorithm remains the same as that of the standard SOM. The effect of this extension is that the SOM-SD will map a given set of graphs, and all sub-graphs onto the same map. The SOM-SD includes the standard SOM and the SOM for data sequences as special cases.

With contextual SOM for graphs (CSOM-SD), the network input is formed by additionally concatenating the state of parent nodes and children nodes to an input vector such that $\mathbf{x}_v = [\mathbf{l} \ \mathbf{y}_{\text{ch}[v]} \ \mathbf{y}_{\text{pa}[v]}]$, where $\mathbf{y}_{\text{pa}[v]}$ are the states of the parent nodes and $\mathbf{y}_{\text{ch}[v]}$ are the states of the children nodes. The problem with this definition is that a circular functional dependency is introduced between the connected vertices v and $\text{pa}[v]$, and so, neither the state for node v nor the state of its parents $\text{pa}[v]$ can be computed. One possibility to compute these states could be to find a joint stable fix point to the equations involving all the vertices of a structure. This could be performed by initializing all the states with random values and then updating these initial states using the above mentioned equations, till a fixed point is reached. Unfortunately, there is no guarantee that such a fixed point would be reached. Moreover, even if sufficient conditions can be given over the initial weights of the map to guarantee stability, i.e. the existence of the fixed point, there is no guarantee that training will remain valid on such sufficient conditions over the weights.

A (partial) solution to this dilemma has been proposed in [2]. The approach is based on an K -step approximation of the dynamics described above: Let

$$\mathbf{y}^t = h(\mathbf{x}_v^{t-1}), t = 1, \dots, K \quad (4)$$

where $h(\cdot)$ computes the state of node v by mapping the input \mathbf{x}_v^{t-1} , and $\mathbf{x}_v^{t-1} = [\mathbf{l}_v \ \mathbf{y}_{\text{ch}[v]}^{t-1} \ \mathbf{y}_{\text{pa}[v]}^{t-1}]$. The algorithm is initialized by setting $\mathbf{y}_{\text{ch}[v]}^0 = \mathbf{y}_{\text{pa}[v]}^0 = k$, where $k = [-1, -1]$, an impossible winning coordinate. In other words, the approach iteratively re-computes the states of every node in a graph K -times. Then, the network input can be formed by setting $\mathbf{x}_v = [\mathbf{l} \ \mathbf{y}_{\text{ch}[v]}^K \ \mathbf{y}_{\text{pa}[v]}^K]$. A suitable value for K could be, for instance, the maximum length of any path between any two nodes in the graph. Although such a choice does not guarantee the full processing of contextual information due to possible latency in the transfer of contextual information from one vertex of the structure to its neighbors vertices, this value for K seems to be a good tradeoff between contextual processing and computational cost.

Training is performed similar to the training of SOM-SD with the difference that $\mathbf{\Lambda}$ is now a $n \times n$ matrix, $n = \dim(\mathbf{x})$ with $\lambda_{m+1,m+1} \cdots \lambda_{n,n}$ set to the constant μ_3 . All other elements in $\mathbf{\Lambda}$ are the same as defined before.

3 Experiments

The corpus (`m-db-s-0`) considered consists of 9,640 XML formatted documents which were made available as part of the INEX Initiative (INitiative for the Evaluation of XML Retrieval). Each of the XML formatted documents describes

an individual movie (e.g. the movie title, list of actors, list of reviewers, etc.). It was built using the IMDB database. Each document is labelled by one thematic category which represents the genre of the movie in the original collection and one structure category. There are 11 thematic categories and 11 possible structure categories which correspond to transformations of the original data structure. Note that the target labels are used solely for testing purposes, and hence, are ignored during the training process.

A tree structure was extracted for each of the documents in the dataset by following the general XML structure within the documents. The resulting dataset featured 9,640 tree structured graphs, one for each XML document in the dataset. The maximum depth of any graph is 3, the maximum outdegree is 6,418, and the total number of nodes in the dataset is 684,191. Hence, the dataset consists of shallow tree structures which can be very wide. A three-dimensional data label is attached to every node in the dataset indicating the XML-tag it represents (more on this below). There were a total of 197 different tags in the dataset.

While for the SOM-SD and CSOM-SD there is no specific need to pre-process this set of graphs, we decided to apply a pre-processing step in order to reduce the dimensionality of the dataset. This allows for a reasonable turn around time for the experiments. Dimension reduction was achieved by consolidating XML tags as follows: Repeated sequences of tags within the same level of a structure are consolidated. For example, the structure:

<pre><BB> <a> <a> <a> </BB></pre>	is consolidated to	<pre><BB> <a> </BB></pre>
---	--------------------	---

A justification for taking this step is inspired by operations in regular expressions. For example, the expression $(ab)^n$ can be simulated by repeatedly presenting ab n -times. Hence, it suffices to process the consolidated structure n times. There were many trees which exhibited such repeated sequences of tags. The consequence of this pre-processing step is that the maximum outdegree is reduced to just 32.

A further dimension reduction is achieved by collapsing sequences into a single node. For example, the sequential structure $\langle A \rangle \langle b \rangle \langle c \rangle \langle /c \rangle \langle /b \rangle \langle /A \rangle$ can be collapsed to $\langle A \rangle \langle b \&c \rangle \langle /b \&c \rangle \langle /A \rangle$, and further to $\langle A \&b \&c \rangle$. Since the deepest graph is of depth 3, this implies that the longest sequence that can be collapsed is of length 3. This pre-processing step reduces the total number of nodes in the dataset to 247,140.

A unique identifier (ID) is associated with each of the 197 XML tags. In order to account for nodes which represent collapsed sequences, we attach a three dimensional data label to each node. The first element of the data label gives the ID of the XML tag it represents, the second element of the data label is the

Label	Frequency
1	598
10	386
11	448
2	486
3	701
4	172
5	435
6	231
7	261
8	769
9	333
Total	4820

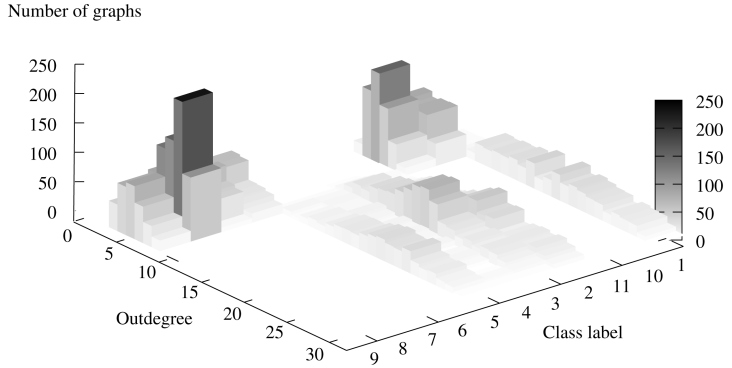


Fig. 1. Properties of the training set: The table (left) shows the number of graphs in each of the 11 classes. The plot (right) shows the distribution of outdegrees in the dataset. Shown are the number of graphs (z-axis) which have a given outdegree (y-axis) and belong to a given class (x-axis).

ID number of the first tag of a collapsed sequence of nodes, and consequently, the third element is the ID of the tag of the leaf node of a collapsed sequence. For nodes which do not represent a collapsed structure, the second and third element in the data label will be set to zero.

The resulting dataset consists of 4,820 graphs containing a total of 124,360 nodes (training set), and 4,811 graphs containing a total of 122,780 nodes (test set). The training set was analysed for its statistical properties; and the results are shown in Figure 1. It is observed that the training set is unbalanced. For example, the table on the left of Figure 1 shows that there are only 172 samples of the pattern instance denoted by “4” but over 700 instances of patterns from the instance denoted by “3”. Also, the 3-D plot in Figure 1 shows that the distribution of outdegrees can vary greatly. For example, there is only one instance in the pattern class denoted by “8” which has an outdegree of 10 while there are over 270 instances for the same pattern class with outdegree 5. There are also a number of pattern classes which are similar in features such as class “10” and class “11” which are of similar size and are of similar structure.

There are 2,872 unique sub-structures in the training set. This is an important statistical figure since it gives an indication to how much more information is provided to a SOM-SD when compared to the flat vectors used for the SOM. And hence, the larger the number of unique sub-structures in the training set, the greater the potential diversification in the mapping of the data will be. Similarly, there are 96,107 unique nodes in different contextual configurations in the training set. This shows that the CSOM-SD is provided with a greater set of diverse features in the training set, and hence, may be capable of diversifying in the mapping of the data even further. Thus, this dataset provides a challenging learning problem on which various SOM models will be tested.

All SOMs illustrated in this section used a hexagonal topology, and a Gaussian neighborhood function. For the SOM-SD and the CSOM-SD, when generating the input vectors \mathbf{x}_i for nodes with less than the maximum outdegree, padding was performed using the impossible coordinate $[-1, -1]$.

The standard SOM is trained on 4,820 data *vectors*, each one represents an XML document. The i -th element in the data *vectors* represents the frequency of the i -th XML tag within a document. Thus, the input vectors for the SOM are 197 dimensional containing all the information about the XML tags in a document but do not contain any information about the topological structure between the XML tags.

Thus, the SOM is trained on relatively few high-dimensional data vectors while the SOM-SD or the CSOM-SD is being trained on a large number of nodes which are represented by a relatively small size vectors. For the SOM we chose $64 \times 48 = 3,072$ as the size of the network. The total number of network parameters for the SOM is $3,072 \times 197 = 605,184$. Since the codebook dimensions for the SOM-SD is $3 + 32 \times 2 = 67$, this implies that a SOM-SD needs to feature at least 9,033 codebooks to allow a fair comparison. Accordingly, the CSOM-SD should feature at least 8,771 neurons. However, since the SOM-SD (and to an even greater extent the CSOM-SD) is to encode a larger feature set which includes causal (contextual) information about the data, this implies that the SOM-SD (CSOM-SD) will potentially diversify the mapping of the data to a greater extent than a SOM would do. Hence, this would justify the choice of even larger networks for the SOM-SD and CSOM-SD respectively for the comparisons. However, we chose to use these network sizes as these suffice to illustrate the principal properties of the models.

It is evident that a simple quantization error is an insufficient indicator of the performance of a SOM-SD or a CSOM-SD since such an approach neglects to take into account the fact that structural information is being mapped. In fact, there are a number of criteria with which the performance of a SOM-SD or a CSOM-SD can be measured.

Retrieval capability (R): This reflects the accuracy of retrieved data from the various SOM models. This can be computed quite simply if for each XML document d_j a target class $y_j \in \{t_1, \dots, t_q\}$ is given. Since each XML document is represented by a tree, in the following, we will focus our attention just on the root of the tree. Thus, with r_j we will refer to the input vector for SOM, SOM-SD or CSOM-SD representing the root of the XML document d_j . The R index is computed as follows: the mapping of every node in the dataset is computed; then for every neuron i the set $win(i)$ of root nodes for which it was a winner is computed. Let $win_t(i) = \{r_j | r_j \in win(i) \text{ and } y_j = t\}$, the value $R_i = \max_t \frac{|win_t(i)|}{|win(i)|}$ is computed for neurons with $|win(i)| > 0$ and the index R computed as $R = \frac{1}{W} \sum_{i, |win(i)| > 0} R_i$, where $W = \sum_{i, |win(i)| > 0} 1$ is the total number of neurons which were activated at least once by a root node.

Classification performance (C): This can be computed as follows:

$$C_j = \begin{cases} 1 & \text{if } y_j = t_r^*, \quad t_r^* = \arg \max_t |win_t(r)| \\ 0 & \text{else} \end{cases},$$

where r is the index of the best matching codebook for document d_j (typically measured at the root node). Then,

$$C = \frac{1}{N} \sum_{j=0}^N C_j,$$

where N is the number of documents (graphs) in the test set. Values of C and R can range within $(0 : 1]$ where values closer to 1 indicate a better performance.

Clustering performance (P): A more sophisticated approach is needed to compute the ability of a SOM-SD or a CSOM-SD to suitably group data on the map. In this paper the following approach is proposed:

1. Compute the quantities R_i as defined above, and let $t_i^* = \arg \max_t |win_t(i)|$.
2. For any activated neuron compute the quantity:

$$P_i = \frac{\sum_{j=1}^{|\mathcal{N}_i|} \frac{|win_{t_i^*}(j)|}{|win(j)|} + \frac{|win_t(i)|}{|win(i)|}}{|\mathcal{N}_i| + 1} = \frac{\sum_{j=1}^{|\mathcal{N}_i|} \frac{|win_{t_i^*}(j)|}{|win(j)|} + R_i}{|\mathcal{N}_i| + 1}$$

where $\mathcal{N}_i = \{v | v \in ne[i], win(v) \neq \emptyset\}$.

3. The overall neural network performance is then given by:

$$P = \frac{\sum_i P_i}{W}.$$

A performance value close to 1 indicates a perfect grouping, while a value closer to 0 indicates a poor clustering result. Thus, this measure indicates the level of disorder inside a SOM-SD or CSOM-SD.

Structural mapping precision (e and E): These indices measure how well structural (e) and contextual structural (E) information are encoded in the map. A suitable method for computing the structural mapping precision was suggested in [2]. In this case, just the skeleton of the trees is considered, i.e. the information attached to vertices is disregarded, and only the topology of the trees is considered. Notice that these measures do not consider the information about the class to which an XML document (i.e., a tree) belongs. For this reason, all the neurons of a map are now considered, since we are also interested in neurons which are winners for sub-structures. These two measures e and E are respectively computed as follows

$$e = \frac{1}{N} \sum_{i=1, n_i \neq 0}^N \frac{m_i}{n_i} \quad \text{and} \quad E = \frac{1}{N} \sum_{i=1, n_i \neq 0}^N \frac{M_i}{n_i}$$

where n_i is the number of sub-structures mapped at location i , m_i is the greatest number of sub-structures which are identical and are mapped at location i . Similarly, M_i is the greatest number of identical complete trees which are associated with the sub-structure mapped at location i . N is the total number of neurons activated by at least one sub-structure during the mapping process. Hence, e is an indicator of the quality of the mapping of sub-structures, and E indicates the quality of the contextual mapping process. Values of e and E close to 1 indicate a very good mapping (indeed a *perfect* mapping if the value is 1), and values closer to 0 indicate a poor mapping.

Compression ratio: This is the ratio between the total number of root nodes in the training/test set, and the number of neurons actually activated by root nodes in the training/test set. The higher the compression, the fewer the number of neurons are involved in the mapping process. Extremely high or extremely low compression ratios can indicate a poor performance. The compression ratio can vary between 0 and N , where N is the number of root nodes in the training/test set.

A number of SOMs, SOM-SDs, and CSOM-SDs were trained by varying the training parameters, and initial network conditions. We used the classification measure C as a general benchmark on which to optimize the performance of the various models. A total of 56 experiments were executed for each of the SOM models, and every experiment was repeated 10 times using a different random initialization of the map as a starting point. The experiments varied the following training parameters: number of training iterations i , initial neighborhood radius $r(0)$, initial learning rate $\alpha(0)$, and the weight values μ (in this order). The set of training parameters which maximised the classification performance of the three models is shown below.

	size	# iterations	$\alpha(0)$	$r(0)$	μ_1	μ_2	μ_3
SOM	64×48	32	1.0	4	1.0	–	–
SOM-SD	110×81	62	1.0	38	0.11	0.89	–
CSOM-SD	110×81	12	0.7	15	0.11	0.88	0.01 ²

It is observed that the SOM-SD required more training iterations and a larger initial neighborhood radius to achieve optimum classification performance (on the training set). It was also observed that the classification performance of the CSOM-SD improved with smaller values for μ_3 reaching an optimum for $\mu_3 = 0.0$. However, setting μ_3 to zero would reduce the CSOM-SD to a SOM-SD, and hence, would be an unsuitable choice for the comparisons. In this case we have set μ_3 to a small value.

Table 1. Best results obtained during the experimentation with maps of size 64×48 (SOM), and for maps of size 110×81 (SOM-SD and CSOM-SD)

	train set						test set					
	C	R	P	e	E	Z	C	R	P	e	E	Z
SOM	90.5%	0.90	0.73	1.0	1.0	2.45	76.5%	0.92	0.73	1.0	1.0	2.45
SOM-SD	92.5%	0.92	0.78	0.77	0.50	5.13	87.3%	0.93	0.79	0.76	0.50	4.9
CSOM-SD	83.9%	0.87	0.73	0.91	0.30	8.53	78.6%	0.88	0.71	0.90	0.37	8.54

The performances of the three SOM models are illustrated in Table 1 with the above mentioned performance measures. From Table 1 it can be observed that a standard SOM is capable of classifying over 90% of patterns in the training

² Smallest non-zero value tried. Setting $\mu_3 = 0.0$ resulted in a better classification performance but would reduce the CSOM-SD to a SOM-SD.

set correctly despite of no information about the underlying causal or contextual configuration of XML tags is provided to the training algorithm. However, it was found that the SOM generalizes poorly. In comparison, the SOM-SD improved the classification rate by a noticeable amount, and was able to generalize over unseen data very well. As is observed from the compression ratio Z , the performance increase of the SOM-SD comes despite a doubling of the compression ratio. This is a clear evidence that causal information about the order of XML tags allows (a) to diversify the mapping of nodes to a considerably larger extend, and (b) the diversification in the mappings can result in an overall improvement of the classification or clustering performances. In contrast, the inclusion of contextual information did not help to improve on the classification performance as it is observed from the results obtained from the CSOM-SD. It is found that contextual information helped to diversify the mapping of nodes by almost double when compared to the SOM-SD. This is indicated by the larger compression ratio. Thus, it is evident that a correct classification of the graphs in the dataset is independent of the contextual information about the XML tags within the original documents. When paired with the greater data compression which is the result of a greater diversification in the mapping of nodes, this produced a relative overall reduction in classification performance for the CSOM-SD, and explains the observation that the performance optimum of the CSOM-SD is at $\mu_3 = 0$.

In addition, it is observed that a CSOM-SD performs worse on the performance measure E than a SOM-SD. This is a result which arose out of the fact that the experiments were to optimize the classification performance C . It was found that a CSOM-SD improves on C when using $\mu_3 \rightarrow 0$. However, setting $\mu_3 = 0$ would reduce the CSOM-SD to a SOM-SD and would have denied us from making a comparison between the models. Instead, we chose a small value for μ_3 so as to allow such comparisons, and still produce reasonable classification performances. Using a very small μ_3 reduces the impact of contextual information to the training algorithm. When paired with the increased compression ratio in the mapping of root nodes, this resulted in a relative decrease in the performance on E . Note that the standard SOM performed at $e = E = 1$. This is due to the fact that a SOM handles the simplest type of data structures (viz. single nodes). These render all structures in the dataset identical, resulting in the observed performance values.

A closer look at the mapping of (training) data is made in the standard SOM Figure 2(a). The hexagons in Figure 2(a) refer to the neurons on the map. The brightness of the grid intersection represents the number of training data which are assigned to the grid point due to their closeness in the original input space. Thus by examining the brightness in the grid, it is possible to gain an appreciation of the way the given training dataset can be grouped together, according to their closeness in the original input space. Every neuron is also filled in with a pattern indicating the class that most frequently activated the neuron. There are 11 different fill in patterns for the 11 possible classes. Neurons which are not filled in are not activated by any vector in the training set. It can be observed that a number of well distinct clusters have formed on the map,

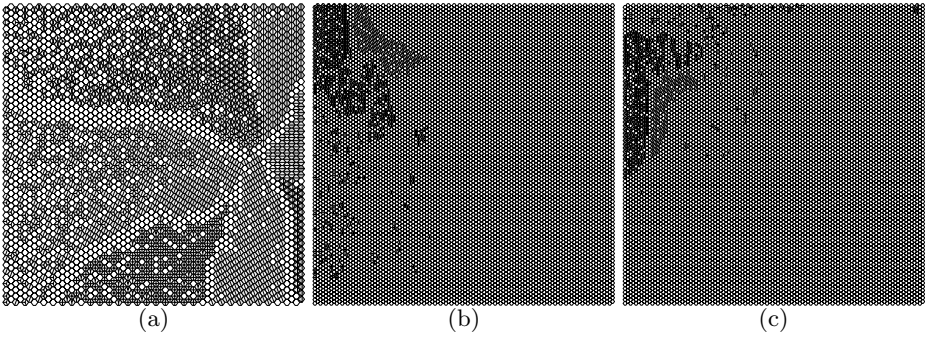


Fig. 2. The mapping of the training vectors on a standard SOM is shown in (a). The mapping of root nodes (training vectors) on a SOM-SD is shown in (b). The mapping of root nodes (training vectors) on a CSOM-SD is shown in (c).

most of which correspond nicely with the target label that is associated with the training data. Most clusters are separated from each other by an area of neurons which were not activated. This may indicate a good result since the presence of such border regions should allow for a good generalization performance; a statement which could not be confirmed when evaluating the test set.

In comparison, the mapping of root nodes in the training set on a trained SOM-SD is shown in Figure 2(b). Neurons which are not filled in are either not activated by a root node, or are activated by a node other than the root node. It can be observed in Figure 2(b) that large sections of the map are not activated by any root node. This is due to the fact that root nodes are a minority in the dataset. Only 4,824 nodes out of the total 124,468 nodes in the training set are root nodes. Hence, only a relatively small portion of the map is activated by root nodes. It is also observed that graphs belonging to different classes form clear clusters some of which are very small in size. This observation confirms the experimental findings which show that the SOM-SD will be able to generalize well.

Figure 2(c) gives the mapping of the root nodes as produced by the CSOM-SD. Again, it is found that the largest portion of the map is filled in by neurons which are either not activated or are activated by nodes other than the labelled root nodes. Clear clusters are formed which are somewhat smaller in size when compared to those formed in the SOM-SD case. This illustrates quite nicely that the CSOM-SD is compressing the “root” data considerably more strongly than the SOM-SD since contextual information is also encoded which requires additional room in the map. Nevertheless, the observation confirms that the CSOM-SD will also be able to generalize well even though some of the performance indices may be worse than when compared to a SOM-SD of the same size. This can be expected since the CSOM-SD compresses the “root” data more strongly.

4 Conclusions

The clustering of graphs and sub-graphs can be a hard problem. This paper demonstrated that the clustering task of general types of graphs can be

performed in linear time by using a neural network approach based on Self-Organizing Maps. In addition, it was shown that SOM-SD based networks can produce good performances even if the map is considerably smaller than the size of the training set. Using larger maps will generally improve the performance further though this was not illustrated in this paper.

Specifically, it was shown that the given learning problem depends on the availability of causal information about the XML tags within the original document in order to produce a good grouping or classification of the data. The incorporation of contextual information did not help to improve on the results further.

The training set used in this paper featured a wide variety of tree structured graphs. We found that most graphs are relatively small in size, only few graphs were either very wide or featured many nodes. This creates imbalances in features represented in a training set which is known to negatively impact the performance of a neural network. Similarly it is true when considering the way we generated data labels for the nodes. An improvement of these aspects (i.e. balancing the features in the training set, using unique labels which are equiv-distant to each other) should help to improve the network performances. An investigation into the effects of these aspects is left as a future task.

Furthermore, it was shown that the (C)SOM-SD models map graph structures onto a finite regular grid in a topology preserving manner. This implies that similar structures are mapped onto nearby areas. As a consequence, these SOM models would be suitable for inexact graph matching tasks. Such applications are considered as a future task.

References

1. M. Hagenbuchner, A. Sperduti, and A. Tsoi. A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks*, 14(3):491–505, May 2003.
2. M. Hagenbuchner, A. Sperduti, and A. Tsoi. Contextual processing of graphs using self-organizing maps. In *European symposium on Artificial Neural Networks*, Poster track, Bruges, Belgium, 27 - 29 April 2005.
3. T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995.
4. C. Leslie, E. Eskin, and W. Noble. Spectrum kernel: A string kernel for svm protein classification. *Proceedings of the Pacific Symposium on Biocomputing*, pages 474–485, 2002.
5. D. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 1st edition, 2003.
6. S. A. Scholkopf, B. *Learning with Kernels*. MIT Press, Cambridge, MA, 1st edition, 2002.

Finding Time Series Discords Based on Haar Transform

Ada Wai-chee Fu¹, Oscar Tat-Wing Leung¹, Eamonn Keogh², and Jessica Lin³

¹ Department of Computer Science and Engineering,
The Chinese University of Hong Kong
{adafu, twleung}@cse.cuhk.edu.hk

² Department of Computer Science and Engineering,
University of California, River, CA 92521
eamonn@cs.ucr.edu

³ Department of Information and Software Engineering,
George Mason University
jessica@ise.gmu.edu

Abstract. The problem of finding anomaly has received much attention recently. However, most of the anomaly detection algorithms depend on an explicit definition of anomaly, which may be impossible to elicit from a domain expert. Using discords as anomaly detectors is useful since less parameter setting is required. Keogh et al proposed an efficient method for solving this problem. However, their algorithm requires users to choose the word size for the compression of subsequences. In this paper, we propose an algorithm which can dynamically determine the word size for compression. Our method is based on some properties of the Haar wavelet transformation. Our experiments show that this method is highly effective.

1 Introduction

In many applications, time series has been found to be a natural and useful form of data representation. Some of the important applications include financial data that is changing over time, electrocardiograms (ECG) and other medical records, weather changes, power consumption over time, etc. As large amounts of time series data are accumulated over time, it is of interest to uncover interesting patterns on top of the large data sets. Such data mining target is often the common features that frequently occur. However, to look for the unusual pattern is found to be useful in some cases. For example, an unusual pattern in a ECG can point to some disease, unusual pattern in weather records may help to locate some critical changes in the environments.

Algorithms for finding the most unusual time series subsequences are proposed by Keogh et al in [6]. Such a subsequence is also called a time series *discord*, which is essentially a subsequence that is the least similar to all other subsequences. Time series discords have many uses in data mining, including improving the quality of clustering [8,2], data cleaning and anomaly detection [9,1,3]. By a

comprehensive set of experiments, these previous works demonstrated the utility of discords on different domains such as medicine, surveillance and industry.

Many algorithms have been proposed for detecting anomaly in a time series database. However, most of them require many un-intuitive parameters. Time series discords, which were first suggested by Keogh et al, are particular attractive as anomaly detectors because they only require three parameters. The efficiency of the algorithm in [5] is based on an early pruning step and a reordering of the search order to speed up the search. Each time series is first compressed into lower dimensions by a piecewise linear transformation, so that the result is a shorter string (word) of alphabets, where each alphabet corresponds to a range of measured values that has been replaced by the mean value. Hence users are required to choose two parameters, the cardinality of the alphabet size a , and the word size w . For the parameter a , previous works reported that a value of either three or four is the best for any task on any dataset that have been tested.

However, for the parameter w , there is no single suitable value for any task on any dataset. It has been observed that relatively smooth and slowly changing datasets favor a smaller value of w ; otherwise a larger value for w is more suitable. Unfortunately, we still have questions on how to determine a time series is smooth or not and what is the meaning of a larger value of w .

In this paper we propose a word size free algorithm by first converting subsequences into Haar wavelets, then we use a breadth first search to approximate the perfect search order for outer loop and inner loop. In this way, it is possible to dynamically select a suitable word size.

2 Background

We first review some background material on time series discords, which is the main concern of our proposed algorithm. Then Haar transform will be introduced, which provides an efficiency way to estimate the discord in a given time series and it plays an important role in our algorithm.

2.1 Time Series Discords

In general, the best matches of a given subsequence (apart from itself) tend to be very close to the subsequence in question. Such matches are called trivial matches. When finding discords, we should exclude trivial matches; otherwise, we may fail to obtain true patterns. Therefore, we need to formally define a non-self match.

Definition 1. *Non-self Match: Given a time series T , containing a subsequence C of length n beginning at position p and a matching subsequence M beginning at q , we say that M is a non-self match to C if $|p - q| \geq n$*

We now can define time series discord by using the definition of non-self matches:

Definition 2. *Time Series Discord: Given a time series T , the subsequence D of length n beginning at position l is said to be the discord of T if D has the*

largest distance to its nearest non-self match. That is, all subsequence C of T , non-self match M_D of D , and non-self match M_C of C , minimum Euclidean Distance of D to $M_D >$ minimum Euclidean Distance of C to M_C .

The problem to find discords can obviously be solved by a brute force algorithm which considers all the possible subsequences and finds the distance to its nearest non-self match. The subsequence which has the greatest such value is the discord. However, the time complexity of this algorithm is $O(m^2)$, where m is the length of time series. Obviously, this algorithm is not suitable for large dataset.

Keogh et al introduced a heuristic discord discovery algorithm based on the brute force algorithm and some observations [5]. They found that actually we do not need to find the nearest non-self match for each possible candidate subsequence. According to the definition of time series discord, a candidate cannot be a discord, if we can find any subsequence that is closer to the current candidate than the current smallest nearest non-self match distance. This basic idea successfully prunes away a lot of unnecessary searches and reduces a lot of computational time.

2.2 Haar Transform

The Haar wavelet Transform is widely used in different applications such as computer graphics, image, signal processing and time series querying [7]. We propose to apply this technique to approximate the time series discord, as the resulting wavelet can represent the general shape of a time sequence. Haar transform can be seen as a series of averaging and differencing operations on a discrete time function. We compute the average and difference between every two adjacent values of $f(x)$. The procedure to find the Haar transform of a discrete function $f(x) = (9\ 7\ 3\ 5)$ is shown below.

Example

Resolution Averages Coefficients		
4	(9 7 3 5)	
2	(8 4)	(1 -1)
1	(6)	(2)

Resolution 4 is the full resolution of the discrete function $f(x)$. In resolution 2, (8 4) are obtained by taking average of (9 7) and (3 5) at resolution 4 respectively. (1 -1) are the differences of (9 7) and (3 5) divided by two respectively. This process is continued until a resolution of 1 is reached. The Haar transform $H(f(x)) = (c\ d_0^0\ d_0^1\ d_1^1) = (6\ 2\ 1\ -1)$ is obtained which is composed of the last average value 6 and the coefficients found on the right most column, 2, 1 and -1. It should be pointed out that c is the overall average value of the whole time sequence, which is equal to $(9 + 7 + 3 + 5)/4 = 6$. Different resolutions can be obtained by adding difference values back to or subtract difference from an average. For instance, (8 4) = (6+2 6-2) where 6 and 2 are the first and second coefficient respectively.

Haar transform can be realized by a series of matrix multiplications as illustrated in Equation (1). Envisioning the example input signal x as a column

vector with length $n = 4$, an intermediate transform vector \mathbf{w} as another column vector and Haar transform matrix \mathbf{H}

$$\begin{bmatrix} x'_0 \\ d_0^1 \\ x'_1 \\ d_1^1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (1)$$

The factor $1/2$ associated with the Haar transform matrix can be varied according to different *normalization* conditions. After the first multiplication of \mathbf{x} and \mathbf{H} , half of the Haar transform coefficients can be found which are d_0^1 and d_1^1 in \mathbf{w} interleaving with some intermediate coefficients x'_0 and x'_1 . Actually, d_0^1 and d_1^1 are the last two coefficients of the Haar transform. x'_0 and x'_1 are then extracted from \mathbf{w} and put into a new column vector $\mathbf{x}' = [x'_0 \ x'_1 \ 0 \ 0]^T$. \mathbf{x}' is treated as the new input vector for transformation. This process is done recursively until one element is left in \mathbf{x}' . In this particular case, c and d_0^0 can be found in the second iteration.

Hence can convert a time sequence into Haar wavelet by computing the average and difference values between the adjacent values in the time series recursively. It can be also varied according to different normalization conditions. The algorithm shown in Algorithm 1 is using the orthonormal condition. This transformation can preserve the Euclidean distance between two time series, and is therefore useful in our algorithm. If we only consider a prefix of the transformed sequences, the Euclidean distance between two such prefixes will be a lower bounding estimation for the actual Euclidean distance, the longer the prefix the more precise the estimation. Also note that the transformation can be computed quickly, requiring linear time in the size of the time series.

Algorithm 1. Haar Transform

```

1: // Initialization
2: w = size of input vector
3: output vector = all zero
4: dummy vector = all zero
5:
6: //start the conversion
7: while w > 1 do
8:   w = w/2
9:   for i = 0; i < w; i++ do
10:    dummy vector[i] =  $\frac{\text{input vector}[2*i] + \text{input vector}[2*i+1]}{\sqrt{2}}$ 
11:    dummy vector[i + w] =  $\frac{\text{input vector}[2*i] - \text{input vector}[2*i+1]}{\sqrt{2}}$ 
12:   end for
13:   for i = 0; i < (w * 2); i++ do
14:    output vector[i] = dummy vector[i]
15:   end for
16: end while

```

3 The Proposed Algorithm

We follow the framework of the algorithm in [6]. In this algorithm, we extract all the possible candidate subsequences in outer loop, then we find the distance to the nearest non-self match for each candidate subsequence in inner loop. The candidate subsequence with the largest distance to its nearest non-self match is the discord. We shall refer to this algorithm as the base Algorithm.

Algorithm 2. Base Algorithm

```

1: //Initialization
2: discord distance = 0
3: discord location = 0
4:
5: // Begin Outer Loop
6: for Each p in T ordered by heuristic Outer do
7:   nearest non-self match distance = infinity
8:   //Begin Inner Loop
9:   for Each q in T ordered by heuristic Inner do
10:    if  $|p - q| \geq n$  then
11:      Dist = Euclidean Distance ( $t_p, t_{p+1}, \dots, t_{p+n-1}, t_q, t_{q+1}, \dots, t_{q+n-1}$ )
12:      if Dist < discord distance then
13:        break;
14:      end if
15:      if Dist < nearest non-self match distance then
16:        nearest non-self match distance = Dist
17:      end if
18:    end if
19:  end for
20:  //End For Inner Loop
21:  if nearest non-self match distance > discord distance then
22:    discord distance = nearest non-self match distance
23:    discord location = p
24:  end if
25: end for
26: //End for Outer Loop
27:
28: //Return Solution
29: Return (discord distance, discord location)

```

In the above algorithm, we the heuristic search order for both outer and inner can affect the performance. In fact, if a sequential search order is used, this algorithm will become a brute force algorithm. Note that the discord D is the one that maximizes the minimum distance between D and any other non-self subsequence E

$$\max_D(\min_E(Dist(D, E)))$$

The Outer heuristic should order the true discord first since it will get the maximum value for discord distance which has the best chance to prune other

candidates at Line 12. Given the subsequence p , the Inner heuristic order should pick the subsequence q closest to p first, since it will give the smallest $Dist$ value, and which will have the best chance to break the loop at Line 12. In this section, we will discuss our suggested heuristic search order, so that the inner loop can often be broken in the first few iterations saving a lot of running time.

3.1 Discretization

We shall impose the heuristic Outer and Inner orders based on the Haar transformation of subsequences. We first transform all of the incoming sequences by the Haar wavelet transform. In order to reduce the complexity of time series comparison, we would further transform each of the transformed sequences into a sequence (word) of finite symbols. The alphabet mapping is decided by discretizing the value range for each Haar wavelet coefficient. We assume that for all i , the i^{th} coefficient of all Haar wavelets in the same database tends to be evenly distributed between its minimum and maximum value, so we can determine the "cutpoints" by partitioning this specify region into several equal segments. The cutpoints define the discretization of the $i - th$ coefficient.

Definition 3. Cutpoints: For the i^{th} coefficient, cutpoints are a sorted list of numbers $B_i = \beta_{i,1}, \beta_{i,2}, \dots, \beta_{i,m}$, where m is the number of symbols in the alphabet, and

$$\beta_{i,j} - \beta_{i,j+1} = \frac{\beta_{i,a} - \beta_{i,0}}{a} \quad (2)$$

$\beta_{i,0}$ and $\beta_{i,a}$ are defined as the smallest and the largest possible value of the i^{th} coefficient, respectively.

We then can make use of the cutpoints to map all Haar coefficients into different symbols. For example, if the i^{th} coefficient from a Haar wavelet is in between $\beta_{i,0}$ and $\beta_{i,1}$, it is mapped to the first symbol 'a'. If the i^{th} coefficient is between $\beta_{i,j-1}$ and $\beta_{i,j}$, it will be mapped to the j^{th} symbol, etc. In this way we form a word for each subsequence.

Definition 4. Word mapping: A word is a string of alphabet. A subsequence C of length n can be mapped to a word $\hat{C} = \hat{c}_1, \hat{c}_2, \dots, \hat{c}_n$. Suppose that C is transformed to a Haar wavelet $\bar{C} = \{\bar{c}_1, \bar{c}_2, \dots, \bar{c}_n\}$. Let α_j denote the j^{th} element of the alphabet, e.g., $\alpha_1 = a$ and $\alpha_2 = b, \dots$. Let $B_i = \beta_{i,1}, \dots, \beta_{i,m}$ be the Cutpoints for the i -th coefficient of the Haar transform. Then the mapping from to a word \hat{C} is obtained as follows:

$$\hat{c}_i = \alpha_j \iff \beta_{i,j-1} \leq \bar{c}_i < \beta_{i,j} \quad (3)$$

3.2 Outer Loop Heuristic

First, we transform all the subsequences, which are extracted by sliding a window with length n across time series T , by means of the Haar transform. The transformed subsequences are transformed into words by using our proposed discretizing algorithm. Finally, all the words are placed in an **array** with a pointer referring back to the original sequences. Figure 1 illustrates this idea.

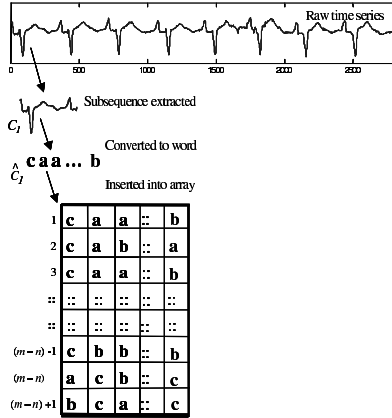


Fig. 1. An array of words for building an augmented trie

Next, we make use of the array to build an augmented **trie** by an iterative method. At first, there is only a root node which contains a linked list index of all words in the trie. In each iteration all the leaf nodes are split. In order to increase the tree height from h to $h+1$, where h is the tree height before splitting, the $(h + 1)^{th}$ symbol of each word under the splitting node is considered. If we consider all the symbols in the word, then the word length is equal to the subsequence length. In previous work [6] a shorter word length is used by using a piecewise linear mechanism to compress the subsequence, which means that user need to determine the word length beforehand. Here we make use of the property of Haar wavelets to dynamically adjust the effective word length according to the data characteristics. The word length is determined by the following heuristic:

Word length heuristic: Repeating the above splitting process in a breadth first manner in the construction of the trie until (i) there is only one word in any current leaf node or (ii) the n^{th} symbol has been considered.

The Haar coefficient can help us to view a subsequence in different resolutions, so the first symbol of each word gives us the lowest resolution for each subsequence. In our algorithm, more symbols are to be considered when the trie grow taller, which means that higher resolution is needed for discovering the discord. The reason why we choose to stop at the height where some leaf node contains only one word (or subsequence) is that the single word is much more likely to be the discord compared to any other word which appears with other words in the same node, since such words in the same node are similar at the resolution at that level. Hence the height at that point implies a good choice for the length of the word that can be used in the algorithm.

We found that the performance of this breadth first search on the trie approach is pretty good, since it can efficiently group all the similar subsequences under the same tree node and the distance between any two subsequences under the same node are very small.

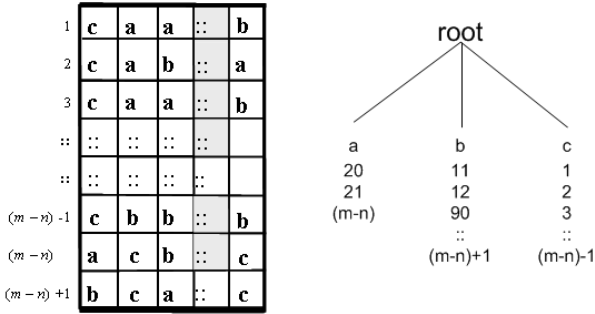


Fig. 2. 1st symbol is considered for splitting the root node. All leaf nodes will be split, since no leaf node contains only 1 word.

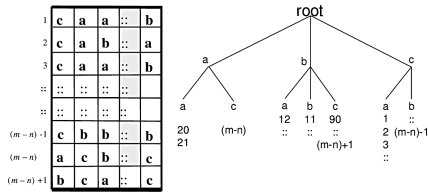


Fig. 3. 2nd symbol is considered. No tree node is split in next iteration, since there is only 1 word mapped to 'ac'.

Heuristic: *the leaf nodes are visited in ascending order according to the word count.*

We search all the subsequences in the nodes with the smallest count first, and we search in random order for the rest of the subsequences. The intuition behind our Outer heuristic is the following. When we are building an augmented trie, we are recursively splitting all the leaf nodes until there is only one subsequence in a leaf node. A trie node with only one subsequence is more likely to be a discord since there are no similar nodes that are grouped with it in the same node. This will increase the chance that we get the true discord as the subsequence used in the outer loop, which can then prune away other subsequences quickly.

More or less, the trie height can reflect the smoothness of the datasets. For smooth dataset, the trie height is usually small, as we can locate the discord at low resolution. On the other hand, the tire height is usually large for a more complex data set.

From this observation, it is obvious that the first subsequence that map to a unique word is very likely to be an unusual pattern. On the contrary, the rest of the subsequences are less likely to be the discord. As there should be at least two subsequences map to same tree node, the distance to their nearest non-self match must be very small.

3.3 Inner Loop Heuristic

When the i^{th} subsequence P is considered in the outer loop, we look up words in the i^{th} level of the trie.

Heuristic: *We find a node which gives us the longest matching path to p in the trie, all the subsequences in this node are searched first. After exhausting this set of subsequences, the unsearched subsequences are visited in a random order.*

The intuition behind our Inner heuristic is the following. In order to break the inner loop, we need to find a subsequence that has a distance to the i^{th} word in the outer loop less than the `best_so_far` discord distance, hence the smallest distance to p will be the best to be used. As subsequences in a node with a path close to p are very likely to be similar, by visiting them first, the chance for terminating the search is increased.

4 Empirical Evaluation

We first show the utility of time series discords, and then we show that our algorithm is very efficient for finding discords. The test datasets, which represent the time series from different domains, were obtained from "The UCR Time Series Data Mining Archive" [4].

4.1 Anomaly Detection

Anomaly Detection in a time series database has received much attention [9,1,3]. However, most anomaly detection algorithms require many parameters. In contrast our algorithm only requires two simple parameters, one is the length of the discord, another is the alphabet size, and for the alphabet size, it is known that either 3 or 4 is best for different tasks on many datasets from previous studies.

To show the utility of discords for anomaly detection, we investigated electrocardiograms (ECGs) which are time series of the electrical potential between two points on the surface of the body caused by a beating heart. In the experiment, we set the length of the discord to 256, which is approximately one full heartbeat and set the alphabet size to be three.

Our algorithm has successfully uncovered the discord in figure 4. In this example, it may seem that we can discover the anomaly by eye. However, we typically have a massive amount of ECGs, and it is impossible to examine all of them manually.

4.2 The Efficiency of Our Algorithm

Next we study the efficiency of the algorithm. From 4 datasets, 10 data sequences were picked. For each data sequence, prefixes of lengths 512, 1024, 2048, 4096 and 8192 were truncated, forming 4 derived datasets of varying dimensions. In figure 5, we compared the base Algorithm with our proposed algorithm in terms of the number of times the Euclidean distance function is called. In this experiment, we set the length of the discord to 128 and found the discord on all

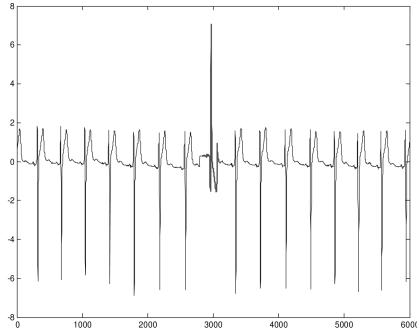


Fig. 4. A time series discord (marked in bold line) was found at position 2830

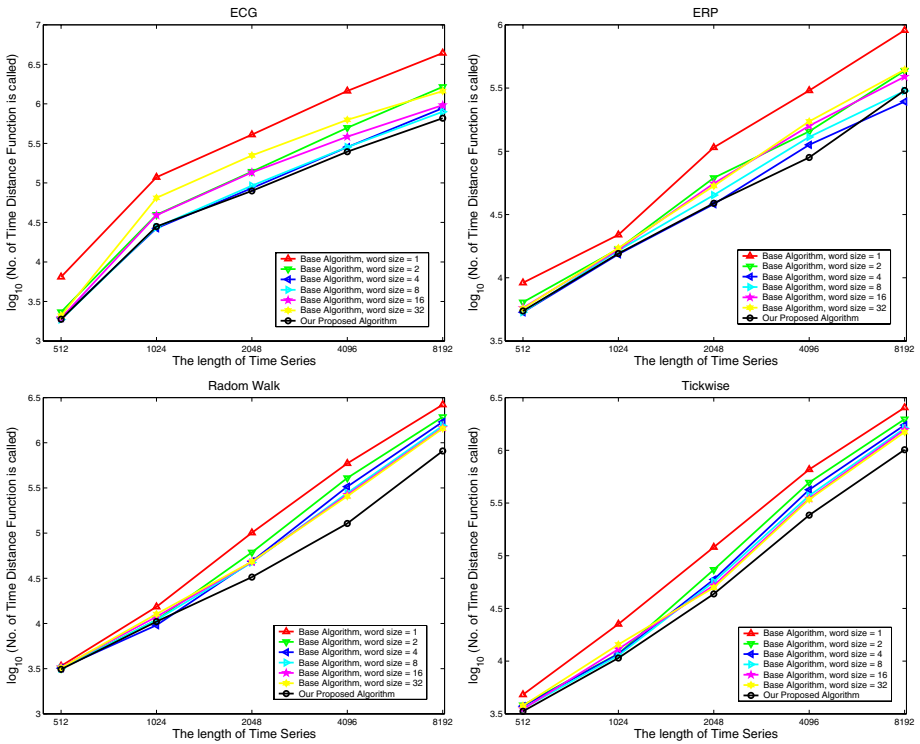


Fig. 5. Number of times distance function is called by the base Algorithm and Our Proposed Algorithm

the created subsequences. Each of the experiments was repeated 10 times and the average value was taken.

In this experiment, we did not measure the CPU time directly in order to ensure that there was no implementation bias. In fact, it has been discovered that the distance function accounts for more than 99% of the running time. By

measuring the number of distance computation, we have a relatively fair measure for the running time.

From the experimental results, we found that there was no special value for word size which was suitable for any task on any dataset. The results suggested that relatively smooth and slowly changing datasets favor a smaller value of word size, whereas more complex time series favor a larger value of word size. However, it is difficult to determine whether a dataset is smooth or otherwise. In most cases our proposed algorithm gave a better performance without the consideration of the word size comparing with the base Algorithm.

5 Conclusion and Future Work

We introduce a novel algorithm to efficiently find discords. Our algorithm only requires one intuitive parameter (the length of the subsequence). In this work, we focused on finding the most unusual time series subsequence. We plan to extend our algorithm to K time series discord which refers to finding K discords with the largest distance to its nearest non-self match.

Acknowledgements. This research was supported by the RGC Research Direct Grant 03/04, and the RGC Earmarked Research Grant of HKSAR CUHK 4120/05E.

References

1. D. Dasgupta and S. Forrest. Novelty detection in time series data using ideas from immunology, 1996.
2. E. Keogh. Exact indexing of dynamic time warping, 2002.
3. E. Keogh, S. Lonardi, and B. Chiu. Finding surprising patterns in a time series database in linear time and space. In *Proceedings of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, Edmonton, Alberta, Canada, July 2002.
4. E. Keogh and T. Folias. The ucr time series data mining archive.
5. E. J. Keogh, S. Lonardi, and C. A. Ratanamahatana. Towards parameter-free data mining. In *KDD*, pages 206–215, 2004.
6. J. Lin, E. J. Keogh, A. W.-C. Fu, and H. V. Herle. Approximations to magic: Finding unusual medical time series. In *CBMS*, pages 329–334, 2005.
7. K. pong Chan and A. W.-C. Fu. Efficient time series matching by wavelets. In *ICDE*, pages 126–133, 1999.
8. C. A. Ratanamahatana and E. J. Keogh. Making time-series classification more accurate using learned constraints. In *SDM*, 2004.
9. C. Shahabi, X. Tian, and W. Zhao. TSA-tree: A wavelet-based approach to improve the efficiency of multi-level surprise and trend queries on time-series data. In *Statistical and Scientific Database Management*, pages 55–68, 2000.

Learning with Local Drift Detection

João Gama^{1,2} and Gladys Castillo^{1,3}

¹ LIACC - University of Porto
Rua de Ceuta 118-6, 4050 Porto, Portugal

² Fac. Economics, University of Porto

³ University of Aveiro

jgama@liacc.up.pt, gladys@mat.ua.pt

Abstract. Most of the work in Machine Learning assume that examples are generated at random according to some stationary probability distribution. In this work we study the problem of learning when the distribution that generates the examples changes over time. We present a method for detection of changes in the probability distribution of examples. The idea behind the drift detection method is to monitor the on-line error-rate of a learning algorithm looking for significant deviations. The method can be used as a wrapper over any learning algorithm. In most problems, a change affects only some regions of the instance space, not the instance space as a whole. In decision models that fit different functions to regions of the instance space, like Decision Trees and Rule Learners, the method can be used to monitor the error in regions of the instance space, with advantages of fast model adaptation. In this work we present experiments using the method as a wrapper over a decision tree and a linear model, and in each internal-node of a decision tree. The experimental results obtained in controlled experiments using artificial data and a real-world problem show a good performance detecting drift and in adapting the decision model to the new concept.

1 Introduction

In many applications, learning algorithms acts in dynamic environments where the data flows continuously. If the process is not strictly stationary (as most of real world applications), the target concept could change over time. Nevertheless, most of the work in Machine Learning assume that training examples are generated at random according to some stationary probability distribution. In [1], the authors present several examples of real problems where change detection is relevant. These include user modelling, monitoring in bio-medicine and industrial processes, fault detection and diagnosis, safety of complex systems, etc.

The PAC learning framework assumes that examples are independent and randomly generated according to some probability distribution D . In this context, some model-class learning algorithms (like Decision Trees, Neural Networks, some variants of k-Nearest Neighbours, etc) could generate hypothesis that converge to the Bayes-error in the limit, that is, when the number of training examples increases to infinite. All that is required is that D must be stationary, the distribution must not change over time.

In this work we review the Machine Learning literature for learning in the presence of drift, we propose a taxonomy for learning in dynamic environments, and present a method to detect changes in the distribution of the training examples. The method is presented as a wrapper over any learning algorithm. We show that it can be used to detect drift in local regions of the instance space, by using inside inner-nodes of a Decision Tree.

The paper is organized as follows. The next section presents related work in detecting concept drifting and proposes a taxonomy for drift detection. In Section 3 we present the theoretical basis of the proposed method. In Section 4 we discuss evaluation methods in time changing environments, and evaluate the proposed method in one artificial dataset and one real-world dataset. Section 5 concludes the paper and presents future work.

2 Tracking Drifting Concepts

Concept drift means that the concept about which data is being collected may shift from time to time, each time after some minimum permanence. Changes occur over time. The evidence for changes in a concept are reflected in some way in the training examples. Old observations, that reflect the behaviour of nature in the past, become irrelevant to the current state of the phenomena under observation and the learning agent must forget that information.

Suppose a supervised learning problem, where the learning algorithm observe sequences of pairs (\mathbf{x}_i, y_i) where $y_i \in \{C_1, C_2, \dots, C_k\}$. At each time stamp t the learning algorithm outputs a class prediction \hat{y}_t for the given feature vector \mathbf{x}_t . Assuming that examples are independent and generated at random by a stationary distribution \mathcal{D} , some model class algorithms (e.g. Decision Trees, Neural Networks, etc) can approximate \mathcal{D} with arbitrary accuracy (bounded by the *Bayes error*) whenever the number of examples increases to infinite.

Suppose now the case where \mathcal{D} is not stationary. The data stream consists of sequences of examples $e_i = (\mathbf{x}_i, y_i)$. Suppose further that from time to time, the distribution that is generating the examples change. The data stream can be seen as sequences $\langle S_1, S_2, \dots, S_k, \dots \rangle$ where each element S_i is a set of examples generated by some stationary distribution \mathcal{D}_i . We designate as *context* each one of these sequences. In that case, and in the whole dataset, no learning algorithm can guarantee arbitrary precision. Nevertheless, if the number of observations within each sequence S_i is large enough, we could approximate a learning model to \mathcal{D}_i . The main problem is to detect change points whenever they occur. In real problems between two consecutive sequences S_i and S_{i+1} there could be a transition phase where some examples of both distributions appear mixed. An example generated by a distribution \mathcal{D}_{i+1} is noise for distribution \mathcal{D}_i . This is another difficulty faced by change detection algorithms. They must differentiate *noise* from *change*. The difference between noise and examples of another distribution is *persistence*: there should be a consistent set of examples of the new distribution. Algorithms for change detection must combine *robustness* to noise with *sensitivity* to concept change.

2.1 The Nature of Change

The nature of change is diverse and abundant. In this work, we identify two dimensions for analysis. The *causes* of change, and the *rate* of change. In a first dimension, the causes of change, we can distinguish between changes due to modifications in the context of learning, because of changes in hidden variables, from changes in the characteristic properties in the observed variables. Existing Machine Learning algorithms learn from observations described by a finite set of attributes. This is the *closed world* assumption. In real world problems, there can be important properties of the domain that are not observed. There could be *hidden* variables that influence the behaviour of nature [7]. Hidden variables may change over time. As a result, concepts learned at one time can become inaccurate. On the other hand, there could be changes in the characteristic properties of the nature.

The second dimension is related to the *rate of change*. The term *Concept Drift* is more associated to gradual changes in the target concept (for example the rate of changes in prices), while the term *Concept Shift* refers to abrupt changes. Usually, detection of abrupt changes are easier and require few examples for detection. Gradual changes are more difficult to detect. Detection algorithms must be resilient to noise. At least in the first phases of gradual change, the perturbations in data can be seen as noise by the detection algorithm. They often require more examples to distinguish change from noise.

Whenever a change in the underlying concept generating data occurs, the class-distribution changes, at least in some regions of the instance space. Nevertheless, it is possible to observe changes in the class-distribution without concept drift. This is usually referred as *virtual drift* [23].

2.2 Characterization of Drift Detection Methods

There are several methods in Machine Learning to deal with changing concepts: [13,12,11,23]. All of these methods assume that the most recent examples are the relevant ones. In general, approaches to cope with concept drift can be analysed into four dimensions: data management, detection methods, adaptation methods, and decision model management.

Data Management. The data management methods characterize the information about data stored in memory to maintain a decision model consistent with the actual state of the nature. We can distinguish:

1. *Full Memory.* Methods that store in memory sufficient statistics over all the examples. Examples include weighting the examples accordingly to their age. Weighted examples are based on the simple idea that the importance of an example should decrease with time. Thus, the oldest examples have less importance, see [16,17].
2. *Partial Memory.* Methods that store in memory only the most recent examples. Examples are stored in a *first-in first-out* data structure. Examples in the *fifo* define a time-window over the stream of examples. At each time

step, the learner induces a decision model using only the examples that are included in the window. The key difficulty is how to select the appropriate window size: a small window can assure a fast adaptability in phases with concept changes but in more stable phases it can affect the learner performance, while a large window would produce good and stable learning results in stable phases but can not react quickly to concept changes.

- (a) *Fixed Size* windows. These methods store in memory a fixed number of the most recent examples. Whenever a new example is available, it is stored in memory and the oldest one is discarded. This is the simplest method to deal with concept drift and can be used as a baseline for comparisons.
- (b) *Adaptive Size* windows. In this method, the set of examples in the window is variable. It is used in conjunction with a detection model. The most common strategy consists of decreasing the size of the window whenever the detection model signals drift and increasing otherwise.

Dynamic environments with non-stationary distributions require the forgetfulness of the observations not consistent with the actual behaviour of the nature. Drift detection algorithms must not only adapt the decision model to newer information but also forget old information. The memory model also indicates the *forgetting mechanism*. Weighting examples corresponds to a *gradual* forgetting. The relevance of old information is less and less important. Time windows corresponds to *abrupt* forgetting. The examples are deleted from memory. We can combine, of course, both forgetting mechanisms by weighting the examples in a time window, see [11].

Detection Methods. The Detection Model characterizes the techniques and mechanisms for drift detection. One advantage of the detection model is that they can provide meaningful description (indicating change-points or small time-windows where the change occurs) and quantification of the changes. They may follow two different approaches:

1. Monitoring the evolution of performance indicators. Some indicators (e.g. performance measures, properties of the data, etc.) are monitored over time (see [13] for a good overview of these indicators).
2. Monitoring distributions on two different time-windows.

Most of the work in drift detection follows the first approach. Relevant work in this approach is the FLORA family of algorithms developed by [23]. FLORA2 includes a window adjustment heuristic for a rule-based classifier. To detect concept changes, the accuracy and the coverage of the current learner are monitored over time and the window size is adapted accordingly. In the context of information filtering, the authors of [13] propose monitoring the values of three performance indicators: *accuracy*, *recall* and *precision* over time, and their posterior comparison to a confidence interval of standard sample errors for a moving average value (using the last M batches) of each particular indicator. In [12], Klinkenberg and Joachims present a theoretically well-founded method to

recognize and handle concept changes using properties of Support Vector Machines. The key idea is to select the window size so that the estimated generalization error on new examples is minimized. This approach uses unlabelled data to reduce the need for labelled data, it does not require complicated parametrization and it works effectively and efficiently in practice.

An example of the latter approach, in the context of learning from Data Streams, has been present by [10]. The author proposes algorithms (statistical tests based on Chernoff bound) that examine samples drawn from two probability distributions and decide whether these distributions are different. In the same line [5] system VFDTc has the ability to deal with concept drift, by continuously monitoring differences between two class-distribution of the examples: the distribution when a node was built and the class-distribution when a node was a leaf and the weighted sum of the class-distributions in the leaves descendant of that node.

Adaptation Methods. The Adaptation model characterizes the adaptation of the decision model. Here, we consider two different approaches:

1. *Blind Methods:* Methods that adapt the learner at regular intervals without considering whether changes have really occurred. Examples include methods that weight the examples according to their age and methods that use time-windows of fixed size.
2. *Informed Methods:* Methods that only modify the decision model after a change was detected. They are used in conjunction with a detection model.

Blind methods adapt the learner at regular intervals without considering whether changes have really occurred. Examples of this approach are *weighted examples* and *time windows* of fixed size. Weighted examples are based on the simple idea that the importance of an example should decrease with time (references related to this approach can be found in: [13,12,18,19,23]).

Decision Model Management. Model management characterizes the number of decision models needed to maintain in memory. The key issue here is the assumption that data generated comes from multiple distributions, at least in the transition between contexts. Instead of maintaining a single decision model several authors propose the use of multiple decision models. A seminal work is the system presented by Kolter and Maloof [15]. The Dynamic Weighted Majority algorithm (DWM) is an ensemble method for tracking concept drift. DWM maintains an ensemble of base learners, predicts target values using a weighted-majority vote of these *experts*, and dynamically creates and deletes experts in response to changes in performance. DWM maintains an ensemble of predictive models, each with an associated weight. Experts can use the same algorithm for training and prediction, but are created at different time steps so they use different training set of examples. The final prediction is obtained as a weighted vote of all the experts. The weights of all the experts that misclassified the example are decreased by a multiplicative constant β . If the overall prediction

is incorrect, a new expert is added to the ensemble with weight equal to the total weight of the ensemble. Finally, all the experts are trained on the example. Later, the same authors present the AddExp algorithm [14], a variant of DWM extended for classification and regression, able to prune some of the previous generated experts.

Another important aspect is the *granularity* of decision models. When drift occurs, it does not have impact in the whole instance space, but in particular regions. Adaptation in global models (like naive Bayes, discriminant functions, SVM) require reconstruction of the decision model. Granular decision models (like decision rules and decision trees¹ can adapt parts of the decision model. They only need to adapt those parts that cover the region of the instance space affected by drift. An instance of this approach is the CVFDT algorithm [9] that generate alternate decision trees at nodes where there is evidence that the splitting test is no more appropriate. The system replaces the old tree with the new one when the last becomes more accurate.

3 A Drift Detection Method Based on Statistical Control

In most of real-world applications of Machine Learning data is collected over time. For large time periods, it is hard to assume that examples are independent and identically distributed. At least in complex environments it is highly provable that class-distributions changes over time. In this work we assume that examples arrive one at a time. The framework could be easy extended to situations where data comes on batches of examples. We consider the online learning framework: when an example becomes available, the decision model must take a decision (e.g. a prediction). Only after the decision has been taken, the environment reacts providing feedback to the decision model (e.g. the class label of the example). In the PAC learning model [20], it is assumed that if the distribution of the examples is stationary, the error rate of the learning algorithm (p_i) will decrease when the number of examples (i) increases². This sentence holds for any learning algorithm with infinite-capacity (e.g. decision trees, neural-networks, etc.). A significant increase in the error of the algorithm when trained using more examples, suggests a change in the intrinsic properties in the process generating examples and that the actual decision model is no more appropriate.

3.1 The Method

Suppose a sequence of examples, in the form of pairs (\mathbf{x}_i, y_i) . For each example, the actual decision model predicts \hat{y}_i , that can be either True ($\hat{y}_i = y_i$) or False ($\hat{y}_i \neq y_i$). For a set of examples, the error is a random variable from Bernoulli trials. The Binomial distribution gives the general form of the probability for the random variable that represents the number of errors in a sample of n examples.

¹ Nodes in a decision tree correspond to hyper-rectangles in particular regions of the instance space.

² For an infinite number of examples, the error rate will tend to the Bayes error.

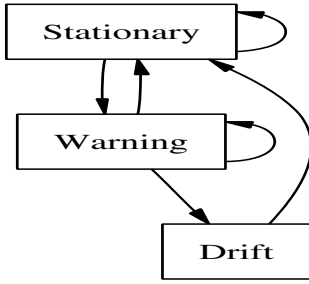


Fig. 1. The Space State Transition Graph

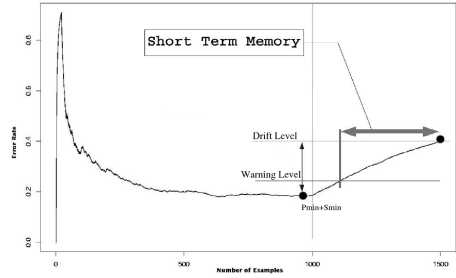


Fig. 2. Dynamically constructed Time Window. The vertical line marks the change of concept.

For each point i in the sequence, the error-rate is the probability of observe False, p_i , with standard deviation given by $s_i = \text{sqrt}(p_i(1 - p_i)/i)$. The drift detection method manages two registers during the training of the learning algorithm, p_{min} and s_{min} . For each new processed example i , if $p_i + s_i$ is lower than $p_{min} + s_{min}$ these values are updated.

For a sufficient large number of example, the Binomial distribution is closely approximated by a Normal distribution with the same mean and variance. Considering that the probability distribution is unchanged when the context is static, then the $1 - \alpha/2$ confidence interval for p with $n > 30$ examples is approximately $p_i \pm z * s_i$. The parameter z depends on the desired confidence level.

Suppose that in the sequence of examples, there is an example j with correspondent p_j and s_j . We define three possible states for the system:

- *In-Control*: while $p_j + s_j < p_{min} + \beta * s_{min}$. The error of the system is stable. The example j is generated from the same distribution of the previous examples.
- *Out-of-Control*: whenever $p_j + s_j > p_{min} + \alpha * s_{min}$. The error is increasing, and reach a level that is significantly higher from the past recent examples. With probability $1 - \alpha/2$ the current examples are generated from a different distribution.
- *Warning*: whenever the system is in between the two margins. The error is increasing but without reaching an action level. This is a not decidable state. The causes of error increase can be due to noise, drift, small inability of the decision model, etc. More examples are needed to make a decision.

The graph describing the state transition is presented in figure 1. It is not possible to move from a stationary state to a drift state without passing the warning state. If is possible to move from a warning state to a stationary state. For example, we can observe an increase of the error reaching the warning level, followed by a decrease. We assume that such situations corresponds to a *false alarms*, most probably due to noisy data, without changing of context.

We use a warning level to define the optimal size of the context window. The context window will contain the old examples that are on the new context and a minimal number of examples on the old context. Suppose that, in the sequence of examples that traverse a node, there is an example i with correspondent p_i and s_i . In the experiments described next the confidence level for warning has been set to 95%, that is, the warning level is reached if $p_i + s_i \geq p_{min} + 2 * s_{min}$. The confidence level for drift has been set to 99%, that is, the drift level is reached if $p_i + s_i \geq p_{min} + 3 * s_{min}$. Suppose a sequence of examples where the error of the actual model increases reaching the warning level at example k_w , and the drift level at example k_d . This is an indication of a change in the distribution of the examples. A new context is declared starting in example k_w , and a new decision model is induced using only the examples starting in k_w till k_d .

Figure 2 details the dynamic window structure. With this method of learning and forgetting we ensure a way to continuously keep a model better adapted to the present context.

3.2 Discussion and Limitations

A main characteristic of the detection method we present here is the use of the variance of the error estimate to define the action boundaries. The boundaries are not fixed but decrease with the confidence increase in the error estimates (as far as we have more points for estimation). This method could be applied to any learning algorithm. It could be directly implemented inside online and incremental algorithms, and could be implemented as a wrapper to batch learners. The proposed method assumes that in the flow of training examples there are sequences of examples with a stationary distribution, and this sequences are large enough to ensure some kind of *convergence* of the classifier. We denote those sequences of examples as *context*.

From the practical point of view, when a drift is signalled, the method defines a dynamic time window of the more recent examples used to train a new classifier. Here the key point is how fast the change occurs. If the change occurs at slow rate, the prequential error will exhibit a small positive slope. More examples are needed to evolve from the warning level to the action level and the window size increases. For abrupt changes, the increase of the prequential error will be also abrupt and a small window (using few examples) will be chosen. In any case the ability of training a new accurate classifier depends on the rate of changes and the capacity of the learning algorithm to converge to a stable model. The last aspect depends on the number of examples in the context.

3.3 Local Drift Detection

In the previous section we described a general method for change detection. The method can be applied as a wrapper with any classification learning algorithm. The inconvenience of this approach is that a new decision model must be learned from scratch whenever a change is detected. This behaviour is useful if and only

if the concept change in the instance space as a whole, which is rare. In most cases, changes occur in some regions of the instance space [4]. Some decision models fit different models to different regions of the instance space. This is the case of rule learners and decision trees. In decision trees, each leaf (and each internal node) corresponds to a hyper-rectangle in the instance space. The root node covers all the instance space, and descent nodes covers sub-spaces of the space covered by their parent node.

This is the rationale behind the algorithm that follows. Each inner node in a decision tree is bounded with a drift detection mechanism that traces the training examples traversing the node. If it triggers a change only the sub-tree rooted at that node is pruned. The node becomes a leaf, and a new subtree starts to be learned. In this way, changes that only affect particular regions in the instance space can be captured locally by the nodes that cover these regions. A major advantage of this schema is the reduced cost of updating the decision model.

The drift detection mechanism could be implemented in several ways. One possibility is to consider very simple classifiers (like the majority class) in each internal node of the tree. We have obtained a good traded between simplicity and faster detection rates using naive Bayes classifiers. We have implemented an incremental decision tree that maintains at each inner node a naive Bayes classifier. The decision tree algorithm is oriented towards processing data streams. It is based on VFDT algorithm [21] oriented towards continuous attributes. It uses the splitting criteria presented in UFFT [6] algorithm. The decision model starts with a single leaf. When there is statistical evidence in favor to a splitting test, the leaf becomes a decision node, and two descendant leaves are generated. The leaves store the sufficient statistics to computing the merit (information gain) of each splitting test. These sufficient statistics constitute also a naive Bayes classifier used for drift detection. After the leaf becomes a node, all examples that traverse the node will be classified by the naive-Bayes. The basic idea of the drift detection method is to control this online error-rate. If the distribution of the examples is stationary, the error rate of naive-Bayes decreases. If there is a change on the distribution of the examples the naive-Bayes error increases. When a naive Bayes detects a statistical significant increase of the error, it is signalled a change in the distribution of the examples that falls in the instance space covered by the corresponding node. This suggest that the splitting-test installed at this node is no longer appropriate. The subtree rooted at that node is pruned, and the node becomes a leaf. All the sufficient statistics of the leaf are initialized.

An advantage of this method is that it continuously monitors the online error of naive Bayes. It can detect changes at any time. All internal nodes contain naive Bayes to detect local changes. This correspond to detect shifts in different regions of the instance space. Nodes near the root should be able to detect abrupt changes in the distribution of the examples, while deeper nodes should detect localized, smoothed and gradual changes.

4 Experimental Evaluation

In this section we describe the evaluation of the proposed method in two scenarios: at a global level, as a wrapper over a decision tree learning algorithm³, and at local level where drift detection is used in each node of a decision tree.

We use two datasets: the SEA concepts, previously used in concept drift detection [22] and the Electricity Market Dataset, a real-world problem previously used in [8]. Using artificial datasets allow us to control relevant parameters to evaluate drift detection algorithms. For example, we can measure how fast the detection algorithm reacts to drift.

Evaluation Methodology in Drift Environments. Changes occur over time. Drift detection algorithms assume that data is sequential. Standard evaluation methods, like cross-validation, assume examples are independent and identically distributed. How to evaluate learning algorithms in sequential data? The main constraint is that we must guarantee that all test examples have a time-stamp larger than those in the training data. Two viable alternatives are:

1. Train-Test: hold-out an independent test set. Given a large enough training and test sets, the algorithm learns a model from the training set and makes predictions for test set examples. It is assumed that all test examples have a time-stamp larger than those in training data.
2. Predictive Sequential (*Prequential*) [3], where the error of a model is computed from the sequence of examples. For each example in the sequence, the actual model makes a prediction for the next example. The prediction is then compared to the observed value. The prequential-error is a computed based on an accumulated sum of a loss function between the prediction and observed values.

The train-test method provides a single point estimate of the error. The prequential approach provides much more information. It uses all available data for training and test, providing a kind of learning curve that traces the evolution of the error. The main problem of this approach is its sensitivity to the order of the examples.

Error rate is the most relevant criteria for classifier evaluation. Other criteria relevant for change detection methods include:

1. The number of examples required to detect a change after the occurrence of a change.
2. Resilience to noise. That is, ability to not detect drift when there is no change in the target concept. We designate this type of errors as Type 1 error.
3. Type 2 error: does not detect drift when drift exist.

Evaluation on Stationary Environments. We have tested the method using the dataset *ADULT* [2]. This dataset was created using census data in a specific point of time. Most probable, the concept is stable. Using a decision tree as

³ We have used the CART implementation in R.

Table 1. Error rate of Decision Tree algorithms on SEA Concepts. The table reports the error on an independent test set from the last concept. The second and third columns report the error setting on/off the drift detection method as a wrapper over the learning algorithm. Similarly, the fourth and fifth columns refer to local drift detection.

	Wrapper		Local		CVFDT
	No detection	Detection	No Detection	Detection	
Mean	15.71	13.72	14.65	12.51	14.72
Variance	0.29	0.28	1.26	1.89	1.06

inducer, the proposed method as a wrapper did not detect any drift; when used locally we observed 1.7% of drift signals (using 10-fold cross validation), all of them triggered by deeper nodes in the tree. This is an important aspect, because it presents evidence that the method is robust to false alarms.

Results on Artificial Domains - SEA concepts. The goal of these experiments is to study the effect of the proposed drift detection method on the generalization capacity of each learning algorithm. The use of artificial datasets allow us to design controlled experiments, defining where drift occurs and at which rate it occurs. We show the method is efficient in detecting drift, and is independent of the learning algorithm.

The *sea concepts* dataset has been used as a benchmark in drift detection problems [22]. For our analysis, it is interesting to describe the process used to generate data. We first generate 60000 random points in a three-dimensional space. All features take values between 0 and 10. Those points are divided into 4 blocks with different concepts. In each block, a data point belongs to class + if $f_1 + f_2 < \theta$, where f_i represents feature i and θ represents a threshold value between the two classes. We use threshold values of 8,9,7, and 9.5 for the four data blocks. We introduce class noise by changing the class label in 10% of the data points in each block. We should note that attribute f_3 is irrelevant. Transitions between concepts are abrupt. The first transition (at example 15k) affects a smaller region of the instance space than the other two changes (at examples 30k and 45k).

The Electricity Market Dataset. The data used in this experiments was first described by M. Harries [8]. The ELEC2 dataset contains 45312 instances dated from 7 May 1996 to 5 December 1998. Each example of the dataset refers to a period of 30 minutes, i.e. there are 48 instances for each time period of one day. Each example on the dataset has 5 fields, the day of week, the time stamp, the NSW electricity demand, the Vic electricity demand, the scheduled electricity transfer between states and the class label. The class label identifies the change of the price related to a moving average of the last 24 hours. The class level only reflects deviations of the price on a one day average and removes the impact of longer term price trends. The interest of this dataset is that it is a real-world dataset. It is not known if there exists drift or when it occurs.

We consider two problems. The first problem (last-day) consists in short term predictions. The test set contains the examples corresponding to the last day. The

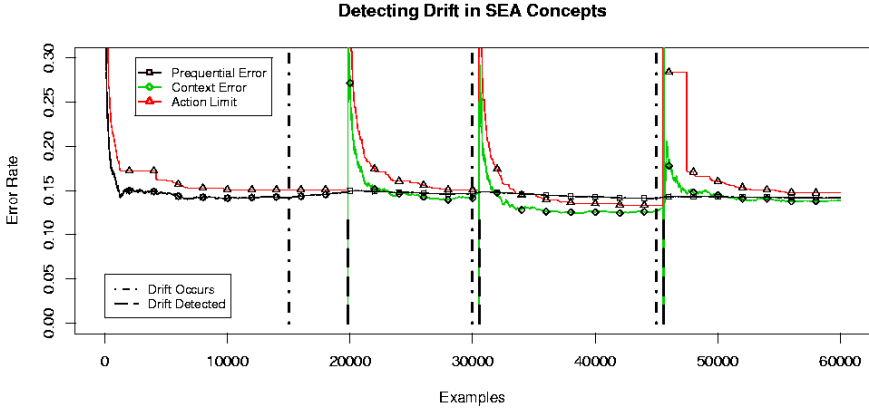


Fig. 3. Detecting Drift in SEA concepts. The figure plots the prequential error, the context error and the drift action level using a Decision tree learning algorithm.

Table 2. Evolution of the context error at the end of each Context of a decision tree and a GLM model. For each context we show the number of example needed to reach the warning and action levels. For both algorithms no false alarm was signalled.

Algorithm	Context 1		Context 2		Context 3		Context 4	
	Error	Warn Detect	Error	Warn Detect	Error	Warn Detect	Error	
Decision Tree	14.18	1788 4822	14.12	143 527	12.63	150 526	13.84	
GLM	11.30	290 1220	11.03	245 604	11.25	70 507	10.52	

Table 3. Error rate on an independent test set using examples from the last day and last week on the Elect problem. The second and third columns report the error setting on/off the drift detection method as a wrapper over the learning algorithm. Similarly, the fourth and fifth columns refer to local drift detection.

	Wrapper		Local		CVFDT
	No detection	Detection	No Detection	Detection	
Last Week	23.52	20.45	25.89	19.64 (8)	18.21
Last Day	18.75	12.50	16.67	12.50 (8)	12.50

second problem (last-week) consists in predicting the changes in the prices relative to the last week of examples recorded. The learning algorithm used is the implementation of CART available in R. The algorithm learns a model from the training data. We use the detection algorithm as a wrapper over the learning algorithm. After seeing all the training data, the final model classifies the test data. In the 1-day dataset, the trees are built using only the last 3836 examples on the training dataset. In the 1-week dataset, the trees are built with the 3548 most recent examples. This is the data collected since 1998/09/16. Table 3 shows the error rate obtained with the 1-day and 1-week prediction. In both problems the

controlling drift allow substantial improvements in performance. This is an evidence indicating the presence of drift in the dataset and the advantages of using control mechanisms to detect and reacting to drift in real-world applications.

5 Conclusions

In this work we analyse and discuss change detection methods for machine learning. We present a method for concept drift detection in the distribution of the examples. The drift detection method continuously monitors the prequential error of a learning algorithm searching for significant deviations from previous stable state. It can be applied to problems where the information is available sequentially over time. The method can be used either as a wrapper over any learning algorithm or locally at each internal node of a decision tree. When a drift is detected, the wrapper approach requires learning a new model, the local approach only require to adapt the part of the decision model covering the region of the instance space affected by the change. In the experimental section we present controlled experiments using artificial data that illustrate the ability of the method for detecting abrupt and smoothed changes. The method is resilient to false alarms, and improves the learning capability of learning algorithms when modelling non-stationary problems. We are now exploring the application of the method with other loss-functions. Preliminary results in the regression domain using *mean-squared error* loss function confirm the results presented here.

Acknowledgements. The authors reveal its gratitude to the projects Adaptive Learning Systems II(POSI/EIA/55340/2004), RETINAE, and FEDER through the plurianual support to LIACC.

References

1. Michele Basseville and Igor Nikiforov. *Detection of Abrupt Changes: Theory and Applications*. Prentice-Hall Inc, 1987.
2. C. Blake, E. Keogh, and C.J. Merz. UCI repository of Machine Learning databases, 1999.
3. A. P. Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society-A*, 147:278–292, 1984.
4. Wei Fan. Systematic data selection to mine concept-drifting data streams. In J. Gehrke and W. DuMouchel, editors, *Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining*. ACM Press, 2004.
5. J. Gama, R. Fernandes, and R. Rocha. Decision trees for mining data streams. *Intelligent Data Analysis*, 10(1):23–46, 2006.
6. João Gama, Pedro Medas, and Pedro Rodrigues. Learning decision trees from dynamic data streams. In Hisham Haddad, Lorie M. Liebrock, Andrea Omicini, and Roger L. Wainwright, editors, *Proceedings of the 2005 ACM Symposium on Applied Computing*, pages 573–577. ACM Press, March 2005.
7. M. Harries, C. Sammut, and K. Horn. Extracting hidden context. *Machine Learning*, 32:101, 1998.

8. Michael Harries. Splice-2 comparative evaluation: Electricity pricing. Technical report, The University of South Wales, 1999.
9. Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106. ACM Press, 2001.
10. Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *VLDB 04: Proceedings of the 30th International Conference on Very Large Data Bases*, pages –. Morgan Kaufmann Publishers Inc., 2004.
11. R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 2004.
12. R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In Pat Langley, editor, *Proceedings of ICML-00, 17th International Conference on Machine Learning*, pages 487–494, Stanford, US, 2000. Morgan Kaufmann Publishers.
13. R. Klinkenberg and I. Renz. Adaptive information filtering: Learning in the presence of concept drifts. In *Learning for Text Categorization*, pages 33–40. AAAI Press, 1998.
14. J. Kolter and M. Maloof. Using additive expert ensembles to cope with concept drift. In L Raedt and S. Wrobel, editors, *Machine Learning, Proceedings of the 22th International Conference*. OmniPress, 2005.
15. Jeremy Z. Kolter and Marcus A. Maloof. Dynamic weighted majority: A new ensemble method for tracking concept drift. In *Proceedings of the Third International IEEE Conference on Data Mining*, pages 123–130. IEEE Computer Society, 2003.
16. I. Koychev. Gradual forgetting for adaptation to concept drift. In *Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning. Berlin, Germany*, pages 101–106, 2000.
17. I. Koychev. Learning about user in the presence of hidden context. In *Proceedings of Machine Learning for User Modeling: UM-2001.*, 2001.
18. C. Lanquillon. *Enhancing Text Classification to Improve Information Filtering*. PhD thesis, University of Magdeburg, Germany, 2001.
19. M. Maloof and R. Michalski. Selecting examples for partial memory learning. *Machine Learning*, 41:27–52, 2000.
20. Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.
21. Domingos P. and Hulten G. Mining High-Speed Data Streams. In *Proceedings of the Association for Computing Machinery Sixth International Conference on Knowledge Discovery and Data Mining*, pages 71–80. ACM Press, 2000.
22. W. Nick Street and YongSeog Kim. A streaming ensemble algorithm SEA for large-scale classification. In *Proc. seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 377–382. ACM Press, 2001.
23. Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23:69–101, 1996.

A Fast Algorithm for Maintenance of Association Rules in Incremental Databases

Xin Li, Zhi-Hong Deng*, and Shiwei Tang

National Laboratory on Machine Perception,
School of Electronics Engineering and Computer Science
Peking University, Beijing 100871, China
lix@cis.pku.edu.cn
zhdeng@cis.pku.edu.cn
tsw@pku.edu.cn

Abstract. In this paper, we propose an algorithm for maintaining the frequent itemsets discovered in a database with minimal re-computation when new transactions are added to or old transactions are removed from the transaction database. An efficient algorithm called EFPIM (Extending FP-tree for Incremental Mining), is designed based on EFP-tree (extended FP-tree) structures. An important feature of our algorithm is that it requires no scan of the original database, and the new EFP-tree structure of the updated database can be obtained directly from the EFP-tree of the original database. We give two versions of EFPIM algorithm, called EFPIM1 (an easy vision to implement) and EFPIM2 (a fast algorithm), they both mining frequent itemsets of the updated database based on EFP-tree. Experimental results show that EFPIM outperforms the existing algorithms in terms of the execution time.

1 Introduction

Data mining or knowledge discovery in databases has attracted much attention in database research since it was first introduced in [1]. This is due to its wide applicability in many areas, including decision support, market strategy and financial forecast. Many algorithms have been proposed on this problem such as Apriori [2] (and its modifications), hash based algorithm [3], FP-growth [4, 5], and vertical method [6].

The rules discovered from a database only reflect the current state of the database. However, the database is not a static database, because updates are constantly being applied to it. Because of these update activities, new association rules may appear and some existing association rules would become invalid at the same time. To re-mine the frequent itemsets of the whole updated database is clearly inefficient, because all the computations done in the previous mining are wasted. Thus, maintenance of discovered association rules is an important problem.

The FUP algorithm proposed in [7] and its developed form FUP2 algorithm [8] are both similar to Apriori-like algorithms, which has to generate large number of candidates and repeatedly scan the database. In [9], the negative border is maintained along

* Corresponding author.

with the frequent itemsets to perform incremental updates. This algorithm still requires a full scan of the whole database if an itemset outside the negative border gets added to the frequent itemsets or its negative border. A recent work called AFPIM [10] is designed to efficiently find new frequent itemsets based on adjusting FP-tree structure. However, it will cost much a lot to adjust FP-tree of the original database according to the changed transactions.

In this paper, an algorithm called EFPIM (Extending FP-tree for Incremental Mining), is designed to efficiently find new frequent itemsets with minimum re-computation when new transactions are added to or old transactions are removed from the database. In our approach, we use the structure EFP-tree, which is equal to the FP-tree when the system runs the mining algorithm for the first time and an expanded form of FP-tree structure during the next incremental mining. The EFP-tree of the original database is maintained in addition to the frequent itemsets. Without needing to re-scan the original database, the EFP-tree of the updated database is obtained from the preserved EFP-tree. We will give two visions of EFPIM algorithm, called EFPIM1 (an easy vision to implement) and EFPIM2 (a fast algorithm), they both mining frequent itemsets of the updated database based on EFP-tree.

2 Problem Description

Let DB be a database of original transactions, and $I = \{i_1, i_2, \dots, i_m\}$ be the set of *items* in DB . A set of items is called an *itemset*. The *support count* of an itemset X in DB , denoted as $Sup_{DB}(X)$, is the number of transactions in DB containing X . Given a *minimum support threshold* $s\%$, an itemset X is called a *frequent itemset* in DB if $Sup_{DB}(X) \geq |DB| \times s\%$.

Let L_{DB} refer to the set of frequent itemsets in DB . Moreover, let db_+ (db_-) denote the set of added (deleted) transactions, and $|db_+|$ ($|db_-|$) be the number of added (deleted) transactions. The updated database, denoted as UD , is obtained from $DB \cup db_+ - db_-$. Define L_{UD} to denote the new set of frequent itemsets in UD . The number of transactions in UD , $|UD|$, is $|DB| + |db_+| - |db_-|$. Also, the support count of an itemset X in UD , denoted as $Sup_{UD}(X)$, is equal to $Sup_{DB}(X) + Sup_{db_+}(X) - Sup_{db_-}(X)$. In other words, the update problem of association rules is to find L_{UD} efficiently.

3 Extending FP-Tree for Incremental Mining (EFPIM)

3.1 Basic Concept

In this paper, a lesser threshold, called *pre-minimum support*, is specified. For each item X in a database, if its support count is no less than pre-minimum support, X is named a *pre-frequent item*. Otherwise, X is an *infrequent item*. If the support of a pre-frequent item X is no less than minimum support also, X is called a *frequent item*. Our strategy is designed based on extending FP-tree structure to maintain the updated frequent itemsets efficiently. The following information after mining the original database DB needs to be maintained: 1) all the items in DB along with their support count in DB , and 2) the FP-tree of DB for pre-frequent items in DB .

When the system runs the mining algorithm for the first time, we use the algorithm introduced in [4], and get a FP-tree. In the FP-tree of DB , each path follows the frequency descending order of pre-frequent items in DB . After insertion or deletion occurs in DB , some items may become infrequent ones in UD . These items have to be removed from the FP-tree. In addition, the paths of nodes may become unordered. If we adopt the method [10] of adjusting the paths of nodes to make it follow the order of pre-frequent items in UD , it will find that this step costs much a lot to adjust FP-tree according to the changed transactions.

We propose a new structure called *EFP-tree* which is equal to the FP-tree when running the mining algorithm for the first time. After database is updated, we first remove those items becoming infrequent in UD from the EFP-tree. Then expand (decrease) the EFP-tree with the transactions of db_+ (db_-), which are arranged as the order of pre-frequent items in DB . The EFP-tree is still a prefix-tree structure for storing compressed and crucial information in transactions, though it is not compressed as constrictive as FP-tree. However, it still conserves all the information in UD without a loss. When the database's update is small in proportion to DB , using EFP-tree is really an effective and efficient method, which will be discussed later.

3.2 An Easy Update Algorithm EFPIM1

Here, we give an easy update algorithm EFPIM1 based on the EFP-tree structure defined above and FP-growth algorithm [4].

Algorithm EFPIM1

Input: the pre-maintained information of DB , db_+ , and db_- .

Output: the frequent itemsets in UD .

1. Read in the items in DB and their support counts in DB .
2. Scan db_+ and db_- once. Compute the support count of each item X in UD .
3. Judge whether all the frequent items of UD are covered in EFP-tree of DB .
 - 3.1 If there exists a frequent item of UD not in the EFP-tree, scan the whole UD to reconstruct an EFP-tree according to the pre-frequent items in UD .
 - 3.2 Otherwise, read in the stored EFP-tree of DB .
 - 3.2.1 Remove items becoming infrequent in EFP-tree.
 - 3.2.2 Arrange items of each transaction in db_+ according to the support descending orders in DB , and insert them into the EFP-tree. Similarly, transactions in db_- are removed from EFP-tree by decreasing count in nodes.
4. Apply FP-Growth algorithm [4] on the updated EFP-tree.
5. Store the support counts of items and EFP-tree of UB .

Step3 and step3.1 explain the purpose of using pre-frequent items. Thinking of the situation that there is an item x appearing frequent in db_+ but not existing in the original EFP-tree (which means not frequent in DB), however, it appears frequent in UD ($Sup_{UD}(X) > min_sup$). If we omit step3, the output will not contain all the situations. If we do not construct EFP-tree according to the pre-frequent items, the test in step3 will often lead to step3.1 (which means doing the re-mining over all).

[Example 1] Let the original DB be illustrated in Table1 (a). The minimum support is 0.2 and the pre-minimum support is 0.15. Scan DB once. The items with support counts no less than $2(13*0.15=1.95)$ are pre-frequent items. Thus, A, B, C, D, E, and F are pre-frequent items. After sorting them in support descending order, the result is F:7, B:6, C:5, E:4, D:3, and A:2. The FP-tree of DB is constructed as Fig. 1(a).

Table 1. Sample database

T_ID	Items	After ordering
1	BDEF	FBED
2	F	F
3	ABEF	FBEA
4	CH	CH
5	BF	FB
6	B	B
7	ABEF	FBEA
8	CG	CG
9	BF	FB
10	CDE	CED
11	F	F
12	CD	CD
13	C	C

(a)DB

T_ID	Items	After ordering
14	BCDEF	FBCED
15	BDEF	FBED
16	BCD	BCD
17	BD	BD
18	D	D

(b) db+

Then five transactions are inserted into DB , shown as Table 1(b). In new database UD , a pre-frequent item must have support counts no less than 3 (i.e. $(13+5)*0.15=2.7$). Therefore, the pre-frequent items in UD , shown in the order as before, are F:9, B:10, C:7, E:6, D:8, A:2. Accordingly, the constructed FP-tree of UD is shown as Fig. 1(b). Then apply FP-growth algorithm to find out all the itemsets.

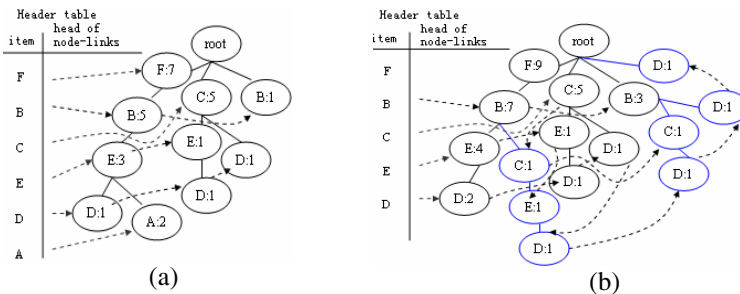


Fig. 1. The EFP-tree of example 1

3.3 A Fast Update Algorithm EFPIM2

The EFPIM1 algorithm could work well, and it can be implemented much easier than methods like AFPIM. Further more, we have the following conclusions by analyzing the algorithm described above:

First, due to the nature of EFP-tree, the new EFP-tree constructed for an updated database is just an extension to the old one, which means that all existing nodes and their arrangement did not change at all. So we can reuse old trees to the utmost extent. Note that extending an existing tree is relatively much faster than constructing a new tree, especially when the tree is big resulting from prolific or long patterns.

Second, based on its divide-and-conquer idea, FP-growth decomposes the mining task into a set of smaller tasks for mining confined patterns in conditional databases. Although EFP-tree is rather compact, its construction of the first time still needs two scans of a conditional database. So it would be beneficial to extend an existing one. Materialization plays a very important role and provides a foundation in our work.

These features and considerations combined together form the core of our new algorithm EFPIM2. This algorithm is much similar to EFPIM1, but calling EFP-growth instead of FP-growth algorithm. Limited by space, we only list the different step compared with EFPIM1.

Algorithm EFPIM2

- 3.2.1 Call *EFP-tree_extension*, return the EFP-tree of *UD*.
4. Apply *EFP-growth algorithm* on the EFP-tree of *UD*.

The method *EFP-tree_extension* and *EFP-growth* are outlined as follows.

Algorithm EFP-tree_extension

Input: the EFP-tree of *DB*, db_i , db_j

Output: extended EFP-tree

1. For each item that is pre-frequent in *DB* and becoming infrequent in *UD*, remove the corresponding nodes from EFP-tree and the header table.
2. Arrange items of each transaction in db_i according to the support descending orders in *DB*, and insert them into the EFP-tree. If there are some new items appearing in db_i but not in *DB*, arrange them according to their descending orders, and put them behind those items which have appeared in *DB*. So the new items will be in the bottom of the EFP-tree of *UB*.
Similarly, each transaction in db_j is removed from EFP-tree by decreasing the count in the nodes.

Algorithm EFP-growth

Input: EFP-tree, pattern α

Output: the frequent itemsets in *UD*.

1. For each item a_i in EFP-tree, which appears in db_i but not in *DB*, do

- Generate pattern $\beta = \alpha \cup a_i$ with support = a_i . support;
 Construct β 's conditional pattern base and then
 construct β 's conditional EFP-tree $Tree\beta$;
 If $Tree\beta \neq NULL$, then call $FP\text{-}growth(Tree\beta, \beta)$ and store
 the support counts of items and EFP-tree of $Tree\beta$.
- For each item a_i in EFP-tree, which appeared in DB , do
 $\beta = \alpha \cup a_i$;
 Read in the stored β 's conditional EFP-tree as $S\text{Tree}\beta$;
 Construct β 's conditional pattern base;
 Call $EFP\text{-}tree_extension(S\text{Tree}\beta, db_+, db_-)$ and get β 's
 new conditional FP-tree $Tree\beta$;
 Call $EFP\text{-}growth(Tree\beta, \beta)$ and store the support counts
 of items and EFP-tree of $Tree\beta$.

Obviously, any item appearing in db_+ but not in DB can not have a stored conditional EFP-tree when doing mining on DB . So we have to construct a new one and call FP-growth method, since the recursions starting from this item have never been done before. But for the items have already appeared in DB , it needs not to reconstruct all. We just extend the stored EFP-tree and call EFP-growth recursively. The correction of this process is guaranteed by that each path of the EFP-tree of UD is arranged as the order of pre-frequent items in DB .

[Example 2] Consider when we construct item E's conditional database and conditional EFP-tree in example 1. In DB , E's conditional database is $\{\{C:1\}, \{F:3, B:3\}\}$, and its conditional EFP-tree is shown in Fig. 2(a). When database becomes to UD , the conditional database of item E becomes $\{\{C:1\}, \{F:3, B:3\}, \{F:1, B:1\}, \{F:1, B:1, C:1\}\}$, and its conditional EFP-tree is show in Fig. 2(b).

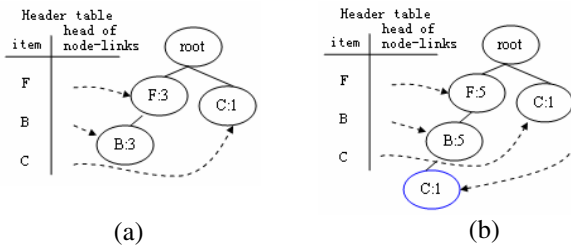


Fig. 2. The EFP-tree of example 2

We can find that the EFP-tree can be directly expanded from the old structure.

4 Experiments

To evaluate the performance of EFPIM algorithm, the algorithms EFPIM1, EFPIM2 are implemented on a personal computer. In this section, we present a performance comparison of re-mining by re-executing FP-growth.

The experiments are performed on synthetic data generated using the same technique as in [1]. The parameter Tx.Iy.Dm.dn is used to denote that average size of the transactions $|T|=x$, average size of the potentially frequent itemsets $|I|=y$, number of transactions $|D|=m*1000$, and number of inserted/deleted transactions $|d_{+}|/|d_{-}|=n*1000$. The number of various items is 1000.

We first compare the execution time of re-executing FP-growth, EFPIM1 and EFPIM2 on an updated database T10.I4.K10.d1. As shown in Fig. 3(a), in general, the smaller the minimum support is, the larger the speed-up ratio of EFPIM2 over EFPIM1 and re- executing method. The reason is that a small minimum support will induce a large number of frequent itemsets, which greatly increase the computation cost. In Fig. 3(b), the chart shows the ratio of execution time by two EFPIM algorithms when comparing with the re-mining method.

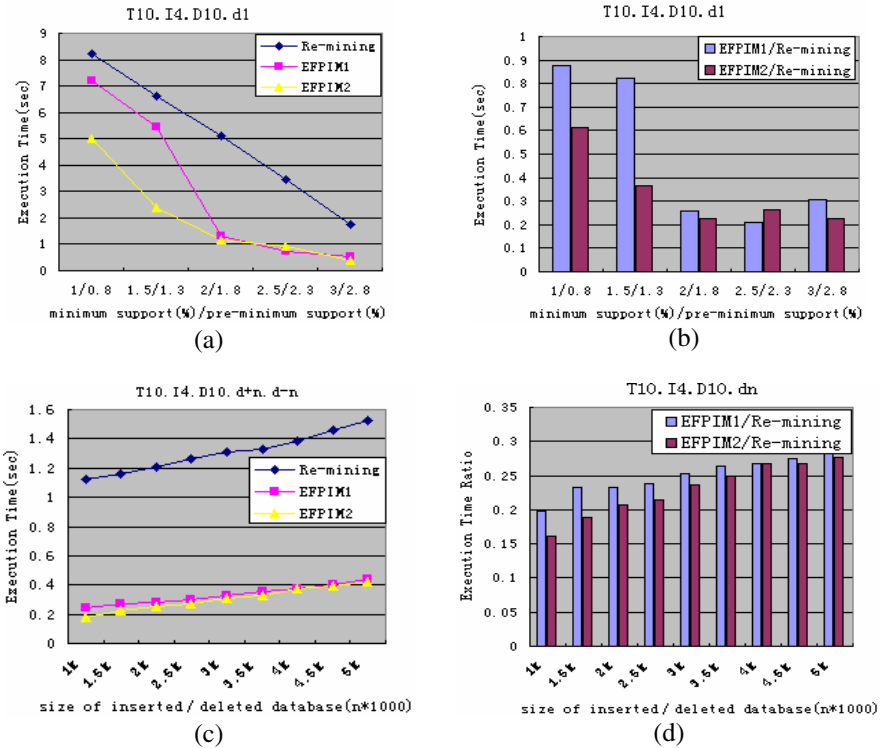


Fig. 3. Experiment Results

We then evaluate the effect of the size of updates on these algorithms. The minimum support is 3% in Fig. 3(c). Fig. 3(d) shows that two EFPIM algorithms are much faster than re-mining method.

5 Conclusion

In this paper, an efficient and general incremental updating technique is proposed for updating frequent itemsets when old transactions are removed from or new transactions are added into a transaction database. This approach uses EFP-tree structure constructed from a previous mining to reduce the possibility of re-scanning the updated database. The EFP-tree structure of the updated database is obtained, by extending the previous EFP-tree, to discover the corresponding frequent itemsets. Performance studies show that the proposed EFPIM1 and EFPIM2 algorithms are significantly faster than the re-mining by re-executing FP-growth algorithms. In particular, it works well for small minimum support setting.

Recently, there have been some interesting studies at mining maximal frequent itemsets [11] and closed frequent itemsets [12, 13]. The extension of our technique for maintaining these special frequent itemsets is an interesting topic for future research.

References

1. R. Agrawal, T. Imielinski, and A. Swami. "Mining Association Rules between Sets of Items in Large Databases," Proceedings of ACM SIGMOD, May 1993, 207-216
2. R. Agrawal and R.Srikant. "Fast algorithm for mining Association rules," In VLDB'94, 487-499.
3. Park J S et al. "An effective hash based algorithm for mining of association rules," In Proceedings of ACM SIGMOD Conference on Management of Data, May 1995, 175-186
4. J. Han, J. Pei, and Y. Yin. "Mining Frequent Patterns without Candidate Generation," in Proceedings of the ACM SIGMOD Int. Conf. on Management of Data, 2000, 1-12
5. J. Han and J. Pei. "Mining frequent patterns by pattern-growth: methodology and implications," In SIGKDD'00, 14-20.
6. Zaki and K. Gouda. "Fast vertical mining using diffsets," In SIGKDD'03, 326-335
7. D.W. Cheung, J. Han, V.T. Ng, and C.Y. Wong. "Maintenance of Discovered Association Rules in Large Databases: An Incremental Update Technique," In: Proceedings of International Conference on Data Engineering, 1996, 106-114
8. D.W. Cheung, S.D. Lee, and Benjamin Kao. "A General Incremental Technique for Maintaining Discovered Association Rules," in Proc. of the 5th International Conference on Database Systems for Advanced Applications, 1997, 185-194
9. S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka. "An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases," in Proc. of 3rd International conference on Knowledge Discovery and Data Mining, 1997, 263-266
10. Jia-Ling Koh and Shui-Feng Shieh. "An Efficient Approach for Maintaining Association Rules Based on Adjusting FP-Tree Structures," DASFAA 2004, 417-424
11. D. Burdick, M. Calimlim, and J. Gehrke. "MAFIA: A maximal frequent itemset algorithm for transactional databases," In ICDE'01, 443-452.
12. M. Zaki and C. Hsiao. "CHARM: An efficient algorithm for closed itemset mining," In SDM'02, 12-28
13. J. Y. Wang, J. Han, and J. Pei. "CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets," In SIGKDD'03, 236-245

Extending OLAP with Fuzziness for Effective Mining of Fuzzy Multidimensional Weighted Association Rules

Mehmet Kaya¹ and Reda Alhajj^{2,3}

¹ Dept. of Computer Eng, Firat University, Elazig, Turkey

² Dept. of Computer Science, University of Calgary, Calgary, Alberta, Canada
alhajj@cpsc.ucalgary.ca

³ Dept. of Computer Science, Global University, Beirut, Lebanon

Abstract. This paper contributes to the ongoing research on multidimensional online association rules mining by proposing a general architecture that utilizes a *fuzzy data cube* combined with the concepts of weight and multiple-level to mine fuzzy weighted multi-cross-level association rules. We compared the proposed approach to an existing approach that does not utilize fuzziness. Experimental results on the adult data of the United States census in year 2000 demonstrate the effectiveness and applicability of the proposed fuzzy OLAP based mining approach.

Keywords: association rules, data mining, fuzzy data cube, multidimensional mining, OLAP, weighted mining.

1 Introduction

OLAP is attractive in data mining because the data is organized in a way that facilitates the “preprocess once query many” concept. The mining process is improved by eliminating the need to start with the raw data each time mining is required. Some approaches have already been developed to tackle the problem. For instance, the approach described in [3] reduces the retrieval time of itemset data. Pruning the database removes data, which is not useful [4]. Hidber [5] developed an algorithm to compute large itemsets online.

Han [6] proposed a model to perform data analysis on multidimensional data by integrating OLAP tools and data mining techniques. Another work by Han [7] integrates several data mining components to the architecture described in [6]. Finally, Han and Fu [8] introduced an Apriori based top-down progressive approach for multiple-level association rules mining from large transactional databases. Kamber *et al* [9] proposed a data cube model for mining multidimensional association rules. Lu *et al* [10] introduced the notion of multidimensional association rules by implementing several algorithms (as extensions of Apriori) for finding inter-transaction association rules. Tung *et al* [11] introduced and extensively studied a new solution for mining inter-transaction association rules; they implemented two algorithms (EH-Apriori and FITI) for this purpose. Agarwal and Yu [12] proposed an OLAP-style algorithm to compute association rules.

They achieved this by preprocessing the data effectively into predefined itemsets with corresponding support values more suitable for repeated online queries. Although these algorithms improved the online generation of association rules in response to changing requirements, it is still an open problem that needs more investigation; our approach presented in this paper is a major effort in this direction.

As a result, all of the studies reported so far on OLAP mining use data cubes with binary attributes or data cubes that discretize the domains of their attributes. However, none of the data cube based approaches encountered in the literature include quantitative attributes with fuzzified and weighted domains, which are more natural and understandable by humans. Motivated by this, we developed a novel approach for online fuzzy weighted association rules mining. We contribute to the ongoing research on multidimensional online data mining by proposing a general architecture that constructs and uses fuzzy data cubes for knowledge discovery. The idea behind introducing the fuzzy data cube architecture for online mining is to allow users to query a given database for fuzzy association rules based on different values of support and confidence. We present a method for multi-dimensional fuzzy association rules mining. To the best of our knowledge, this is the first attempt to utilize fuzziness in OLAP mining. The proposed method has been tested using a subset from the adult data of the United States census in year 2000. Also, the proposed method has been compared with the discrete approach of Srikant and Agrawal [1]. Experiments demonstrate the effectiveness and applicability of the proposed mining approach.

The rest of this paper is organized as follows. Section 2 covers basics used in the rest of the paper. Section 3 presents the proposed fuzzy data cube based mining method. Experimental and comparison results for 100K transactions extracted from the adult data of the United States census in year 2000 are reported and discussed in Section 4. Section 5 is summary and conclusions.

2 Basic Terminology and Model Basics

2.1 Fuzzy Association Rules

To elaborate on fuzzy association rules, consider a database of transactions T , its set of attributes I , and the fuzzy sets associated with quantitative attributes in I . Notice that each transaction t_i contains values of some attributes from I and each quantitative attribute in I is associated with at least two fuzzy sets. The target is to find out some interesting and potentially useful regularities, i.e., fuzzy association rules with enough support and high confidence. We use the following form for fuzzy association rules [2].

Definition 1. [*Fuzzy Association rules*] A fuzzy association rule is defined as:

$$\begin{array}{l} \text{If } X = \{x_1, x_2, \dots, x_p\} \text{ is } A = \{f_1, f_2, \dots, f_p\} \\ \text{then } Y = \{y_1, y_2, \dots, y_q\} \text{ is } B = \{g_1, g_2, \dots, g_q\}, \end{array}$$

where X, Y are itemsets, i.e., disjoint subsets of I , and A and B contain fuzzy sets associated with corresponding attributes in X and Y , respectively, i.e., f_i is a fuzzy set related to attribute x_i and g_j is a fuzzy set related to attribute y_j .

For a rule to be interesting, it should have enough support and high confidence, both defined within the fuzzy context as given next in the paper. Finally, an example fuzzy association rule may be written as:

If {quizzes, midterm, final} is {average, good, excellent}
 then {standing} is {very good};

This rule is simply read as if quizzes, midterm and final are, respectively, identified as average, good and excellent, then the standing is very good.

2.2 Multi-cross-level Association Rules Mining

Benefiting from the fuzzy data cube structure, we take the model one step further to deal with multi-cross-level mining in a fuzzy data cube. In such a case, large itemsets can include nodes from more than one level. As only terminal nodes appear in transactions and since each level has different minimum support threshold, the computation of support thresholds of such itemsets is not trivial; it is more complicated than those of non-cross-level itemsets. Support thresholds of cross-level itemsets are computed as follows.

Proposition 1. Let $A(A_1, A_2, \dots, A_q)$ be an itemset. By definition, items in an itemset cannot be ancestors or descendants of each other. Then, the support threshold of itemset A is: $\min_{i=1}^q (FMinSup(A_i))$, where $FMinSup(A_i)$ is the minimum fuzzy support value of the level of A_i .

Weighting Items at Multiple Concept Levels. In the approach proposed in this paper, we also realized the importance of assigning weights to items at multiple cross levels. Actually, the algorithms developed earlier to mine weighted association rules handle the single-concept level, e.g., [13,14], i.e., only items appearing in transactions are weighted to find more specific and important knowledge. However, we argue that it is necessary to specify weight values for classes or concepts at higher levels instead of actual items present in transactions. This is true because sometimes weighting internal nodes in a tree may be more meaningful and enough, while another time both an ancestor and its descendant may need to be weighted. Also, weighting the nodes at different levels adds an important measure of interestingness. Doing this permits users to control the mining results because users' intuition is included in the association rules mining process. Finally, we handle weighting at multiple levels as follows.

Proposition 2. Consider $m+1$ items, say x, y_1, y_2, \dots, y_m ; and assume that x is ancestor of y_1, y_2, \dots, y_m , i.e., the latter items are descendants of x . While weighting nodes at multiple concept levels, cases that may be observed are:

1. If only ancestor x is weighted, then all descendants of x get the same weight as x .
2. If the weights of all descendants of x are prespecified, then the weight of x is computed as follows:

$$W_x = \frac{\sum_{j=1}^m (TotalCount_{y_j} \cdot W_{y_j})}{TotalCount_x}$$

where $TotalCount_{y_i} = \sum_{i=1}^n (v_{iy_j})$, here n is the number of transactions and v_{iy_j} is the value from domain D_i of the descendant y_j .

3. If the weights of ancestor x and $m - 1$ of its descendants are given (assume the weight of descendant y_p is not given), then the weight of descendant y_p is computed with respect to the given weights:

$$W_{y_p} = \frac{TotalCount_x \cdot W_x - \sum_{j=1, j \neq p}^m (TotalCount_{y_j} \cdot W_{y_j})}{TotalCount_{y_p}}$$

4. If the weights of ancestor x and e (with $e < m$) of its descendants are given then the weight of each of the remaining $(m - e)$ descendants is computed as:

$$W_{y_{(e+k)}} = \frac{TotalCount_x \cdot W_x - \sum_{j=1}^{e+k-1} (TotalCount_{y_j} \cdot W_{y_j})}{\sum_{j=e+k}^m (TotalCount_{y_p j})}, \quad k=1, m - e$$

3 OLAP-Based Mining of Multidimensional Rules Without Repetitive Items

In this method, the correlation is among a set of dimensions, i.e., the items forming a rule come from different dimensions. Therefore, each dimension should be partitioned at the fuzzy set level.

Proposition 3. Consider a fuzzy data cube with 3 dimensions, if values of attributes x_i , y_j and z_k have membership degrees $\mu^m(x_i)$, $\mu^n(y_j)$, and $\mu^o(z_k)$ in fuzzy sets f_i^m , f_j^n and f_o^k , respectively, then the sharing rate, denoted SR , of the corresponding cell is computed as $\mu^m(x_i) \cdot \mu^n(y_j) \cdot \mu^o(z_k)$.

In fact, we can use functions other than *product*, e.g., *min*, *max*; but *product* gives simple and reasonable results. It takes membership of all dimensions in a data cube into account.

Proposition 3 may be generalized for n -dimensions. This way, the frequency for each itemset can be directly obtained from one cell of the fuzzy data cube. Each cell stores the product of the membership grades of different items, one item per dimension. In other words, each cell stores the product of the membership grades of three items, one from each of the three dimensions. The fuzzy support value of each cell is computed as follows:

$$FSupport(Y, F) = \frac{SR(x_i \cdot f_i^m, y_j \cdot f_j^n, z_k \cdot f_o^k)}{SumTotal}$$

Finally, it is worth pointing out that we can find more interesting rules by using advantages of the summarization of data in the fuzzy data cube.

4 Experimental Results

We performed some empirical tests on real-life data in order to evaluate the effectiveness and applicability of the proposed approach and to analyze their scalability. We have used real data set to construct and study dense fuzzy data cubes, and we have analyzed the scalability by constructing sparse fuzzy data

cubes. All the experiments were conducted on a Pentium IV 2GHz CPU with 512 MB of memory and running Windows XP. As the dataset is concerned, we constructed the fuzzy data cube using 12 attributes and 100K transactional records from the adult data of the United States census in year 2000.

Each set of experiments necessitates the construction of a fuzzy data cube which complies with the requirements of the method under investigation; for each dimension of the cube, a set of attributes was picked from the experimental census database. The density of the constructed fuzzy data cube is 77%; it is computed as follows:

$$Cube\ Density = \frac{\text{number of non-empty cells}}{\text{cube size}}$$

With 77% density, we can classify the constructed fuzzy data cube as dense. In all the experiments conducted for this part of the study, two different cases were considered as the per attribute number of fuzzy sets (FS) is concerned, namely 3 and 4 fuzzy sets, denoted FS3 and FS4, respectively. Also, in order to show the effectiveness of the fuzzy data cube over the traditional data cube in association rules mining, we compared our fuzzy approach with the discrete method proposed by Srikant and Agrawal [1]. For this purpose, we distributed the values of each quantitative attribute into 3 and 4 intervals, denoted Discrete3 and Discrete4, respectively. Finally, the conducted experiment consists of four tests to evaluate our approach with respect to the following dimensions: 1) number of large itemsets generated for different values of minimum support; 2) number of association rules generated for different values of minimum confidence; 3) number of association rules as function of the average weight; and 4) execution time. Here, note that in all the tests, unless otherwise specified, the minimum support value has been set to 15% for level 1, 5% for level 2 and 2% for level 3.

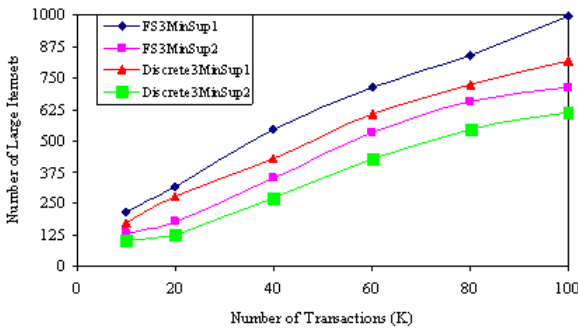


Fig. 1. Number of large itemsets vs number of transactions

In the experiments, we applied the proposed approach on a 3-dimensional fuzzy data cube; each dimension has 4 quantitative attributes. The results are shown in Figures 1-5. The curves plotted in Figure 1 show the change in the number of large itemsets for two variants of minimum support as the number of transactions increases. From Figure 1, it can be easily seen that the number of

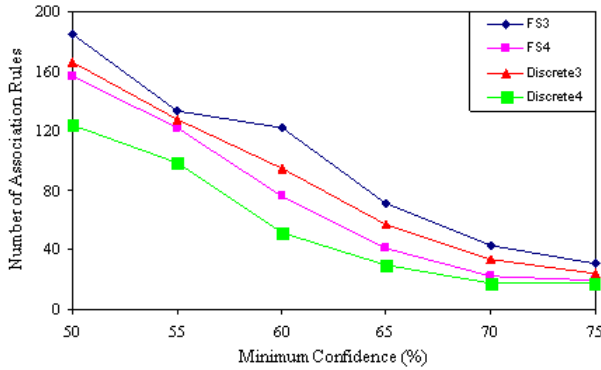


Fig. 2. Number of association rules vs minimum confidence

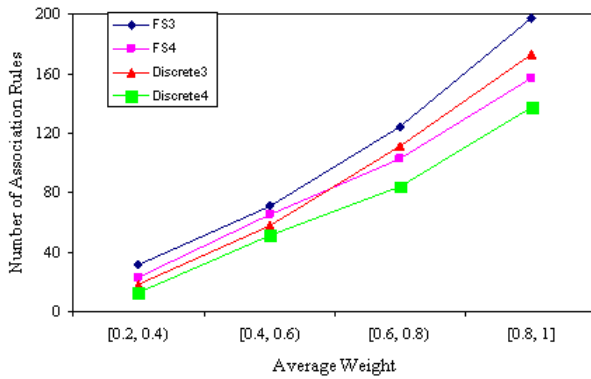


Fig. 3. Number of association rules vs random weight intervals

large itemsets increases along with the increase in the number of transactions. This is very reasonable because more transactions increase the probability of large itemsets for a certain minimum support threshold. When we reduced the value of the minimum support from $MinSup_1$ to $MinSup_2$ (12% for level 1, 4% for level 2 and 1.6% for level 3) we observed that the numbers of large itemsets highly decreased. However, the decrease in the fuzzy-based curves is less than the discrete method.

The numbers of association rules produced for different values of minimum confidence are reported in Figure 2. In this case, each itemset is coming from different dimensions. Figure 3 shows the number of association rules mined for different intervals of average weights. The minimum confidence is set as 50%. We used four intervals, from which random weights were generated. The increase in the number of rules is almost linear with respect to the average weight.

The results shown in Figure 4 demonstrate that the execution time decreases when the number of fuzzy sets increases. So, the execution time is inversely propositional to the minimum support value and the number of fuzzy sets.

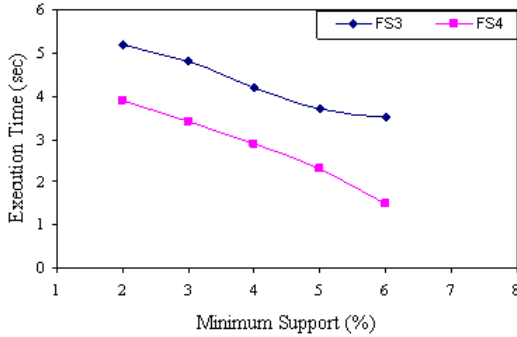


Fig. 4. Execution Time as minimum support changes

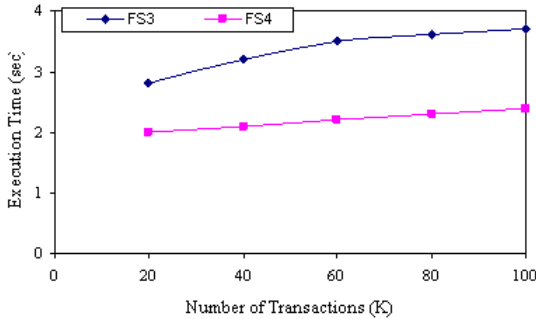


Fig. 5. Scalability

In the last experiment, we examine the scalability. For this purpose, we increase the number of transactions from 20K to 100K. The results reported in Figure 5 demonstrate that the proposed approach scales well. The mining time required for each of FS3 and FS4 scales quite linearly. However, the execution time of FS3 is larger than that of FS4; this is because the former finds larger number of association rules as shown in Figure 2.

5 Summary and Conclusions

OLAP is one of the most popular tools for on-line, fast and effective multidimensional data analysis. However, the research done so far on using OLAP techniques for data analysis have concentrated mainly on binary attributes, whereas, in general most databases that exist in real life include quantitative attributes. Moreover, the use of the fuzzy set theory in data mining systems when considering quantitative attributes leads to more generalized rules and enhances the understandability of the discovered knowledge. In order to tackle this bottleneck, we proposed in this paper a general architecture that utilizes a fuzzy data cube

for knowledge discovery that involves quantitative attributes. Also, we presented a method for the online mining of fuzzy association rules from the proposed architecture. The proposed method handle the mining process in multidimensional fuzzy data cube. Then, we integrated the multiple-level and weighting concepts with the proposed method. This leads to the fact that more interesting and more important rules can be extracted by the integrated method.

References

1. R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," *Proc. of ACM SIGMOD*, pp.1-12, 1996.
2. C.M. Kuok, A.W. Fu and M.H. Wong, "Mining fuzzy association rules in databases," *SIGMOD Record*, Vol.17, No.1, pp.41-46, 1998.
3. J. S. Park, M.S. Chen and P.S. Yu, "An effective hash-based algorithm for mining association rules," *Proc. of ACM SIGMOD*, pp.175-186, 1995.
4. R. Ng, L. V.S. Lakshmanan, J. Han and A. Pang, "Exploratory mining and pruning optimizations of constrained associations rules," *Proc. of ACM SIGMOD*, pp. 13-24, 1998.
5. C. Hidber, "Online Association Rule Mining," *Proc. of ACM SIGMOD*, pp. 145-156, 1999.
6. J. Han, "OLAP Mining: An Integration of OLAP with Data Mining," *Proc. of IFIP ICDS*, pp.1-11, 1997.
7. J. Han, "Towards on-line analytical mining in large databases," *Proc. of ACM SIGMOD*, 1998.
8. J. Han and Y. Fu, "Mining multiple-level association rules in large databases," *IEEE TKDE*, Vol.11, No.5, pp.798-804, 1999.
9. M. Kamber, J. Han and J.Y. Chiang, "Meta-rule guided mining of multidimensional association rules using data cubes," *Proc. of ACM KDD*, pp.207-210, 1997.
10. H. Lu, L. Feng and J. Han, "Beyond Intratransaction Association Analysis: Mining Multidimensional Intertransaction Association Rules", *ACM TOIS*, Vol.18, No.4, pp.423-454, 2000.
11. A. K. H. Tung, H. Lu, J. Han and L. Feng, "Efficient Mining of Intertransaction Association Rules", *IEEE TKDE*, Vol.15, No.1, pp. 43-56, 2003.
12. C.C. Agarwal and P.S. Yu, "A new approach to online generation of association rules," *IEEE TKDE*, Vol.13, No.4, pp.527-540, 2001.
13. C.H. Cai, et al, "Mining Association Rules with Weighted Items," *Proc. of IDEAS*, pp.68-77, 1998.
14. S. Yue, et al., "Mining fuzzy association rules with weighted items," *Proc. of IEEE SMC*, 2000.

Incremental Maintenance of Association Rules Based on Multiple Previously Mined Results

Zhuohua Duan^{1,2}, Zixing Cai², and Yan Lv²

¹ Department of Computer, School of Information Engineering, Shaoguan University, Shaoguan, Guangdong 512003, China
duanzhuohua@163.com

² School of Information Science and Engineering, Central South University, Changsha, Hunan, China
zxcai@csu.edu.cn

Abstract. Incrementally maintaining association rules based on two or more classes of frequent item sets may reduce the costs of scanning the original database remarkably. However, it was considered as a method of saving time with more storage spaces. It is suggested in this paper that all frequent item sets of several minimal supports can be stored in a table with a little additional storage, and a representation model is given. Based on this model, the paper systematically discusses the problem of incremental maintenance based on discovered association rules of several minimal supports. Theoretical analysis and experiments show that the approach makes full use of the previous results and reduces the complexity of incremental maintenance algorithms.

1 Introduction

Mining association rules is a very important issue in knowledge discovery in databases (KDD) proposed originally by Agrawal [1, 2], which mines potentially useful association rules from transaction database.

Apriori, the classical algorithm proposed by Agrawal, must scan the transaction database for many times. Due to the transaction database is usually very huge, D. W. Cheung et al. proposed the incrementally updating algorithm to reduce the times of scanning the database [3]. Hereafter, many researchers have studied and extended the incrementally updating algorithms [4-7]. It is widely accepted that incrementally updating based on multiple previously mined results can dramatically reduce the times of scanning the original transaction database. However, it is also regarded as a method of saving time by increasing storage spaces [7].

We suggest in the paper that, a little additional storage can store frequent item sets of several minimal supports, and present a representation model. Based on this, we systematically study the problem of incrementally updating association rules according to several previously mined results. Theoretical analysis and experimental results indicate that the method presented in this paper utility the previous results to a great extent.

2 Notations of Association Rules Mining

Let $I = \{ i_1, i_2, \dots, i_m \}$ denote the set of all of the items. An item set is a non-empty subset of I . DB is a transaction database of I . A transaction, T , is a subset of I . Association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$, and $X \cap Y = \Phi$.

Support number of item set X is the length of the set $\{ T \mid X \subseteq T, T \subseteq DB \}$, denoted by $X.count$. The support degree of item set X is denoted by $support(X) = X.count/|DB|$, where $|DB|$ denotes the number of the transactions in the database.

The support degree of item $X \cup Y$ is called as the support degree of association rule $X \Rightarrow Y$. The confidence degree of association rule $X \Rightarrow Y$ is $conf(X \Rightarrow Y) = support(X \cup Y) / support(X)$.

To mine the potentially useful association rules, two important thresholds are given, minimal support degree (denoted by s) and minimal confidence degree (denoted by c). X is called as frequent item set if and only if $support(X) \geq s$.

The key issue in association rules mining is to find all frequent item sets in the transaction database efficiently. Frequent item sets relate to transaction database (DB) and minimal support (s). In this paper, we use $L_{(s,DB)}$ to denote the set of all the frequent item sets of DB with the minimal support s , i.e. $L_{(s,DB)} = \{ X \mid X \subseteq I \text{ and } support(X) \geq s \}$. Let $X.count_{DB}$ be the support number of item set X in DB . A frequent item set with k items is called as frequent k -item.

Lemma 1. If $s < s'$, then $L_{(s',DB)} \subseteq L_{(s,DB)}$.

Proof: $\forall c \in L_{(s',DB)}, \exists c \subseteq I$ and $support(X) \geq s' > s$, i.e. $c \in L_{(s,DB)}$.

3 Representation Model for Frequent Item Sets

The goal is represent n classes of frequent item sets, i.e. $L_{(s_i,DB)}$ ($1 \leq i \leq n$) with minimal storage spaces.

Without loss of generality, let $s_i < s_{i+1}$. The following result is directly deduced according to lemma 1, i.e. $L_{(s_n,DB)} \subseteq L_{(s_{n-1},DB)} \subseteq \dots \subseteq L_{(s_1,DB)}$, as shown in Fig. 1.

$$\text{Let } C(s_i) = \begin{cases} L_{(s_i,DB)} - L_{(s_{i+1},DB)}, & 1 \leq i < n \\ L_{(s_i,DB)}, & i = n \end{cases}, \text{ then } L_{(s_i,DB)} = \bigcup_{j=i}^n C(s_j).$$

Obviously, $\{ C(s_j) \mid 1 \leq j \leq n \}$ is an equivalence partition of $L_{(s_1,DB)}$. Specifically, $\{ C(s_j) \mid 1 \leq j \leq n \}$ is an equivalence partition of $L_{(s_1,DB)}$, $C(s_j)$ ($1 \leq j \leq n$) is an equivalence class.

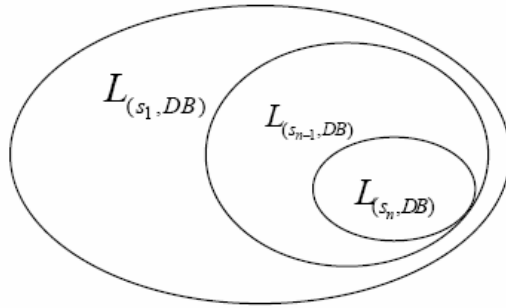


Fig. 1. Relations among several classes of frequent item sets

Given a frequent item set, X, let X.Stag denote the tag of X, indicating to which equivalence class does X belong.

$$X \in C(s_k) \Leftrightarrow k = X.Stag.$$

This can be implemented as follows. Let LargeItem be the table of all records in $L_{(s_1, DB)}$, and let ‘Stag’ be a column of LargeItem.

For example, the test data is the mushroom database provided by UCI (denoted by DB). There 8124 records, 23 kinds of attributes (items), and 2~12 values for each item. For clarity of description, the items (attributes) are denoted by I1, I2, ..., I23. All the frequent 4-item sets of DB with minimal support $s=0.7$ and $s=0.8$ can be represented in Table 1.

For convenience, all the minimal supports, $s_i (1 \leq i \leq n)$, are stored in table MS, MS has two columns, ‘Stag’ and ‘Support’, where ‘Stag’ denotes the index of minimal supports, and ‘Support’ denotes the corresponding minimal support, under the condition of $s_i < s_{i+1}$.

Obviously, the equivalence class $C(s_j)$ can be get using following SQL sentence:
 select * from LargeItem where Stag=j.

It is easy to get $L_{(s_j, DB)}$ via the following SQL statement:

select * from LargeItem where Stag>=j.

Table 1. Frequent 4-item sets of DB with minimal support $s=0.7$ and 0.8

No	I8	I9	I18	I19	I20	count	Stag
1	f	c	p	-	o	6272	1
2	f	c	-	w	o	6272	1
3	f	c	p	w	-	6602	2
4	-	c	p	w	o	6272	1
5	f	-	p	w	o	7288	2

4 Incrementally Updating Association Rules Based on Multiple Mining Transactions

Definition 1. The problem of incrementally updating association rules based on multiple mining transactions is, computing $L_{(s',DB')}$ based on $\{L_{(s_i,DB)} \mid 1 \leq i \leq n\}$, where $(s' \neq s_i)$ and/or $(DB' = DB - db + db')$, $db \subset DB$, $DB \cap db' = \Phi$.

This problem of incrementally updating can be divided into two classes according to whether the transaction database changes or not. The first class is that the transaction database does not change, and the minimal support changes; the second class is that transaction database changes.

4.1 Adjusting the Minimal Support

Given the transaction database DB does not change and the minimal support changes. The problem given by definition 1 is that, given $\{L_{(s_i,DB)} \mid 1 \leq i \leq n\}$, compute $L_{(s,DB)}$.

The problem can be divided into 3 cases:

Case 1: $s > s_n$.

Case 2: $s < s_1$.

Case 3: $s_1 < s < s_n$.

In Case 1, all the useful information are stored in $L_{(s_n,DB)}$, it need not access DB using the 'algorithm 1' proposed by Feng et al. [4]. To fit in with the representation model presented in this paper, we implemented the algorithm with SQL sentence as follows.

Algorithm 1. Incremental update association rules (Case 1), Input: $L_{(s_n,DB)}$, s ($s > s_n$), Output: $L_{(s,DB)}$

```

program Updating_Minimal_Support_Case1
begin
    insert into MS values(n+1,s);
    update LargeItem set Stag=n+1 where Stag=n and
count>=s*|DB|
end.

```

For Case 2, it is unavoidable to scan the transaction database DB. We employ the algorithm DCIUA [5] to deal with this case.

The algorithm 2 is based on DCIUA. It adds some steps to fit in with our representation model.

Algorithm 2. Incremental update association rules (Case 2), Input: $L_{(s_1, DB)}$, s ($s < s_1$), Output:

```

 $L_{(s, DB)}$ 
program Updating_Minimal_Support_Case2
begin
  Update MS set Stag=Stag+1;
  Insert into MS values(1, s);
  m = |  $L_{(s_1, DB)}$  |;
  update LargeItem set Stag=Stag+1;
  For each c computed by DCIUA do
  begin
    Insert into LargeItem values(m+1, c, count, 1);
    m=m+1;
  end
end.

```

For case 3, find j ($1 \leq j < n$), such that $s_j < s < s_{j+1}$. The general method is updating from $L_{(s_j, DB)}$ using ‘algorithm 1’. The time complexity is $O(|L_{(s_j, DB)}|)$. However, when $L_{(s_j, DB)}$ and $L_{(s_{j+1}, DB)}$ are known, for $L_{(s_{j+1}, DB)} \subseteq L_{(s, DB)} \subseteq L_{(s_j, DB)}$, so only the frequent item sets in $L_{(s_j, DB)} - L_{(s_{j+1}, DB)}$ (i.e. $C(s_j)$) have to be examined. This procedure is shown in Algorithm 3. The time complexity is $O(|C(s_j)|)$.

Algorithm 3. Incremental update association rules (Case 3), Input: $L_{(s_j, DB)}$, $L_{(s_{j+1}, DB)}$, s ($s_j < s < s_{j+1}$), Output: $L_{(s, DB)}$

```

program Updating_Minimal_Support_Case3
begin
  Update MS set Stag=Stag+1 where Stag>j;
  Insert into MS values(j+1, s);
  update LargeItem set Stag=j+1 where Stag=j and
  count>=s*|DB|; { Notice that this step scans the records
  in  $C(s_j)$  }
end.

```

4.2 Update the Transaction Database

Theorem 1 (the condition of accessing the original database)

If $s - s' \geq s \times |db| / |DB| + (1 - s) \times |db'| / |DB|$,

then $c \notin L_{(s', DB)} \Rightarrow c \notin L_{(s, DB - db + db')}$.

Proof:

$$\because c \notin L_{(s'', DB)}, \text{ so } c.count_{DB} < s'' \times |DB|, \text{ and } c.count_{db} \geq 0, c.count_{db'} \leq |db'|$$

$$\therefore c.count_{DB-db+db'} = c.count_{DB} - c.count_{db} + c.count_{db'} < s'' \times |DB| + |db'|, \text{ and } s - s'' \geq s \times |db| / |DB| + (1-s) \times |db'| / |DB|$$

$$\therefore c.count_{DB-db+db'} < s \times (|DB| - |db| + |db'|), \text{ i.e. } c \notin L_{(s, DB-db+db')}.$$

Corollary 1. If $s - s'' \geq s \times |db| / |DB| + (1-s) \times |db'| / |DB|$, incrementally updating $L_{(s'', DB)}$ to $L_{(s, DB-db+db')}$ needn't scan DB.

Algorithm 4. Incrementally update association rules (transaction database changes) Input: $L_{(s_i, DB)}$, db , db' , s_i' ; where $s_i - s_i' \geq s_i \times |db| / |DB| + (1-s_i) \times |db'| / |DB|$ ($1 \leq i \leq n$), Output: $L_{(s_i', DB-db+db')}$

```

program BatchUpdate
begin
  for (i=n; i>=1; i--)
  begin
     $s_i' = (s_i \times |DB| + |db'|) / (|DB| - |db| + |db'|)$ ;
    Update MS set Support =  $s_i'$  where Stag=i;
    for each  $c \in C(s_i)$ 
    begin
      new_count =  $c.count_{DB} - c.count_{db} + c.count_{db'}$ ;
      if (new_count  $\geq s_i' \times (|DB| - |db| + |db'|)$ ) then
        c.count = new_count else
          if (i>1) then  $C(s_{i-1}) = C(s_{i-1}) \cup \{c\}$ 
            else delete c from  $C(s_i)$ 
      end{for}
    end{for}
  end.

```

5 Space and Time Efficiency Analysis

5.1 Space Efficiency Analysis

Let m_i denotes the number of frequent item sets in $L_{(s_i, DB)}$, i.e. $m_i = |L_{(s_i, DB)}|$.

In the representation model presented in section 3, n classes of frequent item sets, $L_{(s_i, DB)}$ ($1 \leq i \leq n$), are represented using only m_1 records. Suppose that t bytes are allocated for the storage of each frequent item set. The additional storage spaces contain two components: (1) 4 bytes for each record of $L_{(s_i, DB)}$, i.e. $4 m_1$, (2) $12 \cdot n$ bytes

for table MS. The total additional storage space is $(4 m_1 + 12n)$ bytes. The total storage space is $(t * m_1 + 4 m_1 + 12n)$ bytes. For $12n \ll t * m_1 + 4 m_1$, the storage space does almost not increase with n .

On the contrary, the general approaches cost about $\sum_{i=1}^n (m_i \times t)$ bytes of storage spaces.

For example, given the minimal supports $\{0.30, 0.32, 0.34, 0.36, 0.38, 0.40\}$, $L_{(0.3, DB)}$ is computed with Apriori algorithm, and $L_{(0.32, DB)}$, $L_{(0.34, DB)}$, $L_{(0.36, DB)}$, $L_{(0.38, DB)}$, $L_{(0.4, DB)}$ are incrementally computed with algorithm presented in this paper.

The storage of our representation model and the general model for several cases are compared in Table 2. In table 2, $C4 = \sum_{i=1}^n (m_i \times t)$ denotes storage expenditure of general method, and $C5 = t * m_1 + 4 m_1 + 12 * n$ denotes storage expenditure of our method. It shows that C4 increases quickly and C5 increases very slowly.

Table 2. Storage space efficiency analysis, $C6 = 4 m_1 + 12 * n$, $t = 31$

Minimal supports	$\sum_{i=1}^n (m_i)$	m_1	C4	C5	C6
{1,2}	4686	2735	145266	95749	10964
{1,2,3}	6081	2735	188511	95761	10976
{1,2,3,4}	7168	2735	222208	95773	10988
{1,2,3,4,5}	7955	2735	246605	95785	11000
{1,2,3,4,5,6}	8520	2735	264120	95797	11012

5.2 Time Efficiency Analysis

For Case 3, the time complexity of algorithm 3 is $O(|C(s_j)|)$. Conversely, the previous method, i.e. ‘algorithm 1’ in reference [4], costs about $O(|L_{(s_j, DB)}|)$. The results are shown in Table 3.

Table 3. Time complexities of algorithm3 and ‘algorithm 1’ (in reference [4])

j	1	2	3	4	5	6
s_j	0.3	0.32	0.34	0.36	0.38	0.4
$ L_{(s_j, DB)} $	2735	1951	1395	1087	787	565
$ C(s_j) $	784	556	308	300	222	565

The time complexity of algorithm BatchUpdate is $O(|L_{(s_j, DB)}| \times (|db| + |db'|))$. The results of BatchUpdate are shown in Table 4, in which the database DB is updated to DB-db (db denotes the data set of the last 160 records of DB). It shows that most frequent sets of $L_{(s_j, DB)}$ are updated to $L_{(s_j', DB-db)}$.

Table 4. Experiment results of BatchUpdate

j	1	2	3	4	5	6
s_j	0.3	0.32	0.34	0.36	0.38	0.4
$ L_{(s_j, DB)} $	2735	1951	1395	1087	787	565
s_j'	0.3061	0.3265	0.3469	0.3673	0.3877	0.4081
$ L_{(s_j', DB-db)} $	2727	1943	1389	1081	781	561

6 Conclusion

In this paper, an efficient representation model for frequent item sets of several minimal supports was presented. It needs only a little additional storage spaces.

Based on this, it studied the problem of updating association rules based on several previously mined results, and presented two algorithms: Algorithm3 and BatchUpdate.

Theoretical and experimental results show that our method decreases the scan times of original database with the costs of a little additional storage spaces.

References

1. Agrawal R., Imielinski T., Swami A.: Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data:SIGMOD'93, New York:ACM Press, 1993, pp. 207-216.
2. Agrawal R., Srikant R.: Fast Algorithms for Mining Association Rules. In:Proceedings of the ACM SIGMOD International Conference, 1994, pp. 487-499
3. Cheung D. W., Han J., Ng V. T., Wong C. Y.: Maintenance of discovered association rules in large databases: an incremental updating technique. In Proceedings of the 12th International Conference on Data Engineering, New Orleans, Louisiana, 1996, pp. 106-114
4. Feng Y. C., Feng J. L.: Incremental updating algorithm for association rules. Journal of Software, 1998, vol. 4, pp. 301-306 (in chinese)
5. Lin Z. M.: Incremental updating technique of association rules based on divide and conquer strategy. Pattern recognition and artificial intelligence, 2002, vol. 1, pp.103-107(in chinese)
6. Chen L., Chen G. C.: An improved incremental algorithm for maintaining discovered association rules. Computer Engineering and Design, 2002, vol. 1, pp. 60-63 (in chinese)
7. Ouyang W. M., Zheng C., Cai Q. S.: A time windowing technique for the incremental maintenance of association rules. Mini Micro System, 2001, vol. 1, pp. 55-58(in chinese)

Mining and Validation of Localized Frequent Web Access Patterns with Dynamic Tolerance

Olfa Nasraoui¹ and Suchandra Goswami²

¹ Dept. of Computer Engineering & Computer Science, University of Louisville
Louisville, KY 40292

olfa.nasraoui@louisville.edu
<http://www.louisville.edu/~o0nasr01/>

² Dept. of Electrical and Computer Engineering, The University of Memphis
Memphis, TN 38152-318
sgoswami@memphis.edu

Abstract. Mining user profiles is a crucial task for Web usage mining, and can be accomplished by mining frequent patterns. However, in the Web usage domain, sessions tend to be very sparse, and mining the right user profiles tends to be difficult. Either too few or too many profiles tend to be mined, partly because of problems in fixing support thresholds and intolerant matching. Also, in the Web usage mining domain, there is often a need for post-processing and validation of the results of mining. In this paper, we use criterion guided optimization to mine localized and error-tolerant transaction patterns, instead of using exact counting based method, and explore the effect of different post-processing options on their quality. Experiments with real Web transaction data are presented.

1 Introduction

Association rule mining [2] is a key data mining task that is most widely applied to problems in market basket analysis, where one desires to find baskets of items frequently purchased *together* in a Transactional Database (TDB). More recent applications include text mining [9], scientific applications [6], and bioinformatics [5]. Mining *frequent itemsets* or *Frequent Pattern (FP)* is a crucial prerequisite step to mining association rules that faces several previously acknowledged challenges, such as mining and maintaining association rules in very large databases [3,10,13] and in evolving databases [4,11]. In this paper we will focus our attention on three *other* issues that directly address the FP definition, in particular in the context of TDBs with a *large number of items/dimensionality*, *sparse* data, and *heterogenous* data distributions. Such databases are very common in the context of *e-commerce transactions* on large e-commerce sites that offer a huge number of products. Such TDBs are also very common in the context of mining of *web user clickstream data*, where user sessions are the transactions and URLs are the items. Other kinds of data that satisfy these characteristics occur in the context of mining large collections of *text documents*, where the documents play the role of transactions and keywords play the role of items. *Sparse* data sets suffer from the fact that for a large number of items, the number of non-zero entries is a very small fraction of the total number of entries in the transaction matrix.

For the majority of Web usage sessions or Web transactions data, frequent itemset definition and mining have suffered from the following problems:

(i) Sensitivity to Support Thresholds: Very low support thresholds typically lead to generating too many spurious patterns (that are due to random correlations in data), while high support thresholds risk missing many interesting patterns that occur with low support, but have high confidence. This problem most particularly affects heterogeneous data sets, where certain itemsets may occur on only part of the data (e.g. in only some segments of a customer database depending on the geographical location), and hence will have low support. One of the first researchers who have addressed this issue are Pei, Tung, and Han [8] who defined the notion of *Fault-Tolerant Frequent Patterns (FTFP)*. Unlike frequent itemsets, FTFPs allow a fault tolerance equal to δ , meaning that *up to δ mismatches* in the items are allowed. So instead of finding exact patterns in data the search is for approximate and more general fault-tolerant patterns. Unfortunately, this approach actually requires *two* instead of *one* support threshold: one for the items, and another one for the itemsets. Moreover, an *additional* threshold is required for the amount of tolerance, δ .

(ii) Error intolerance: When counting the support of an itemset, only transactions that completely include *all* the items of an itemset are counted. If a transaction matches an itemset in say 10 items, but fails to match *one* item, then it is completely excluded from the support count. In other words, this is an *all or nothing* counting. Some work has addressed this issue, including Fault-Tolerant Frequent Pattern (FTFP) [8], and Error-tolerant Itemsets (ETI) [12] of two types (*strong* and *weak*). The problem with this approach is that getting away from having to pre-specify support thresholds led to getting trapped in another requirement: having to specify tolerance thresholds.

(iii) Absence of locality in frequent itemset counting and validation: Data may be skewed, heterogeneous, or better modeled by several subsets, each with its own frequent itemsets, possibly with *different support* thresholds and even different levels of *tolerance*. Researchers who have addressed this issue include Aggarwal, Procopiuc, and Yu [1] who have proposed CLASD (CLustering for ASSociation Discovery) that discovers frequent patterns called *metatransactions* by aggregating the item frequencies of transactions assigned to each clusters based on maximum similarity into a frequency vector. The clusters are found using a Hierarchical Agglomerative Clustering that starts with randomly selected transactions as the initial seeds. While this approach proposed some improvements mainly in localization of the search, it still needs to address several problems: (i) sensitivity to prespecified number of clusters (k) and minimum size threshold of a cluster that implicitly plays the role of metatransaction support, (ii) also, since transactions are assigned based only on similarity, transactions that are not very similar to any cluster will still get lumped to the closest cluster, and hence contribute to its support. In other words, the notion of *cluster size* which is equivalent to *metatransaction support* does not take into account the *level* of similarity of the transaction. This corresponds to the *opposite* extreme end of exact/intolerant support counting because even a transaction that does not match any of the items in the candidate pattern will be counted in the support.

(iv) Post-processing and validation: In the Web usage mining domain, there is often a need for post-processing and validation of the results of frequent pattern mining, be-

cause these patterns are to be used downstream for specialized applications such as personalization. Thus, it is interesting to study how post-processing affects the results.

In this paper, we apply an FP mining approach that we have proposed in [14] to mining *local, error-tolerant* frequent patterns (profiles) from Web transaction data, and explore the effect of various post-processing options. We call the special kind of patterns that we mine: *Localized Error Tolerant Frequent Pattern (LET-FP)*. Unlike the preliminary work in [14], in this paper, we will also explore several post-processing options of the mined frequent patterns, and extended our proposed information retrieval inspired validation procedure that simultaneously penalizes against (i) an excess of spurious patterns, and (ii) a lack of accurate patterns, by calculating measures that attempt to answer the following crucial questions: (1) Are all the mined patterns *needed* to summarize the data? (2) Is the data set well summarized/represented by the mined patterns?

2 A Generalized Frameworks for Localized Error-Tolerant Frequent Pattern (LET-FP) Mining

2.1 Frequent Itemsets: A Similarity Based Perspective

Frequent itemsets or patterns (FP) can be considered as one way to form a *summary* of the input data. As a summary, frequent patterns represent a reduced form of the data that is at the same time, *as close as possible* to the original input data. This is compatible with the notion of support as a critical measure of goodness for a FP. *Classical support* measures the *count* of the transactions that *completely include* a FP. Therefore transactions that are very similar to a FP, but perhaps lacking a single item from the FP do not even count in its support. The first step toward including *tolerance* is to allow transactions that are very similar to a FP to count in what we refer to as *partial* support. For this reason, we need to consider using a *similarity* measure to capture *closeness* between a FP and a transaction. We first explain the notation that will be used throughout the rest of this section. Hence, P_i denotes the i^{th} frequent pattern. $|P_i|$ is the number of items in P_i . t_j denotes the j^{th} transaction, $|t_j|$ is the number of items in t_j , and S_{ij} is the Similarity between the i^{th} FP and the j^{th} transaction. An FP should represent a *frequently* occurring trend. Hence it should be as *similar* as possible to *as many* transactions as possible. Hence, we need to assess the similarity between a FP, P_i , and each transaction t_j . For example it can be shown that the classical itemset definition of *APriori* uses a complete FP inclusion based similarity [14], where the similarity is nonzero (and has value 1) only if the pattern is completely included in a transaction. Other possibilities (that are investigated in this paper) include the *cosine* similarity, *precision*, *coverage*, as well as the *minimum of precision and coverage (MinPC)* measures of a candidate pattern, using the transaction as ground-truth reference [14].

2.2 Error Tolerant Support

Let a candidate *Localized Error Tolerant Frequent Pattern*, henceforth referred to as *LET-FP*, be denoted as P_i , and let the transactions in a DB be denoted by t_j . Instead of *APriori's* complete pattern inclusion based matching, we propose to use a generalized,

softer matching measure. This matching measure $Sim(P_i, t_j)$ quantifies how faithfully the frequent pattern P_i serves as a summary for transaction t_j . A dissimilarity $d(P_i, t_j)$ can be defined so that it is inversely related to $Sim(P_i, t_j)$, for example,

$$d(P_i, t_j) = (1 - Sim(P_i, t_j))^2 \quad (9)$$

Let the amount of tolerance ε be dynamic and defined in the same units as $d(P_i, t_j)$. Furthermore, let the tolerance be *localized*, and hence depend on the ETFP itself, i.e

$$\varepsilon_i = \varepsilon(P_i).$$

Next, let a *tolerance-normalized* dissimilarity between LET-FP P_i and transaction t_j be defined as

$$d_\varepsilon(P_i, t_j, \varepsilon_i) = \frac{d(P_i, t_j)}{\varepsilon_i} \quad (10)$$

A lower tolerance will tend to inflate the effect of dissimilarity, hence reflecting a more stringent matching process. Now that the tolerance degree, ε_i has been “absorbed” into the normalized dissimilarity $d_\varepsilon(P_i, t_j, \varepsilon_i)$, a measure of *support* that is *error-tolerant* can be defined. Let this localized error-tolerant support be defined as

$$s(P_i, t_j, \varepsilon_i) = f(d_\varepsilon(P_i, t_j, \varepsilon_i)) = f(d(P_i, t_j), \varepsilon_i),$$

where $f: \mathcal{R} \rightarrow [0,1]$ is a monotonically non-increasing function. The *Total Localized Error-Tolerant support* of LET-FP P_i may be defined by summing the contributions from all transactions as follows

$$Ts(P_i, \varepsilon_i) = \sum_{t_j \in T} s(P_i, t_j, \varepsilon_i) \quad (11)$$

Note that the total support in (11) increases monotonically with the tolerance ε_i . Because tolerance is not known in advance, this will favor higher tolerance values. To put a limit on this bias, we define a “*normalized support*” instead of an absolute support, to be used as an FP goodness criterion, i.e,

$$\rho(P_i, \varepsilon_i) = \frac{Ts(P_i, \varepsilon_i)}{\varepsilon_i} \quad (12)$$

In this equation the tolerance degree can also be considered as a penalty factor P_i . Given the same support, LET-FPs will be rewarded if their tolerance degree is smaller and penalized if their tolerance-degree is higher.

2.3 Avoiding Fixed Support Thresholds: Mining Frequent Patterns by Support Optimization

Instead of searching for the FPs that exceeds a *fixed* support threshold, we propose to seek the FPs that *maximize the error tolerant support* in (12). The FP mining and tolerance search problem can be stated as an *alternating optimization problem* that boils down to two optimization steps, to determine the frequent *patterns* and the error *tolerance* respectively that optimize the error tolerant support: that is (1) Fix ε_i , and solve for $P_i = Arg Max (\rho(P_i, \varepsilon_i))$, and (2) Fix P_i , and solve for $\varepsilon_i = Arg Max (\rho(P_i, \varepsilon_i))$.

Step 2 can be solved by *analytical* optimization if $\rho(P_i, \epsilon_i)$ is *differentiable* with respect to ϵ_i , and a closed *Piccard update equation* for ϵ_i can be derived as shown in [14]. The normalized error-tolerant support in (12) satisfies several desiderata.

- **Localized Support:** Support is defined on increasingly smaller subsets/clusters of the data, providing a *localized* and confined counting.

- **Error-tolerance:** Data tuples that deviate slightly from candidate LET-FP will still contribute to its support, though to a lesser degree, depending on the tolerance amount.

- **Dynamic Tolerance:** Given the local support measure function $s(P_i, t_j, \epsilon_i) = f(d_\epsilon(P_i, t_j, \epsilon_i)) = e^{-d_\epsilon(P_i, t_j, \epsilon_i)}$, we can analytically derive an iterative update equation [14] for dynamic tolerance level ϵ_i based on optimizing the total error-tolerant support given by (12). However $\rho(P_i, \epsilon_i)$ is not in a form that is differentiable with respect to P_i . Therefore, a *non-analytical* optimization approach is needed for Step 1. We use a Genetic algorithm for this purpose, but do not rule out other heuristic optimization methods. This leads to an *alternating optimization* approach, i.e. alternate solving for one of the parameters, while the rest are fixed, and it is common in the optimization and machine learning literature, including the *Expectation Maximization (EM)* algorithm, and *Maximum Likelihood Estimation* methods.

2.4 Localized ETFP Mining by Partitioning and Zooming

Some strong (i.e. highly accurate/confident) associations may lurk in small segments of a huge data set. In this case, they risk being missed because of their low support. In other words, finding frequent itemsets from the entire aggregate data may not be able to reveal patterns that are only valid in *small localized* segments of the data. Data *locality* concepts can offer several advantages in this context. We achieve locality by gradually focusing the search on smaller and smaller segments of the data set. A greedy procedure extracts the unique optimal patterns discovered at each iteration. Redundant patterns are identified based on their compatibility with a previously extracted pattern, and are therefore ignored.

Based on these extracted FPs, the dataset is gradually divided into smaller parts/clusters containing similar transactions. The steps needed to obtain localized LET-FPs may be summarized as follows:

Algorithm LET-FP Mining:

0. Let *current* transaction dataset (D_c) = D (input data), and let the set of final extracted LET-FPs, $P = \emptyset$
1. Initial FP-Generation: Seed the FPs by selecting random samples from *current* transaction dataset (D_c).
2. Iterative LET-FP search and extraction: will result in C LET-FPs P_1, \dots, P_c , and tolerance values $\epsilon_1, \dots, \epsilon_c$
3. Partition transactions by assigning each transaction to nearest LET-FP (based on chosen similarity measure). This will partition the dataset into C subsets T_1, \dots, T_c .
4. For $i = 1, \dots, C$ { // **Step 4 is for zooming (optional)**
 Let $T_i^{\text{out-of-core}} = \{t_j \mid s(P_i, t_j, \epsilon_i) < s_{\text{core}}\}$
 Let $T_i = T_i - T_i^{\text{out-of-core}}$

```

    Let  $T_{zoom} = \bigcup_i T_i^{out-of-core}$ 
  }
5. For each subset  $T_i$  {
  If  $\epsilon_i > \epsilon_{max}$  and  $|T_i| > t_{max}$  Then {
    Let current transaction dataset  $D_c = T_i$ .
    Go to step 1, // repeat search on each cluster
  }
  Else  $P = P \cup P_i$  // Add to final list of LET-FPs
}
6. Let current dataset  $D_c = T_{zoom}$ . Go to Step 1. // Step 6 is for zooming
(optional)

```

Step 2 can be any competent search method, preferably, one that is global, and that can benefit from randomized search to sample the huge search space of all possible LET-FPs, such as a genetic algorithm.

2.5 Validation in an Information Retrieval Context

Frequent itemsets or patterns (FP) can be considered as one way to form a summary of the input data. As a summary, frequent patterns represent a *reduced* form of the data that is at the same time, *as close as possible* to the original input data. This description is reminiscent of an *information retrieval* scenario, in the sense that patterns that are retrieved should be as *close* as possible to the original transaction data. Closeness should take into account both (i) *precision* (a summary FP's items are all correct or included in the original input data, i.e. they include *only* the true data items) and (ii) *coverage/recall* (a summary FP's items are complete compared to the data that is summarized, i.e. they include *all* the data items). We propose a validation procedure that is inspired by information retrieval that simultaneously *penalizes* (i) *an excess of spurious patterns*, and (ii) *a lack of accurate patterns*, by calculating measures that attempt to answer the following crucial questions: **(1)** Are *all* the mined patterns *needed* to (a) *completely* and (b) *faithfully* summarize the data? **(2)** Is the data set (a) *completely* and (b) *faithfully* summarized/represented by the mined patterns? Each of these two questions is answered by computing *coverage/recall* as an *Interestingness measure* to answer part (a), and *precision* as an *Interestingness measure* to answer part (b), while reversing the roles played by the mined patterns and the data transactions, respectively.

First, we compute the following *Interestingness* measures for each LET-FP, letting the *Interestingness measure*, $Int_{ij} = Cov_{ij}$ (i.e., *coverage*: See Eqs. In Sec 3.1.) to answer part (a), and $Int_{ij} = Prec_{ij}$ (i.e., *precision*: See Eqs. In Sec 3.1.) to answer part (b).

Let the set of transactions satisfying interestingness measure Int_{ij} for the i^{th} LET-FP, be

$$T_i^{\text{int}} = \{t_j \mid Int_{ij} > Int_{min}\}.$$

Then the following measure gives the proportion of transactions that are well summarized by the i^{th} LET-FP

$$Int_i^1 = |T_i^{\text{int}}| / |T|$$

The average Interestingness over the entire set P of patterns is given by

$$Int^1 = \sum_i Int_i^1 / |P| \tag{16}$$

The measure in (16) is penalized if there are *too many patterns* that do not satisfy the interestingness criterion *for many transactions*. Hence, this is a very severe validation measure. When $Int_{ij} = Cov_{ij}$, we call Int^1 the *Normalized Count of Coverage*, and it answers Question 1.a. When $Int_{ij} = Prec_{ij}$, we call Int^1 the *Normalized Count of Precision*, and it answers Question 1.b.

Now, if we let $T^* = \{t_j \mid \text{Max}_i (Int_{ij}) > Int_{min}\}$. Then

$$Int^2 = |T^*| / |T| \tag{17}$$

When $Int_{ij} = Cov_{ij}$, we call Int^2 the *Cumulative Coverage of Transactions*, and it answers Question 2.a. When $Int_{ij} = Prec_{ij}$, we call Int^2 the *Cumulative Precision of Transactions*, and it answers Question 2.b.

The measures answering questions 2 quantify the quality of mined patterns from the point of view of providing an accurate summary of the input data. While the measures answering questions 1 quantify the necessity of any pattern at all, hence penalizing against patterns that are spurious, or for which there is no counterpart in the input data. The above measures are computed over the entire range of the Interestingness threshold Int_{min} from 0% to 100% in increments of 10%, and plotted.

2.6 LET-FP Search and Post-processing Options

After the completion of the LET-FP search algorithm, we partition the input transactions into as many clusters as the number, say $|P|$, of the *original* (i.e., *without post-processing*) LET-FPs, $P = \{P_1, \dots, P_{|P|}\}$. Let these transaction clusters be denoted as $T_1, \dots, T_{|P|}$. Then, there are several ways that we may use the LET-FPs, as listed below.

Search Options: First the search for LET-FPs can either use *zooming* or not.

-
- (1) **Standard LET-FPs:** obtained by eliminating steps 4 and 6 in Algorithm LET-FP Mining (see Sec 3.4) and doing *no* post-processing
 - (2) **Zoomed LET-FPs:** We use steps 4 and 6 in Algorithm LET-FP Mining (see Sec 3.4) to gradually zoom into each transaction cluster by peeling off the out-of-core transactions.
-

Post-Processing Options: After completing the search for LET-FPs, there are several options:

-
- (1) **Original LET-FPs:** These are the LET-FPs obtained *without* post-processing
 - (2) **Aggregate LET-FPs:** Frequency vectors computed by averaging the item occurrence frequencies in each cluster separately, then converting to a binary vector (1 if frequency > 0.10, 0 otherwise).
 - (3) **Robustified LET-FPs:** Before aggregating the LET-FP frequencies as in the previous option, we zoom into each cluster, and remove the out-of-core transactions, i.e. $T_i = T_i - \{t_j \mid s(P_i, t_j, \epsilon_i) < s_{core}\}$.
-

Other options are produced by different combinations of search and post-processing options. Table 1 lists the different codes used in our experimental section that designate these different options

Table 1. Category codes corresponding to LET-FP search and post-processing options

Code	search	post-processing
spa	Standard (i.e., no zooming)	post-processing: aggregate
spr	Standard (i.e., no zooming)	post-processing: robustified
so	Standard (i.e., no zooming)	Original (no post-processing)
zpa	Zoomed (w/ steps 4 & 6 of LET-FP Mining Algorithm)	post-processing: aggregate
zpr	Zoomed (w/ steps 4 & 6 of LET-FP Mining Algorithm)	post-processing: robustified
zo	Zoomed (w/ steps 4 & 6 of LET-FP Mining Algorithm)	Original (no post-processing)

3 Experimental Results

Experimental results are obtained by using the LET-FP Mining Algorithm described in Sec. 3.4, and we compare against the performance of *APriori* [2] with varying minimum support levels. The proposed LET-FP Mining algorithm is validated using the different *search* strategies (with or without zooming) and different *similarity* measures. Hence, we validate all the possible combinations listed in Sec. 3.7, as well as the different similarity measure options by computing and plotting the interestingness measures described in Sec. 3.6. To avoid information overload, for each <search & post-processing> category, we report the results *only for the best performing similarity measure*. Also because of the definition of frequent itemsets in *APriori*, precision is always equal to 1. Hence we do not plot it for different minimum interestingness thresholds, but rather list it in tabular format, considering it as threshold of 0.9. To optimize step 1, we use a randomization based optimization method (GA) with population size 100, 30 generations, and binary encoding of transactions. The crossover and mutation rates were 0.9 and 0.001 respectively. The dataset consists of the preprocessed web-log data of a Computer science department website. This dataset has 343 distinct URLs accessed by users. A session was defined as a sequence of URL requests from the same IP address within a prespecified time threshold of 45 minutes [7], leading to 1704 real user sessions. On this data set, *APriori* [2] generated a large number of itemsets, despite the conservative minimum support thresholds, as shown in Table 2, while the proposed LET-FP mining algorithm produced a much smaller number of frequent patterns as shown in Table 3 for the different search strategies (with or without zooming) and for different similarity measures (cosine, precision, coverage, MinPC).

Table 2. Variation in number of *APriori* itemsets with respect to support threshold (Web transaction data)

Support	1-itemsets	2-itemsets	3-itemsets	4-itemsets	5-itemsets	Total
1%	129	159	160	81	2	531
2%	87	40	26	18	-	171
5%	27	12	6	-	-	45

Table 3. Variation in number of LET-FPs for different similarity measures (web transaction data)

	Cosine	Coverage	Precision	MinPC
standard	36	19	31	38
zoomed	32	22	26	30

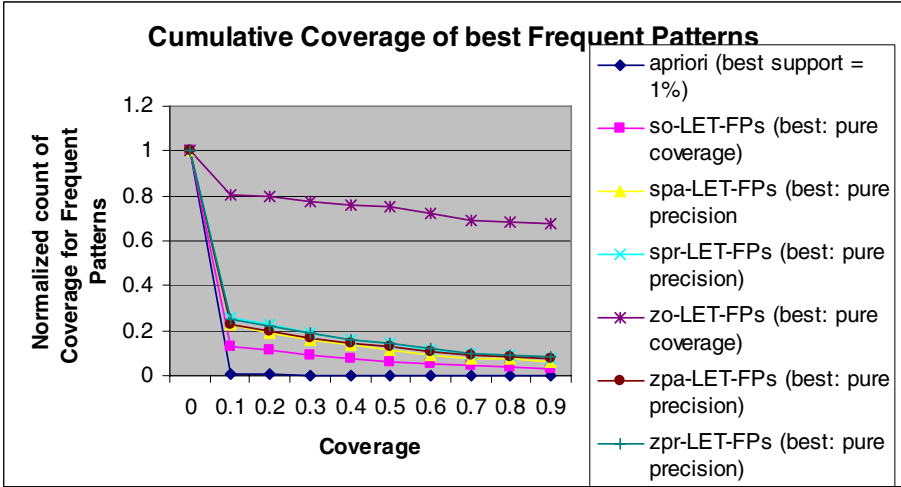


Fig. 1. Normalized-count of Coverage for best Frequent Patterns

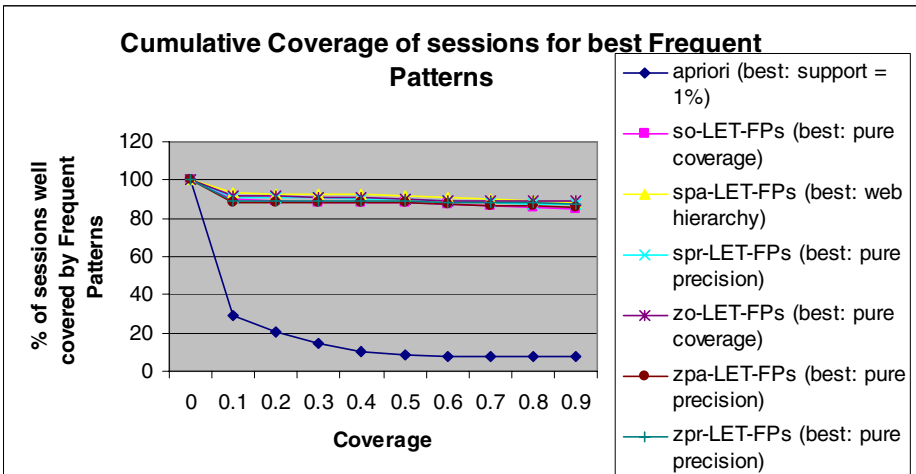


Fig. 2. Cumulative Coverage of sessions for best performing similarity in each category

Figure 1 shows (for all quality thresholds starting from 0 until 0.9) the normalized-count of coverage for FPs using the best similarity measure for LET-FPs and the best support percentage for *Apriori*. We can see that the LET-FPs perform better than the Frequent Patterns obtained by *Apriori*. Figure 2 shows that the percentage of sessions well covered is higher for LET-FPs than with *Apriori*.

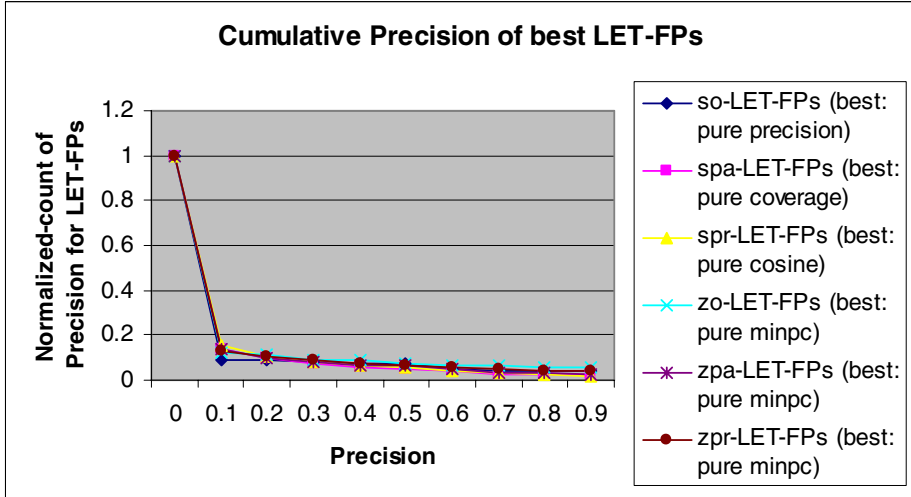


Fig. 3. Normalized-count of Precision for best performing similarity in each category

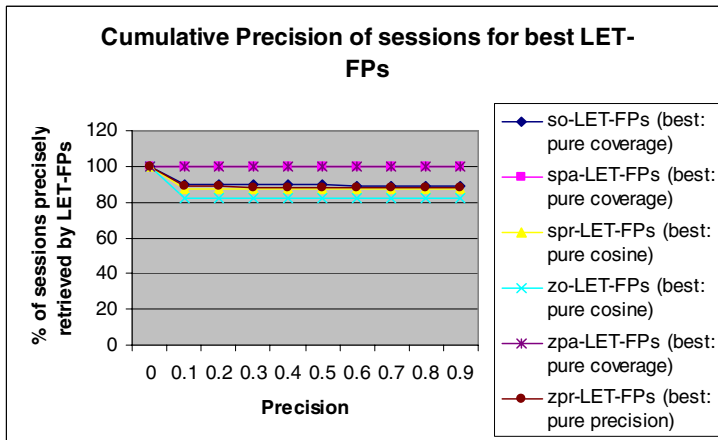


Fig. 4. Cumulative Precision of sessions for best performing similarity in each category

Figure 3 shows that *spr-LET-FPs* (i.e. *standard search, post-processed with robustification*) using pure cosine similarity give the best normalized-count of precision. Figure 4 shows that the *zpa-LET-FPs* using coverage similarity measure give the best

percentage of sessions precisely retrieved by FPs. To summarize all the results, we note that both normalized-counts of coverage and precision of FPs obtained by *Apriori* are less than those of LET-FPs. At the same time, the % of sessions/transactions well covered by any of the FPs obtained by *Apriori* is less than the % of sessions/transactions well covered by any of the LET-FPs obtained by our proposed approach, while the % of sessions/transactions precisely retrieved by any of the FPs obtained by *Apriori* is less than or equal to the one retrieved by any of the LET-FPs obtained by our approach. Furthermore, the LET-FPs obtained by our proposed method come with no pre-specified, fixed support, and require roughly half the time of *APriori*. *Zooming* results in increasing coverage because it can dig out *small localized LET-FPs*. Yet *precision is not significantly affected*. Finally, *Robustified* LET-FPs have the best coverage performance, followed by aggregated, and finally by the raw (not post-processed) LET-FPs.

4 Conclusions

In this paper, we applied a novel FP mining approach to mining *local, error-tolerant* frequent patterns (profiles) from Web transaction data. We also explored zooming as an efficient search strategy, studied several post-processing options of the mined frequent patterns, and investigated an information retrieval inspired validation procedure. Other important issues such as scalability will be addressed in the future.

Acknowledgment

This work is supported by National Science Foundation CAREER Award IIS-0133948 to O. Nasraoui.

References

1. C. Aggarwal, C. Procopiuc, and P. Yu. Finding Localized associations in market basket data. *IEEE Trans. Knowledge and Data Engineering*, Vol 14, No. 1, Jan 2002.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th Int'l Conf. on Very Large Databases*, SanTiago, Chile, June 1994.
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 Intl. Conf. on Data Engineering*, February 1996.
4. V. Ganti, J. Gehrke, and R. Ramakrishnan. Demon: Mining and monitoring evolving data. In *Proc. of the 16th Intl. Conf. on Data Engineering*, pp. 439–448, May 2000.
5. J. Han, H. Jamil, Y. Lu, L. Chen, Y. Liao, and J. Pei. Dna-miner: A system prototype for mining dna sequences. In *Proc. of the 2001 ACM-SIGMOD Int'l. Conf. on Management of Data*, Santa Barbara, CA, May 2001.
6. C. Kamath. On mining scientific datasets. In et al R. L. Grossman, editor, *Data Mining for Scientific and Engineering Applications*, pages 1–21. Kluwer Academic Publishers, 2001.
7. O. Nasraoui and R. Krishnapuram, and A. Joshi. Mining Web Access Logs Using a Relational Clustering Algorithm Based on a Robust Estimator, 8th International World Wide Web Conference, Toronto, pp. 40-41, 1999.

8. J. Pei, A.K.H. Tung, and J. Han, Fault tolerant frequent pattern mining: Problems and challenges, Proc. 2001 ACM-SIGMOD Int. Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'01), Santa Barbara, CA, May 2001.
9. M. Rajman and R. Besan. Text mining - knowledge extraction from unstructured textual data. In *Proc. of the Int'l Conf. Federation of Classification Societies*, pages 473–480, Roma, Italy, 1998.
10. S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka. An efficient algorithm for the incremental updation of association rules. In *Proc. of the 3__ Int'l Conf. on Knowledge Discovery and Data Mining*, August 1997.
11. A. Veloso, W. Meira Jr., M. B. de Carvalho, B. Possas, S. Parthasarathy, and M. Zaki. Mining frequent itemsets in evolving databases. In *Proc. of the 2__ SIAM Int'l Conf. on Data Mining*, Arlington, USA, May 2002.
12. C. Yang, U. Fayyad, and P. Bradley, Efficient Discovery of error-tolerant frequent itemsets in high dimensions, In Proc. of seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 194-203, San Francisco, California, Aug. 2001.
13. M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New parallel algorithms for fast discovery of association rules. *Data Mining and Knowledge Discovery: An International Journal*, 4(1):343–373, December 1997.
14. O. Nasraoui and S. Goswami, Mining and Validating Localized Frequent Itemsets with Dynamic Tolerance, in Proceedings of SIAM conference on Data Mining (SDM 2006), Bethesda, Maryland, Apr 2006.

SA-IFIM: Incrementally Mining Frequent Itemsets in Update Distorted Databases*

Jinlong Wang, Congfu Xu**, Hongwei Dan, and Yunhe Pan

Institute of Artificial Intelligence, Zhejiang University
Hangzhou, 310027, China
zjupaper@yahoo.com, xucongfu@cs.zju.edu.cn,
danhow2008@hotmail.com, panyh@sun.zju.edu.cn

Abstract. The issue of maintaining privacy in frequent itemset mining has attracted considerable attentions. In most of those works, only distorted data are available which may bring a lot of issues in the data-mining process. Especially, in the dynamic update distorted database environment, it is nontrivial to mine frequent itemsets incrementally due to the high counting overhead to recompute support counts for itemsets. This paper investigates such a problem and develops an efficient algorithm SA-IFIM for incrementally mining frequent itemsets in update distorted databases. In this algorithm, some additional information is stored during the earlier mining process to support the efficient incremental computation. Especially, with the introduction of *supporting aggregate* and representing it with bit vector, the transaction database is transformed into machine oriented model to perform fast support computation. The performance studies show the efficiency of our algorithm.

1 Introduction

Recently, privacy becomes one of the prime concerns in data mining. For not compromising the privacy, most of works make use of distortion or randomization techniques to the original dataset, and only the disguised data are shared for data mining [1,2,3].

Mining frequent itemset models from the distorted databases with the reconstruction methods brings expensive overheads as compared to directly mining original data sets [2]. In [3,4], the basic formula from set theory are used to eliminate these counting overheads. But, in reality, for many applications, a database is dynamic in the sense. The changes on the data set may invalidate some existing frequent itemsets and introduce some new ones, so the incremental algorithms [5,6] were proposed for addressing the problem. However, it is not efficient to directly use these incremental algorithms in the update distorted database, because of the high counting overhead to recompute support for itemsets. Although

* Supported by the Natural Science Foundation of China (No. 60402010), Zhejiang Provincial Natural Science Foundation of China (Y105250) and the Science-Technology Program of Zhejiang Province of China (No. 2004C31098).

** Corresponding author.

[7] has proposed an algorithm for incremental updating, the efficiency still cannot satisfy the reality.

This paper investigates the problem of incremental frequent itemset mining in update distorted databases. We first develop an efficient incremental updating computation method to quickly reconstruct an itemset's support by using the additional information stored during the earlier mining process. Then, a new concept *supporting aggregate* (SA) is introduced and represented with bit vector. In this way, the transaction database is transformed into machine oriented model to perform fast support computation. Finally, an efficient algorithm SA-IFIM (**S**upporting **A**ggregate based **I**ncremental **F**requent **I**temset **M**ining in update distorted databases) is presented to describe the process. The performance studies show the efficiency of our algorithm.

The remainder of this paper is organized as follows. Section 2 presents the SA-IFIM algorithm step by step. The performance studies are reported in Section 3. Finally, Section 4 concludes this paper.

2 The SA-IFIM Algorithm

In this section, the SA-IFIM algorithm is introduced step by step. Before mining, the data sets are distorted respectively using the method mentioned by EMASK [3]. In the following, we first describe the preliminaries about incremental frequent itemsets mining, then investigate the essence of the updating technique and use some additional information recorded during the earlier mining and the set theory for quick updating computation. Next, we introduce the *supporting aggregate* and represent it with bit vector to transform the database into machine oriented model for speeding up computations. Finally, the SA-IFIM algorithm is summarized.

2.1 Preliminaries

In this subsection, some preliminaries about the concept of incremental frequent itemset mining are presented, summarizing the formal description in [5,6].

Let D be a set of transactions and $I = \{i_1, i_2, \dots, i_m\}$ a set of distinct literals (items). For a dynamic database, old transactions Δ^- are deleted from the database D and new transactions Δ^+ are added. Naturally, $\Delta^- \subseteq D$. Denote the updated database by D' , therefore $D' = (D - \Delta^-) \cup \Delta^+$, and the unchanged transactions by $D^- = D - \Delta^-$. Let Fp express the frequent itemsets in the original database D , Fp_k denote k -frequent itemsets. The problem of incremental mining is to find frequent itemsets (denoted by Fp') in D' , given Δ^-, D^-, Δ^+ , and the mining result Fp , with respect to the same user specified minimum support s . Furthermore, the incremental approach needs to take advantage of previously obtained information to avoid rerunning the mining algorithms on the whole database when the database is updated. For the clarity, we present s as a relative support value, but $\delta_c^+, \delta_c^-, \sigma_c$, and σ'_c as absolute ones, respectively in $\Delta^+, \Delta^-, D, D'$. And set δ_c as the change of support count of itemset c . Then $\delta_c = \delta_c^+ - \delta_c^-$, $\sigma'_c = \sigma_c + \delta_c^+ - \delta_c^-$.

2.2 Efficient Incremental Computation

Generally, in dynamically updating environment, the important aspect of mining is how to deal with the frequent itemsets in D , recorded in Fp , and how to add the itemsets, which are non-frequent in D (not existing in Fp) but frequent in D' . In the following, for simplicity, we define $|\bullet|$ as the tuple number in the transaction database.

1. For the frequent itemsets in Fp , find the non-frequent or still available frequent itemsets in the updated database D' .

Lemma 1. *If $c \in Fp$ ($\sigma_c \geq |D| \times s$), and $\delta_c \geq (|\Delta^+| - |\Delta^-|) \times s$, then $c \in Fp'$.*

Proof. $\sigma'_c = \sigma_c + \delta_c^+ - \delta_c^- \geq (|D| \times s + |\Delta^+| \times s - |\Delta^-| \times s) = (|D| + |\Delta^+| - |\Delta^-|) \times s = |D'| \times s. \quad \square$

Property 1. When $c \in Fp$, and $\delta_c < (|\Delta^+| - |\Delta^-|) \times s$, then $c \in Fp'$ if and only if $\sigma'_c \geq |D'| \times s$.

2. For itemsets which are non-frequent in D , mine the frequent itemsets in the changed database $\Delta^+ - \Delta^-$ and recompute their support counts through scanning D^- .

Lemma 2. *If $c \notin Fp$, and $\delta_c < (|\Delta^+| - |\Delta^-|) \times s$, then $c \notin Fp'$.*

Proof. Refer to Lemma 1. \square

Property 2. When $c \notin Fp$, and $\delta_c \geq (|\Delta^+| - |\Delta^-|) \times s$, then $c \in Fp'$ if and only if $\sigma'_c \geq |D'| \times s$.

Under the framework of symbol-specific distortion process in [3], ‘1’ and ‘0’ in the original database are respectively flipped with $(1 - p)$ and $(1 - q)$. In incremental frequent itemset mining, the goal is to mine frequent itemsets from the distorted databases with the information obtained during the earlier process. To test the condition for an itemset not in Fp in the situation *Property 2*, we need reconstruct an itemset’s support in the unchanged database D^- through scanning D^{-*} . Not only the distorted support of the itemset itself, but also some other counts related to it need to be tracked of. This makes that the support count computing in *Property 2* is difficult and paramount important in incremental mining. And it is nontrivial to directly apply traditional incremental algorithms to it. To address the problem, an efficient incremental updating operation is first developed through computation with the support in the distorted database, then another method is presented to improve the support computation efficiency in the section 2.3.

In distorted databases, the support computations of frequent itemsets are tedious. Motivated by [3], the similar support computation method is used in incremental mining. With the method, for computing an itemset’s support, we should have the support counts of all its subsets in the distorted database. However, if we save the support counts of all the itemsets, this will be unpractical

and greatly increase cost and degrade indexing efficiency. Thus in incremental mining, when recording the frequent itemsets and their support counts, the corresponding ones in each distorted database are registered at the same time. In this way, for a k -itemset not in Fp , since all its subsets are frequent in the database, we can use the existing support counts in each distorted database to compute and reconstruct its support in the updated database quickly. Thus, the efficiency is improved.

2.3 Supporting Aggregate and Database Transformation

In order to improve the efficiency, we introduce the concept *supporting aggregate* and use bit vector to represent it. By virtue of *elementary supporting aggregate* based on bit vector, the database is transformed into the machine oriented data model, which improves the efficiency of itemsets' support computation.

In the following statement, for transaction database D , let U denote a set of objects (universe), as unique identifiers for the transactions. For simplicity, we refer U as the transactions without differences. For an itemset $A \subseteq I$, a transaction $u \in U$ is said to contain A if $A \subseteq u$.

Definition 1. *supporting aggregate (SA).* For an attribute itemset $A \subseteq I$, denote $S(A) = \{u \in U | A \subseteq u\}$ as its supporting aggregate, where $S(A)$ is the aggregate, composed of the transactions including the attribute itemset A . Generally, $S(A) \subseteq U$. For the supporting aggregate of each attribute items, we call it *elementary supporting aggregate (ESA)*.

Using ESA, the original transaction database is vertically inverted and transformed into attribute-transaction list. Through the ESA, the SA of an itemset can be obtained quickly with set intersection. And the itemsets' support can be efficiently computed. In order to further improve processing speed, for each SA (ESA), we denote it as BV-SA (BV-ESA) with a binary vector of $|U|$ dimensions ($|U|$ is the number of transaction in U). If an itemset's SA contains the i th transaction, its binary vector's i th dimension is set to 1, otherwise, the corresponding position is set to 0. By this representation, the support count of each attribute item can be computed efficiently.

With the vertical database representation, where each row presents an attribute's BV-ESA, the attribute items can be removed sequentially due to download closure property [8], which efficiently reduced the size of the data set. On the other hand, the whole BV-ESA sometimes cannot be loaded into memory entirely because of the memory constraints. Our approach seeks to solve the scalable problem through horizontally partitioning the transaction data set into subsets, which is composed of partial objects (transactions), then load them partition by partition. Through the method, each partition is disjointed with each other, which makes it suitable for the parallel and distributed processing. Furthermore, in reality, the optimizational memory swap strategy can be adopted to reduce the I/O cost.

2.4 Process of SA-IFIM Algorithm

In this subsection, the algorithm SA-IFIM is summarized as Algorithm 1. When the distorted data sets D^{-*} , Δ^{-*} and Δ^{+*} are firstly scanned, they are transformed into the corresponding vertical bit vector representations $BV(D^{-*})$, $BV(\Delta^{-*})$ and $BV(\Delta^{+*})$ partition by partition, and saved into hard disk. From the representations, frequent k -itemsets Fp_k can be obtained level by level. And based on the candidate set generation-and-test approach, candidate frequent k -itemsets (C_k) are generated from frequent $(k-1)$ -itemsets (Fp_{k-1}).

Algorithm 1. Algorithm SA-IFIM

Input: D^{-*} , Δ^{+*} , Δ^{-*} , Fp (Frequent itemsets and the support counts in D), Fp^* (Frequent itemsets of Fp and the corresponding support counts in D^*), minimum support s , and distortion parameter p, q as EMASK [3].

Output: Fp' (Frequent itemsets and the support counts in D')

Method: As shown in Fig.1. In the algorithm, we use some temporal files to store the support counts in the distorted database for efficiency.

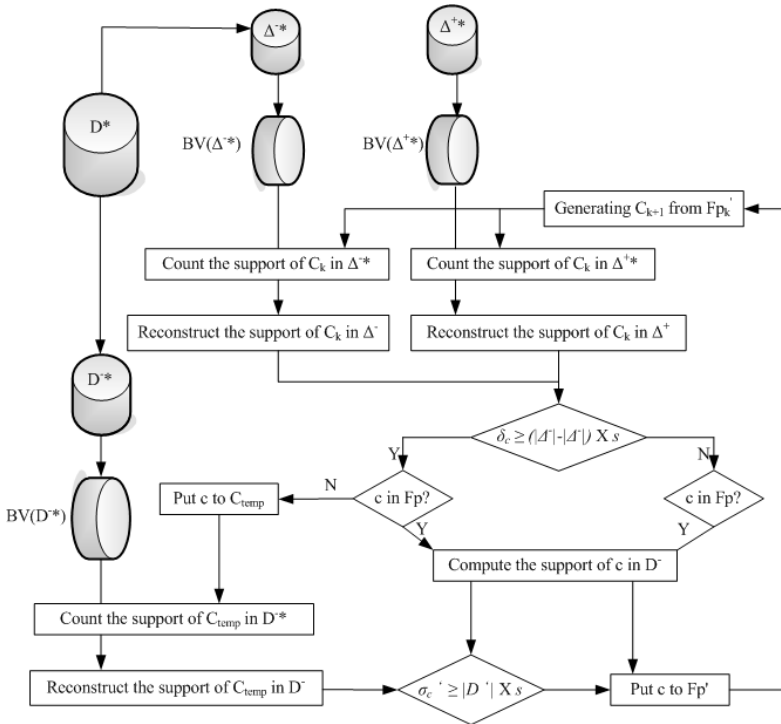


Fig. 1. SA-IFIM algorithm diagram

3 Performance Evaluation

This section performed comprehensive experiments to compare SA-IFIM with EMASK, provided by the authors in [9]. And for the better performance evaluation, we also implemented the algorithm IFIM (Similar as IPPFIM [7]). All programs were coded in C++ using Cygwin with gcc 2.9.5. The experiments were done on a P4, 3GHz Processor, with 1G memory. SA-IFIM and IFIM yield the same itemsets as EMASK with the same data set and the same minimum support parameters.

Our experiments were performed on the synthetic data sets by IBM synthetic market-basket data generator [8]. In the following, we use the notation as D (number of transactions), T (average size of the transactions), I (average size of the maximal potentially large itemsets), and N (number of items), and set $N=1000$. In our method, the sizes of $|\Delta^+|$ and $|\Delta^-|$ are not required to be the same. Without loss of generality, let $|d|=|\Delta^+|=|\Delta^-|$ for simplicity. For the sake of clarity, TxIyDmdn is used to represent an original database with an update database, where the parameters $T = x$ and $I = y$ are the same, only different in the number of the original transaction database $|D| = m$ and the update transaction database $|d| = n$.

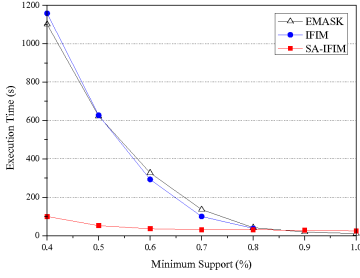
In the following, we used the distorted benchmark data sets as the input databases to the algorithms. The distortion parameters are same as EMASK [3], with $p=0.5$ and $q=0.97$. In the experiments, for a fair comparison of algorithms and scalable requirements, SA-IFIM is run where only 5K transactions are loaded into the main memory one time.

3.1 Different Support Analysis

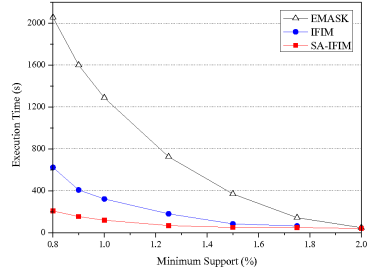
In Fig.2, the relative performance of SA-IFIM, IFIM and EMASK are compared on two different data sets, T25I4D100Kd10K (sparse) and T40I10D100Kd10K (dense) with respect to various minimum support. As shown in Fig.2, SA-IFIM leads to prominent performance improvement. Explicitly, on the sparse data sets (T25I4D100Kd10K), IFIM is close to EMASK, and SA-IFIM is orders of magnitude faster than them; on the dense data sets (T40I10D100Kd10K), IFIM is faster than EMASK, but SA-IFIM also outperforms IFIM, and the margin grows as the minimum support decreases.

3.2 Effect of Update Size

Two data sets T25I4D100Kdm and T40I10D100Kdm were experimented, and the results shown in Fig.3. As expected, when the same number of transactions are deleted and added, the time of rerunning EMASK maintains constant, but the one of IFIM increases sharply and surpass EMASK quickly. In Fig.3, the execution time of SA-IFIM is much less than EMASK. SA-IFIM still significantly outperforms EMASK, even when the update size is much large.

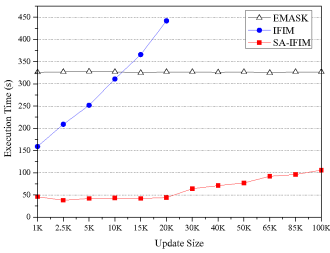


(a) T25I4D100Kd10K

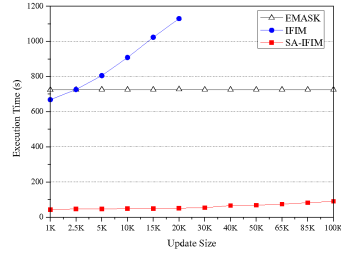


(b) T40I10D100Kd10K

Fig. 2. Extensive analysis for different support

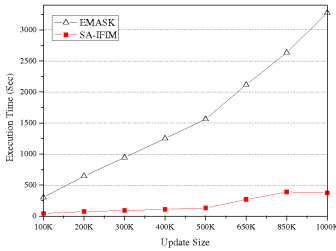


(a) T25I4D100Kdm(s=0.6%)

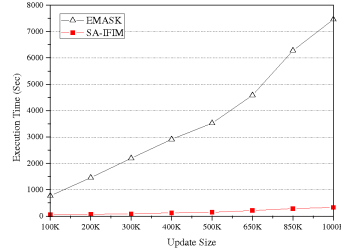


(b) T40I10D100Kdm(s=1.25%)

Fig. 3. Different updating tuples analysis



(a) T25I4Dmd(m/10)(s=0.6%)



(b) T40I10Dmd(m/10)(s=1.25%)

Fig. 4. Scale up performance analysis

3.3 Scale Up Performance

Finally, to assess the scalability of the algorithm SA-IFIM, two experiments, T25I4Dmd(m/10) at $s = 0.6\%$ and T40I10Dmd(m/10) at $s = 1.25\%$, were conducted to examine the scale up performance by enlarging the number of mined data set. The scale up results for the two data sets are obtained as Fig.4, which shows the impact of $|D|$ and $|d|$ to the algorithms SA-IFIM and EMASK.

In the experiments, the size of the update database is as 10% of the original database, and the size of the transaction database m was increased from 100K to 1000K. As shown in Fig.4, EMASK is very sensitive to the updating tuple but SA-IFIM is not, and the execution time of SA-IFIM increases linearly as the database size increases. This shows that the algorithm can be applied to very large databases and demonstrates good scalability of it.

4 Conclusions

In this paper, we explore the issue of frequent itemset mining under the dynamically updating distorted databases environment. We first develop an efficient incremental updating computation method to quickly reconstruct an itemset's support. Through the introduction of the supporting aggregate represented with bit vector, the databases are transformed into the representations more accessible and processible by computer. The support count computing can be accomplished efficiently. Experiments conducted show that SA-IFIM significantly outperforms EMASK of mining the whole updated database, and also have the advantage of the incremental algorithms only based on EMASK.

References

1. Agrawal, R., and Srikant, R.: Privacy-preserving data mining. In: Proceedings of SIGMOD. (2000) 439-450
2. Rizvi, S., and Haritsa, J.: Maintaining data privacy in association rule mining. In: Proceedings of VLDB. (2002) 682-693
3. Agrawal, S., Krishnan, V., and Haritsa, J.: On addressing efficiency concerns in privacy-preserving mining. In: Proceedings of DASFAA. (2004) 113-124
4. Xu, C., Wang, J., Dan, H., and Pan, Y.: An improved EMASK algorithm for privacy-preserving frequent pattern mining. In: Proceedings of CIS. (2005) 752-757
5. Cheung, D., Han, J., Ng, V., and Wong, C.: Maintenance of discovered association rules in large databases: An incremental updating technique. In: Proceedings of ICDE. (1996) 104-114
6. Cheung, D., Lee, S., and Kao, B.: A general incremental technique for updating discovered association rules. In: Proceedings of DASFAA. (1997) 106-114
7. Wang, J., Xu, C., and Pan, Y.: An Incremental Algorithm for Mining Privacy-Preserving Frequent Itemsets. In: Proceedings of ICMLC. (2006)
8. Agrawal, R., and Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of VLDB. (1994) 487-499
9. <http://dsl.serc.iisc.ernet.in/projects/software/software.html>.

Study of Positive and Negative Association Rules Based on Multi-confidence and Chi-Squared Test*

Xiangjun Dong¹, Fengrong Sun², Xiqing Han³, and Ruilian Hou¹

¹ School of Information Science and Technology, Shandong Institute of Light Industry,
250100 Jinan, China

{dxj, hrl}@sdili.edu.cn

² School of Information Science and Engineering, Shandong University, 250100 Jinan, China
sunfr@sdu.edu.cn

³ Dept. of business administration, Shandong Institute of Commerce and Technology,
250103 Jinan, China
hanxq@sict.edu.cn

Abstract. Using a single confidence threshold will result in a dilemmatic situation when simultaneously studying *positive and negative association rule* (PNAR), i.e., the forms $A \Rightarrow B$, $A \Rightarrow \neg B$, $\neg A \Rightarrow B$ and $\neg A \Rightarrow \neg B$. A method based on four confidence thresholds for the four forms of PNARs is proposed. The relationships among the four confidences, which show the necessity of using multiple confidence thresholds, are also discussed. In addition, the chi-squared test can avoid generating misleading rules that maybe occur when simultaneously studying the PNARs. The method of how to apply chi-squared test in mining association rules is discussed. An algorithm PNARMC based on the chi-squared test and the four confidence thresholds is proposed. The experimental results demonstrate that the algorithm can not only generate PNARs rightly, but can also control the total number of rules flexibly.

1 Introduction

Traditional *association rule* (AR) is the form $A \Rightarrow B$, which is used to find the relationships of item sets in transaction database. $A \Rightarrow B$ is a valid rule if its support $s(A \Rightarrow B)$ and confidence $c(A \Rightarrow B)$ meet minimum support (*mins*) and minimum confidence (*minc*) [1]. As an important supplement of the form $A \Rightarrow B$, this paper will study the other three forms $A \Rightarrow \neg B$, $\neg A \Rightarrow B$ and $\neg A \Rightarrow \neg B$, which are called *negative association rules* (NAR), and the traditional form $A \Rightarrow B$ *positive association rules* (PAR) [2].

NARs can provide much useful information for decision-making. They play an important role in many areas, especially in competitive analysis and investment analysis. So, the study of NAR is being attached much more attention recently. The negative relationships between two frequent items were first mentioned in 1997 [3].

* This work was supported by the National Nature Science Foundation of China (NNSFC) under the grant 60271015.

The extended association rules and atom association rules are discussed in [4]. In [5] the authors propose strong negative association rules. In [2] and [6] the authors proposed a PR (Probability Ratio) model and proposed a new method to discover both positive and negative association rules in 2002, and in 2004 they proposed an improved efficient method that could be used to large databases with a pruning strategy and an interestingness measure [7]. In [8] the authors propose a genetic algorithm to discover negative association rules.

However, the above-mentioned papers do not consider the problems caused by using a single confidence threshold when mining positive and negative association rules (PNARs) simultaneously. Intuitively, in basket analysis, the support of almost every item set is very low. Suppose we have two item sets A and B , their support $s(A)$ and $s(B)$ are low, the confidence $c(A \Rightarrow B)$ may be low or high, but $c(\neg A \Rightarrow \neg B)$ must be very high. If we use a single confidence threshold, a dilemmatic situation would occur: if we use a lower confidence threshold, the numbers of PNARs would be so many that the user can not easily choose the right rules; if we use a higher confidence threshold, some valued PARs would be missing. A good solution is using four different confidence thresholds for the four forms of rules. But the relationships among the four confidences must be taken careful consideration when setting these confidence thresholds. Although some articles have discussed the question using different supports and confidences to mine association rules [9], but they are limited to the positive association rules, which is different from this paper.

Furthermore, some self-contradictory association rules would occur when we consider the positive and negative association rules simultaneously.

Example 1. Suppose we are interested in analyzing the transactions of apples (denoted by A) and bananas (denoted by B) in a supermarket. In 10000 transactions, 6000 transactions contain apples, 6500 transactions contain bananas and 3600 transactions contain both apples and bananas. Then,

$$s(A \cup B) = 3600/10000 = 0.36, \text{ and} \\ c(A \Rightarrow B) = 3600/6000 = 0.60.$$

Suppose $mins = 0.25$ and $minc = 0.55$, the rule $A \Rightarrow B$ is a valid association rule, which namely shows that increasing the transactions of apples can correspondingly increase the transactions of bananas. But is it true? Let's see another rule $\neg A \Rightarrow B$.

$$s(\neg A \cup B) = s(B) - s(A \cup B) = 0.65 - 0.36 = 0.29 > mins, \text{ and} \\ c(\neg A \Rightarrow B) = s(\neg A \cup B) / (1 - s(A)) = 0.725 > minc.$$

Obviously, $\neg A \Rightarrow B$ is also a valid association rule, which shows that reducing the transactions of apples can correspondingly increase the transactions of bananas. Obviously, the two valid rules are self-contradictory. In fact, the transaction of apples and bananas is negatively correlated. Buying one will reduce the possibility of buying another. Here the rule $A \Rightarrow B$ is a misleading rule; another case that can generate misleading rules is when two item sets are independent of each other, buying one doesn't affect buying another.

In order to eliminate these misleading rules, many kinds of measure are proposed, such as interestingness, chi-squared test, correlation coefficient, Laplace, Gini-index, Piatetsky-Shapiro, Conviction and so on [10,11]. In these measures, the chi-squared

test is one of the most interesting measures because of its mature theoretical basis. The authors in [3,12] propose a method to test the correlation and the independence of association rules using chi-squared test. In [13] the authors use chi-squared test to prune non-actionable rules or rules at lower significance level. In [14] the authors also use chi-squared independent test in rules classification. In [15] the authors propose a method to express the chi-squared value with support, confidence and lift.

However, the purpose of using the chi-squared test in existing works is to find those item sets whose chi-squared value shows negative correlation and then in turn avoid these item sets generating rules, while our works are not only to find out PARs in the item sets with positive correlation, to avoid generating rules in independent item sets, but also to find out NARs in the item sets with negative correlation.

The contribution of this paper lies in three aspects: 1) study the relationships of the four confidences; 2) look for a method of how to apply chi-squared test in mining PNARs; and 3) propose an algorithm to generate PNARs based on chi-squared test and four confidence thresholds.

The rest of this paper is organized as follows. Section 2 discusses the relationships among the four confidences. Section 3 discusses the method of how to use the chi-squared test in mining PNARs. Section 4 is about algorithm design. Section 5 is about experiment and comparisons. Section 6 is conclusion.

2 The Study of the Relationships Among the Four Confidences

Now we discuss the relationships among the four confidences in the following four cases (A and B are two item sets).

(1) $s(A)$ and $s(B)$ are very low (the typical case in basket analysis); (2) $s(A)$ and $s(B)$ are very high;

(3) $s(A)$ is very high, $s(B)$ is very low; (4) $s(A)$ is very low, $s(B)$ is very high.

However, the concept of high or low here is not explicit. In order to discuss the relationships easily, we give an explicit value that the high is not less than 0.9 and the low not more than 0.1.

Based on the study in [16], we can express the confidences $c(A \Rightarrow \neg B)$, $c(\neg A \Rightarrow B)$ and $c(\neg A \Rightarrow \neg B)$ as the functions of $s(A)$, $s(B)$ and $c(A \Rightarrow B)$ as follows:

$$c(A \Rightarrow \neg B) = \frac{s(A) - s(A \cup B)}{s(A)} = 1 - c(A \Rightarrow B); \quad (1)$$

$$c(\neg A \Rightarrow B) = \frac{s(B) - s(A \cup B)}{1 - s(A)} = \frac{s(B) - s(A) * c(A \Rightarrow B)}{1 - s(A)}; \quad (2)$$

$$c(\neg A \Rightarrow \neg B) = \frac{1 - s(A) - s(B) + s(A \cup B)}{1 - s(A)} = \frac{1 - s(A) - s(B) + s(A) * c(A \Rightarrow B)}{1 - s(A)} \quad (3)$$

Obviously, we can get: (1) $c(A \Rightarrow B) + c(A \Rightarrow \neg B) = 1$ and (2) $c(\neg A \Rightarrow B) + c(\neg A \Rightarrow \neg B) = 1$. The range of the value $c(A \Rightarrow B)$ can be shown in Theorem 1.

Theorem 1. $\text{MAX}(0, \frac{s(A)+s(B)-1}{s(A)}) \leq c(A \Rightarrow B) \leq \text{MIN}(1, \frac{s(B)}{s(A)})$.

Proof: Because $s(A \cup B) \leq \text{MIN}(s(A), s(B))$, thus, $c(A \Rightarrow B) \leq \text{MIN}(1, \frac{s(B)}{s(A)})$;

The minimal value of $s(A \cup B)$ obviously is 0 ,but when $s(A)+s(B)>1$,the minimal value of $s(A \cup B)$ is $s(A)+s(B)-1$, so $c(A \Rightarrow B) \geq \text{MAX}(0, (s(A)+s(B)-1)/s(A))$.

We can easily calculate the value ranges of $c(A \Rightarrow B)$, $c(\neg A \Rightarrow B)$ and $c(\neg A \Rightarrow \neg B)$ according to Theorem 1 and Equation (1) to (3). Table 1 shows the value ranges of the four confidences with the different value of $s(A)$ and $s(B)$.

From Table 1 we can see that the value of confidence has an important impact on the quantity of association rules. Take the first case(the typical basket analysis) for example: if we set $\text{minc}=0.6$, there would be a large number of rules of the form $\neg A \Rightarrow \neg B$ in the result sets, but not any rule of the form $\neg A \Rightarrow B$. It is obviously unreasonable. In order to get the rules of all the four forms, multi-confidences for different forms of rules are necessary in real application.

This paper uses minc_{11} , minc_{10} , minc_{01} and minc_{00} to express the four minimum confidence sof the four forms of rules $A \Rightarrow B$, $c(A \Rightarrow B)$, $c(\neg A \Rightarrow B)$ and $c(\neg A \Rightarrow \neg B)$ respectively.

Table 1. The value ranges of the four confidences with different $s(A)$ and $s(B)$

	$c(A \Rightarrow B)$	$c(A \Rightarrow \neg B)$	$c(\neg A \Rightarrow B)$	$c(\neg A \Rightarrow \neg B)$
$s(A)=0.1, s(B)=0.1$	[0,1]	[0,1]	[0,0.11]	[0.89,1]
$s(A)=0.9, s(B)=0.9$	[0.89,1]	[0,0.11]	[0,1]	[0,1]
$s(A)=0.1, s(B)=0.9$	[0,1]	[0,1]	[0.89,1]	[0,0.11]
$s(A)=0.9, s(B)=0.1$	[0,0.11]	[0.89,1]	[0,1]	[0,1]

3 The Study of Applying Chi-Squared Test to Mining PNARs

In statistics, chi-squared test is a widely used method to test the independence or correlation among variables. The variables are first assumed independent, and then their chi-squared value is calculated. If the value is more than a given value (the value is 3.84 when the significance level $\alpha=0.05$), then the independent assumption would be

refused. The chi-squared value can be calculated by $\chi^2 = \sum \frac{[f_o - f_E]^2}{f_E}$, where f_o

denotes observing frequency, f_E expected frequency. Specifically, for two random variables (X, Y) , their n samples of $(X_1, Y_1) (X_2, Y_2) \dots, (X_n, Y_n)$ are placed in a contingency

table. Suppose the contingency table has R rows and C columns, the frequency of a sample falling into certain a cell denotes as N_{ij} . Then we can get: $N_{i\cdot} = \sum_{j=1}^C N_{ij}$,

$$N_{\cdot j} = \sum_{i=1}^R N_{ij}, \text{ and}$$

$$\chi^2 = \sum \frac{[f_o - f_e]^2}{f_e} = \sum_{i=1}^R \sum_{j=1}^C \frac{(N_{ij} - \frac{1}{n} N_{i\cdot} N_{\cdot j})^2}{\frac{1}{n} N_{i\cdot} N_{\cdot j}}. \tag{4}$$

It obeys χ^2 -distribution of $(R-1)(C-1)$ degrees of freedom, where $f_o = N_{ij}$, $f_e = \frac{1}{n} N_{i\cdot} * N_{\cdot j}$.

Chi-squared testⁿ could be used in multi-dimensional variables, but in association rule we only need two-dimensional variables, i.e., two-dimensional

Table 2. Contingency table of items A and B

	A	$\neg A$	Σ
B	$s(A \cup B) * n$	$s(\neg A \cup B) * n$	$s(B) * n$
$\neg B$	$s(A \cup \neg B) * n$	$s(\neg A \cup \neg B) * n$	$s(\neg B) * n$
Σ	$s(A) * n$	$s(\neg A) * n$	n

contingency table, its degree of freedom is 1. The contingency table of item sets A and B is shown in table 2, where n denotes the total number of transactions in database. The purpose of the data expressed in the form of support in Table 2 is to associate the chi-squared test with the concepts of association rules easily.

According to Equation (4), we can get:

$$\begin{aligned} \chi^2 &= \frac{[s(A \cup B) * n - \frac{1}{n} s(A) * n * s(B) * n]^2}{\frac{1}{n} s(A) * n * s(B) * n} + \frac{[s(\neg A \cup B) * n - \frac{1}{n} s(\neg A) * n * s(B) * n]^2}{\frac{1}{n} s(\neg A) * n * s(B) * n} \\ &+ \frac{[s(A \cup \neg B) * n - \frac{1}{n} s(A) * n * s(\neg B) * n]^2}{\frac{1}{n} s(A) * n * s(\neg B) * n} + \frac{[s(\neg A \cup \neg B) * n - \frac{1}{n} s(\neg A) * n * s(\neg B) * n]^2}{\frac{1}{n} s(\neg A) * n * s(\neg B) * n} \\ &= \frac{n * [s(A \cup B) - s(A)s(B)]^2}{s(A)s(B)(1 - s(A))(1 - s(B))} \end{aligned} \tag{5}$$

Here is an instance. Table 3 shows the contingency table of data in Example 1. Its chi-squared value is:

$$\frac{\left(3600 - \frac{6000 * 6500}{10000}\right)^2}{\frac{6000 * 6500}{10000}} + \frac{\left(2900 - \frac{4000 * 6500}{10000}\right)^2}{\frac{4000 * 6500}{10000}} + \frac{\left(2400 - \frac{6000 * 3500}{10000}\right)^2}{\frac{6000 * 3500}{10000}} + \frac{\left(1100 - \frac{4000 * 3500}{10000}\right)^2}{\frac{4000 * 3500}{10000}} = 16$$

4.8 > 3.84.

Since 16.48 are more than 3.84, we reject the independence assumption at the 5% significance level. Thus, the transactions of apples and bananas are correlated.

But how to judge the correlation between item sets are positive or negative?

Definition 1. The correlation of item sets A and B can be defined as $corr(A,B)=f_o/f_E$, where f_o and f_E is the observing frequency and expected frequency respectively of the cell that A and B cross. Then we can get:(1)if $corr(A,B) > 1$, then $A \Rightarrow B$ is positively correlated, and (2) if $corr(A,B) < 1$, then $A \Rightarrow B$ is negatively correlated .

According to Definition 1 and Table 2 we can get:

- (1) $corr(A,B)=s(A \cup B)/(s(A)s(B))$;
- (2) $corr(A, \neg B)=s(A \cup \neg B)/(s(A)s(\neg B))$;
- (3) $corr(\neg A, B)=s(\neg A \cup B)/(s(\neg A)s(B))$; and
- (4) $corr(\neg A, \neg B)=s(\neg A \cup \neg B)/(s(\neg A)s(\neg B))$.

Now we discuss the relationships of the above four correlations.

Theorem 2. If $corr(A,B) > 1$, then (1) $corr(A, \neg B) < 1$; (2) $corr(\neg A, B) < 1$; (3) $corr(\neg A, \neg B) > 1$ and vice versa.

Table 3. Contingency table of the data in Example 1

Proof: Here we only prove (3), (1) and (2) can be proved in the same way.

(3) Since $corr(A,B) > 1$, thus $s(A \cup B) > s(A)s(B)$,

so $1-s(A)-s(B)+s(A \cup B) > 1-s(A)-s(B)+s(A)s(B)$.

Because $corr(\neg A, \neg B) > 0$, namely $1-s(A)-s(B)+s(A)s(B) > 0$, therefore,

$$\frac{1-s(A)-s(B)+s(A \cup B)}{1-s(A)-s(B)+s(A)s(B)} = \frac{s(\neg A \cup \neg B)}{s(\neg A)s(\neg B)} > 1, \text{ i.e., } corr(\neg A, \neg B) > 1 .$$

These relationships of the four correlations tell us self-contradictory rules would not occur if we judge the correlation between item sets before generating rules. In detail, we only mine rules of the forms $\neg A \Rightarrow B$ and $A \Rightarrow \neg B$ if $corr(A,B) < 1$ and mine rules of the forms $A \Rightarrow B$ and $\neg A \Rightarrow \neg B$ if $corr(A,B) > 1$. In Example 1, we only mine rules of the forms $\neg A \Rightarrow B$ and $A \Rightarrow \neg B$ because of $corr(A,B) = 0.92 < 1$.

Now we give the definition of positive and negative association rules based on chi-squared test.

Definition 2. Let I be a set of items, D a database, $A, B \subseteq I$ and $A \cap B = \emptyset$. $s(A)$, $s(\neg A)$, $s(B)$ and $s(\neg B) > 0$, $mins$ and $minc$ are given by users. Let χ^2 express the Chi-squared value of A, B and χ_α^2 the critical value at the significance level α . If $\chi^2 \leq \chi_\alpha^2$, A and B are independent, we don't generate any rules from them. Otherwise, A and B are correlated, and

(1) If $corr(A,B) > 1$, $s(A \cup B) \geq mins$ and $c(A \Rightarrow B) \geq minc$ then $A \Rightarrow B$ is a PAR;

(2) If $corr(A, \neg B)$ ($corr(\neg A, B)$, $corr(\neg A, \neg B)$) > 1 , $s(A \cup B) \geq mins$ and $c(A \Rightarrow \neg B)$ ($c(\neg A \Rightarrow B)$, $c(\neg A \Rightarrow \neg B)$) $\geq minc$, then $A \Rightarrow \neg B$ ($\neg A \Rightarrow B$, $\neg A \Rightarrow \neg B$) is a NAR;

It is obvious that chi-squared test is a very effective method. But chi-squared test also has some limitations [3]. The chi-squared test rests on the normal approximation to

the binomial distribution. This approximation breaks down when the expected values are small. As a rule of thumb, statistics texts recommend the use of chi-squared test only if (1) all cells in the contingency table have expected value greater than 1; and (2) at least 80% of the cells in the contingency table have expected value greater than 5.

4 Algorithm Design

According to Definition 2, we propose an algorithm PNARMC (Positive and Negative Association Rules based on Multi-confidence and Chi-squared test). We suppose the frequent item sets (those item sets whose support is great than the minimum support threshold $mins$) are saved in set L .

Algorithm. PNARMC

Input: L : frequent item sets; $minc_11, minc_10, minc_01, minc_00$: minimum confidence; χ_α^2 : the critical value at the significance level α ;

Output: PAR : set of positive associate rule; NAR : set of negative associate rule;

(1) $PAR = \emptyset$; $NAR = \emptyset$;

(2) //mining PNARs in L .

```

for any item set  $X$  in  $L$  do {
  for any item set  $A \cup B = X$  and  $A \cap B = \emptyset$  do {
    calculate  $\chi^2$  with formula 5;
     $corr = s(A \cup B) / (s(A) s(B)) \square$ 
    if  $\chi^2 > \chi_\alpha^2$  then {
      if  $corr > 1$  then {
        (2.1) //generate rules of the forms  $A \Rightarrow B$  and  $\neg A \Rightarrow \neg B$ .
        if  $c(A \Rightarrow B) \geq minc\_11$  then
           $PAR = PAR \cup \{A \Rightarrow B\}$ ;
        if  $c(\neg A \Rightarrow \neg B) \geq minc\_00$  then
           $NAR = NAR \cup \{\neg A \Rightarrow \neg B\}$ ; }
      if  $corr < 1$  then {
        (2.2) //generate rules of the forms  $A \Rightarrow \neg B$  and  $\neg A \Rightarrow B$ .
        if  $c(A \Rightarrow \neg B) \geq minc\_10$  then
           $NAR = NAR \cup \{A \Rightarrow \neg B\}$ ;
        if  $c(\neg A \Rightarrow B) \geq minc\_01$  then
           $NAR = NAR \cup \{\neg A \Rightarrow B\}$ ; }
    } } }

```

(3) **return** PAR and NAR ;

Step (1) initializes PAR and NAR with empty set. Step (2) calculates chi-squared value χ^2 and correlation flag $corr$ and generates rules. Step (2.1) generates rules of the forms $A \Rightarrow B$ and $\neg A \Rightarrow \neg B$ and step (2.2) generates rules of the forms $A \Rightarrow \neg B$ and $\neg A \Rightarrow B$. Particularly, it is four confidence thresholds that are used to their corresponding four forms of rules, but not a single confidence threshold. Step (3) returns the result and finishes the whole algorithm.

In fact, the algorithm PNARMC can also be used to generate NARs from infrequent item sets if only we delete the sentence “if $c(A \Rightarrow B) \geq \text{minc_}11$ then $PAR = PAR \cup \{A \Rightarrow B\}$ ” and let L save the infrequent item sets.

5 Experimental Results and Comparison

5.1 Experimental Results

The experimental dataset records areas of www.microsoft.com that each user visited in a one-week timeframe in February 1998. Summary statistical information of the dataset is: 32711 training instances, 5000 testing instances, 294 attributes and the mean area visits per case is 3.0 (http://www.cse.ohio-state.edu/~yanghu/CIS788_dm_proj.htm#datasets). The experimental results are shown in Table 4. The number of PARs generated by traditional algorithm (without considering correlation) is also given for comparison.

Table 4. The number of rules generated by different algorithms and in different confidences ($\text{mins}=0.014$, $\alpha=0.05$)

#	traditional algorithm		PNARMC ($\alpha=0.05$)				total
	$A \Rightarrow B$	$A \Rightarrow \neg B$	$A \Rightarrow \neg B$	$\neg A \Rightarrow B$	$\neg A \Rightarrow \neg B$	##	
$\text{minc_}^*=0.3$	167	158	30	7	339	4	538
$\text{minc_}^*=0.6$	43	43	30	0	339	0	412
$\text{minc_}^*=0.85$	7	7	14	0	263	0	284
$\text{minc_}^*=0.985$	1	1	0	0	32	0	33
$\text{minc_}11=0.6$							
$\text{minc_}10=0.85$	—	43	14	7	32	4	100
$\text{minc_}01=0.3$							
$\text{minc_}00=0.985$							

Note: #: minc_^* denotes $\text{minc_}11$, $\text{minc_}10$, $\text{minc_}01$, $\text{minc_}00$

##: This column expresses the number of rules generated by independent item sets.

From Table 4 we can draw two conclusions as follows:

1. The Chi-squared test is effective.

From Table 4, when $\text{minc_}^*=0.3$, there are 158 PARs generated by PNARMC, while 167 PARs generated by traditional algorithm. This shows that there are 9 misleading PARs detected and eliminated by PNARMC. In these 9 rules, 4 rules are generated by independent item sets, the other 5 rules are maybe generated by negative correlated item sets. Moreover, a lot of NARs are generated by PNARMC. These data adequately show that chi-squared test is effective.

2. Multi-confidence is effective.

First, let's look at the changes of the rule number when using a single minimum confidence minc_^* . When minc_^* changes from 0.3 to 0.6, the rule number of the forms $A \Rightarrow B$ and $\neg A \Rightarrow B$ decreases sharply while the rule number of the forms $A \Rightarrow \neg B$

and $\neg A \Rightarrow \neg B$ doesn't change, the total number of the rules is still great. The only method to decrease the total number of the rules is to increase the minimum confidence $minc_*$. But the higher the minimum confidence $minc_*$ is, the higher the percentage of the rules of the form $\neg A \Rightarrow \neg B$ in the total is, the lower the percentage of the rules of the other three forms in the total is. These data adequately show the shortcomings of using a single minimum confidence.

Second, let's look at the number of the rules when using multiple minimum confidences. when $minc_{11}=0.6$, $minc_{10}=0.85$, $minc_{01}=0.3$ and $minc_{00}=0.985$, the total number of rules is 100, which contain 43 rules of the form $A \Rightarrow B$, 14 rules of the form $A \Rightarrow \neg B$, 7 rules of the form $\neg A \Rightarrow B$, 32 rules of the form $\neg A \Rightarrow \neg B$ and 4 rules generated by independent item sets. It is obvious that the algorithm PNARMC can flexibly control the total number of rules by setting the four confidence thresholds according to the user's actual need. These data adequately show that using multi-confidence is necessary and the algorithm PNARMC is very effective.

5.2 Comparisons

The superiorities of PR model have been illustrated in [6] when the PR model is compared with some other models including interest model, exception mining model and strong negative association model [5]. So, here we only compare the PANRMC model with PR model and positive and negative association rules on correlation (PANRC) model in [16]. The PR model uses a single confidence threshold to generate PARs from frequent item sets and to generate NARs only from infrequent item sets, however, it does not consider the NARs in frequent item sets and does not consider the problems using a single confidence threshold we mentioned above. The PANRC model can generate PARs and NARs from frequent item sets, but it also uses a single confidence threshold. The PANRMC model, as we have seen, has overcome the shortcomings in these models.

6 Conclusions

How to set up the confidence threshold of association rules becomes very important when we simultaneously study the PNARs. It has an important influence on the total number of rules, and in turn on the efficiency for user to choose the valued rules. A single confidence threshold can't meet the need of actual application, while four confidence thresholds are required for the four forms of rule. The relationships of the four confidences are discussed. The corresponding conclusions are very important in how to set up the four minimum confidences. In addition, chi-squared test can avoid generating misleading rules that may occur when simultaneously studying the PNARs. The method of how to apply chi-squared test to mining association rules is discussed. An algorithm PNARMC is proposed to discover both positive and negative association rules based on chi-squared test and four confidence thresholds. The algorithm can not only generate PARs in the item sets with positive correlation, generate NARs in the item sets with negative correlation, but can also detect and eliminate rules generated in independent item sets. It can flexibly control the number of rules, too. The experiment and comparisons demonstrate that the algorithm is effective.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large database. Proceeding of the 1993 ACM SIGMOD International Conference on Management of Data. New York: ACM Press(1993) 207-216
2. Wu, X., Zhang, C., Zhang, S.: Mining both positive and negative association rules. Proceedings of the 19th International Conference on Machine Learning(ICML-2002). San Francisco: Morgan Kaufmann Publishers (2002) 658-665
3. Brin, S., Motwani, R., Silverstein, C.: Beyond Market: Generalizing Association Rules to Correlations. In Processing of the ACM SIGMOD Conference (1997) 265-276
4. Li, X., Liu, Y., Peng, J.: The extended association rules and atom association rules. Journal of Computer Research and Application, China(2002.12)1740-1750
5. Savasere, A., Omiecinski, E., Navathe, S.: Mining for Strong Negative Associations in a Large Database of Customer Transaction. In Proceedings of the 1998 International Conference on Data Engineering(1998) 494-502
6. Zhang, C., Zhang, S.: Association Rule Mining, LNAI 2307, Springer-Verlag Berlin Heidelberg (2002)47-84
7. Wu, X., Zhang, C., Zhang, S.: Efficient Mining of Both Positive and Negative Association Rules, ACM Transactions on Information Systems, **22**(2004), 3: 381-405
8. Boulicaut, J-F., Bykowski, A., Jeudy, B.: Towards the Tractable Discovery of Association Rules with Negations. In Proceedings of the Fourth International Conference on Flexible Query Answering Systems FQAS'00, Warsaw (PL)(2000) 425-434
9. Liu, B., Hsu, W., Ma, Y.: Mining Association Rules with Multiple Minimum Supports. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Diego, CA, USA. (1999)
10. Tan, P., Kumar, V.: Interestingness measures for association patterns: a perspective. KDD-2000 Workshop on Post-processing in Machine Learning and Data Mining(2000)
11. Tan, P-N., Kumar, V., Srivastava, J.: Selecting the Right Interestingness Measure for Association Patterns . Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton (CA) (2002) 32-41
12. Silverstein, C., Brin, S., Motwani, R.: Beyond market baskets: Generalizing association rules to dependence rules. Data Mining and Knowledge Discovery, 2(1)(1998) 39-68
13. Liu, B., Hsu, W., Ma, Y.: Identifying Non-Actionable Association Rules. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Francisco, CA;(2001) 329-334
14. Hilderman, R.J., Hamilton, H.J.: Applying Objective Interestingness Measures in Data Mining Systems. In Proceedings of the 4th European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'00), Lecture Notes in Computer Science, Springer-Verlag, Lyon, France(2000) 432-439
15. Sergio, A. Alvarez.: Chi-squared computation for association rules: preliminary results. Technical Report BC-CS-2003-01 July 2003, Computer Science Dept. Boston College Chestnut Hill, MA 02467 USA (2003)
16. Dong, X., Wang, S., Song, H., Lu, Y.: Study on Negative Association Rules. Transactions of Beijing Institute of Technology, China(2004.11)978-981

Efficiently Mining Maximal Frequent Mutually Associated Patterns

Zhongmei Zhou^{1,2}, Zhaohui Wu¹, Chunshan Wang¹, and Yi Feng¹

¹ College of Computer Science and Technology, Zhejiang University, China

² Department of Computer Science, Zhangzhou Teacher's College, China
{zzm, wzh, cswang, fengyi}@zju.edu.cn

Abstract. Mutually associated pattern mining can find such type of patterns whose any two sub-patterns are associated. However, like frequent pattern mining, when the minimum association threshold is set to be low, it still generates a large number of mutually associated patterns. The huge number of patterns produced not only reduces the mining efficiency, but also makes it very difficult for a human user to analyze in order to identify interesting/useful ones. In this paper, a new task of maximal frequent mutually associated pattern mining is proposed, which can dramatically decrease the number of patterns produced without information loss due to the downward closure property of the association measure and meanwhile improve the mining efficiency. Experimental results show that maximal frequent mutually associated pattern mining is quite a necessary approach to lessening the number of results and increasing the performance of the algorithm. Also, experimental results show that the techniques developed are much effective especially for very large and dense datasets.

1 Introduction

It has been well recognized that frequent pattern mining is essential in many important data mining tasks, such as mining association rules [1, 4], sequential patterns [2], episodes [5], partial periodicity [3], etc. However, in some applications, such as discovering combinations of associated kinds of medicine in TCM (Traditional Chinese Medicine) formula dataset, there is more interest in patterns consisting of infrequent, but highly associated items. Omicinski [6] introduced three alternative interestingness measures, called any-confidence, all-confidence and bond for mining associated patterns. Although Won-young kim [7] and Young-koo lee [8] defined a pattern which satisfies the minimum all-confidence as a correlated pattern, in terms of the definition of all-confidence [6], if a pattern has all-confidence greater than or equal to a given minimum all-confidence, any two sub-patterns X , Y of this pattern have confidence, i.e. conditional probability $P(X/Y)$ and $P(Y/X)$ greater than or equal to the given minimum all-confidence, in other words X and Y are associated. Therefore, in this paper, if a pattern has all-confidence greater than or equal to a given minimum all-confidence, it is called a mutually associated pattern.

However, when the minimum all-confidence is low, it still produces a huge number of mutually associated patterns as frequent pattern mining does. Since the measure

all-confidence has a downward closure property [6], we may mine maximal mutually associated patterns without information loss. Therefore, a new task of maximal frequent mutually associated pattern mining is proposed, which not only significantly decreases the number of patterns generated, but also improves the mining efficiency. Our experiments are performed on three datasets: two dense datasets and a sparse real dataset. Experimental results show that maximal frequent mutually associated pattern mining is quite a valid and necessary approach to decreasing the number of the results and improving the mining efficiency. Also, experimental results show that the techniques developed are much effective for very large and dense databases.

The remainder of this paper is organized as follows: In Section 2, some related definitions are given and an algorithm is developed for finding maximal frequent mutually associated patterns. In Section 3, the experimental results are showed. Section 4 concludes the paper.

2 Mining Maximal Frequent Mutually Associated Patterns

This section first introduces some related concepts, and then gives an example to show the mining process. Finally, an algorithm is developed for finding maximal frequent mutually associated patterns.

We first introduce the measure of all-confidence [6]. Let $T = \{i_1, i_2, \dots, i_m\}$ be a set of m distinct literals called *items* and D be a set of variable length transactions over T . Each transaction contains a set of items, $\{i_{j_1}, i_{j_2}, \dots, i_{j_k}\}$. A transaction also has an associated unique identifier called *TID*. A pattern X is a subset of T . Let $p(X)$ be a power set of a pattern X . The measure all-confidence (denoted as α) of a pattern X is defined as follows [6]:

$$\alpha = \frac{|\{d \mid d \in D \wedge X \subset d\}|}{\text{MAX}\{i \mid \forall l (l \in p(X) \wedge l \neq \phi \wedge l \neq X \wedge i = |\{d \mid d \in D \wedge l \subset d\}|\)}$$

Definition 1 (a frequent mutually associated pattern). A pattern is called a frequent mutually associated pattern if and only if it has support and all-confidence greater than or equal to the given minimum support and the minimum all-confidence respectively.

Definition 2 (a maximal frequent mutually associated pattern). A frequent mutually associated pattern X is called a maximal frequent mutually associated pattern if and only if there exists no pattern Y such that $Y \supset X$ and Y is a frequent mutually associated pattern.

By the definition of all-confidence, it is easy to deduce that if the minimum all-confidence is less than or equal to the minimum support, then patterns which have support greater than or equal to the minimum support must have all-confidence greater than or equal to the minimum all-confidence. Therefore the minimum all-confidence should be set to be higher than the minimum support, or it will lose its function for generating patterns in the mining process.

In this paper, we develop a pre-pruning algorithm for mining the set of all maximal frequent mutually associated patterns. We compare the pre-pruning algorithm with the filter algorithm by experiments in section 3. The filter algorithm is that we first find maximal frequent patterns and then derive all maximal frequent mutually associated patterns from maximal frequent patterns. The pre-pruning algorithm is that we not only use the methods of the filter algorithm but also make use of the downward closure property of the measure all-confidence to prune items which do not satisfy the lemma 1 in advance. From experimental results, we can see that the pre-pruning algorithm has much higher performance than the filter algorithm on dense datasets. Therefore, we give the detail mining process of pre-pruning algorithm in the paper.

The following lemmas provide the theoretical foundations that our algorithm can find maximal frequent mutually associated patterns correctly and efficiently. Let the minimum all-confidence be λ .

Lemma 1. For any item a in the X -conditional database, if a pattern which contains X and a is a maximal frequent mutually associated pattern, the support of a must be less than $\text{sup}(X)/\lambda$.

Lemma 2. If a set Y that contains all frequent items in the X -conditional database appears in every transaction of the X -conditional database, $X \cup Y$ forms a maximal frequent pattern in the X -conditional database and all sub-patterns of Y have the same support in the X -conditional database.

Lemma 3. If all sub-patterns of a pattern Y have the same support and pattern Y is not a maximal frequent mutually associated pattern, then the maximal frequent mutually associated pattern can be derived from Y if and only if by cutting the most frequent item.

Therefore, we can improve the search for maximal mutually associated patterns by earlier pruning items which do not satisfy the lemma 1. Before giving our algorithm, we show the mining process using the following example.

Example. Let us mine maximal frequent mutually associated patterns over transaction database TDB in Table 1, with the minimum support $\xi = 0.4$ and the minimum all-confidence $\eta = 0.5$. The mining process is shown in Figure 1.

Table 1. The transaction database TDB

Transaction ID	Items in transaction
10	a, c, d, e, f
20	a, b, c, e, f
30	c, e, f
40	a, c, d, f
50	c, e .

1. Find frequent items. Scan TDB to find the set of frequent items and derive a (global) frequent item list, called f_list , and $f_list = \langle d : 2, a : 3, f : 4, e : 4, c : 5 \rangle$.

2. Divide search space. The search space can be divided into 5 non-overlap subsets based on the f_list , (1) the ones containing item d , (2) the ones containing item a but no d , (3) the ones containing item f but no a nor d , (4) the ones containing item e but no f , a nor d , and (5) the one containing item only c .

3. Find subsets of maximal frequent mutually associated patterns. The subsets of maximal frequent mutually associated patterns can be mined by constructing corresponding conditional databases and can be mined recursively.

(3a) Find maximal frequent mutually associated patterns containing item d . Only transactions containing d are needed. The d -conditional database, denoted as $TDB|_d$, contains all the transactions having d , which is $\{afec, afc\}$. Notice that item d is omitted in each transaction since it appears in every transaction in the d -conditional database. The support of item d is 2. Since $\text{sup}(c) > \text{sup}(d)/\eta$, c is pruned in advance from the d -conditional database. Items a, f appear twice respectively in $TDB|_d$. That is, every transaction containing d also contains a, f . Moreover e is infrequent since it appears only once in $TDB|_d$. Therefore, pattern daf is a maximal frequent pattern in the d -conditional database after pruning. It is easy to test that pattern daf is a maximal frequent mutually associated pattern which contains item d .

(3b) Find maximal frequent mutually associated patterns containing item a but no d . In the a -conditional database, $TDB|_a = \{fec, fec, fc\}$, $f_a\text{-list} = \{e : 2, f : 3, c : 3\}$. All items $e : 2, f : 3, c : 3$ satisfy the lemma 1 in the a -conditional database, so we first mine maximal frequent mutually associated patterns containing item a but no d in the ae -conditional database. Since $\text{sup}(c) > \text{sup}(ae)/\eta$, we first cut item c from ae -conditional database and then get a maximal frequent mutually associated pattern aef . Then, we mine maximal frequent mutually associated patterns containing item a but no d in the af -conditional database and get a maximal frequent mutually associated pattern afc . Since only pattern ac is in the ac -conditional database and ac is a sub-pattern of pattern afc , ac is not a maximal frequent mutually associated pattern.

(3c) Find maximal frequent mutually associated patterns containing item f but no a nor d . In the f -conditional database, $f_f\text{-list} = \{e : 3, c : 4\}$. We mine maximal frequent mutually associated patterns containing item f but no a nor d in fe -conditional database. Pattern fec is a maximal frequent mutually associated pattern. Only pattern fc is in fc -conditional database and it is a sub-pattern of pattern fec , so fc is not a maximal frequent mutually associated pattern. We finish the mining of maximal frequent mutually associated patterns containing item f but no a nor d .

The remained items e and c in f_list are included in pattern fec , so there are no maximal frequent mutually associated patterns containing item e but no f, a nor d , and no maximal frequent mutually associated patterns containing item only c .

In summary, the complete set of maximal frequent mutually associated patterns is:

$\{daf : 2, aef : 2, afc : 3, fec : 3\}$.

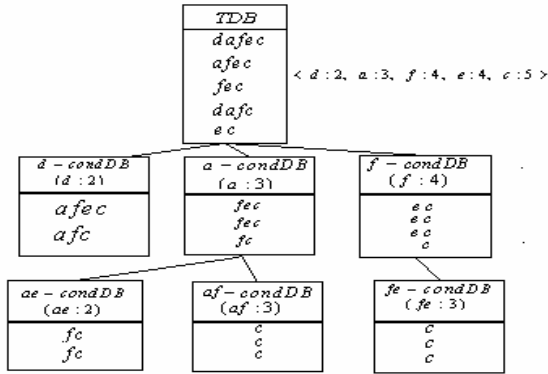


Fig. 1. The process of mining maximal frequent mutually associated patterns

Pre-pruning Algorithm: *MFMAP*: Mining maximal frequent mutually associated patterns.

Input: a transaction database *TDB*, a support threshold ξ and a minimum all-confidence η .

Output: the complete set of maximal frequent mutually associated patterns.

Method:

1. Initialization. Let *FMP* be the set of maximal frequent patterns after pruning in its conditional database, let *MFMAP* be the set of maximal frequent mutually associated patterns. Initialize: $FMP \leftarrow \phi$; $MFMAP \leftarrow \phi$.
2. Find frequent items. Scan transaction database *TDB*, compute frequent item list *f_list*.
3. First prune the items which do not satisfy the lemma 1 and then mine maximal frequent patterns in its conditional database recursively. Call $FMP - (\phi, TDB, f_list, FMP)$.
4. Derive maximal frequent mutually associated patterns from maximal frequent patterns using lemma 2 and lemma 3 recursively. Call $MFMAP - (\phi, TDB, FMP, MFMAP)$.
5. Output all maximal frequent mutually associated patterns *MFMAP*.

Subroutine :

$FMP - (\phi, TDB, f_list, FMP)$, $MFMAP - (\phi, TDB, FMP, MFMAP)$

Parameters:

1. *X*: the frequent pattern if *DB* is an *X*-conditional database, or ϕ if *DB* is *TDB*.
2. *DB*: the transaction database of a condition database;
3. *f_list*: the frequent item list of *DB*.
4. *FMP*: the set of maximal frequent patterns after pruning in its conditional database.
5. *MFMAP*: the set of maximal frequent mutually associated patterns.

Method :

1. Let Y be the set of all items which satisfy lemma 1 in f_list , if Y appears in every transaction of DB , insert $X \cup Y$ to FMP if it is not a proper subset of some pattern in $MFMAP$, else continue to form conditional database until all frequent items which satisfy lemma 1 in certain database appear in every transaction of the conditional database; /lemma 1 and lemma 2.
2. Derive the maximal frequent mutually associated patterns from $X \cup Y$ in the X - conditional database by cutting the most frequent item from pattern Y . If the derived pattern is not a proper subset of some pattern in $MFMAP$, then insert it into $MFMAP$; /lemma 3.
3. Form conditional database for every remaining item in f_list , at the same time, compute local frequent item lists for these condition database;
4. If all the remaining items in (global) f_list are not all included in a certain pattern in $MFMAP$, for each remaining item i in f_list , starting from the first one, call $FMP-(iX, DB/i, f_i-list, FMP)$ and call $MFMAP-$
 $(iX, DB/i, FMP, MFMAP)$, where DB/i is the i - conditional database with respect to DB and f_i-list is the corresponding frequent item list.

3 Experiments

In this section, we report our experimental results. All experiments are performed on three datasets: 1. A dense dataset: Connect-4 game state information dataset, which consists of 67,557 transactions, each with an average length of 43 items. 2. A dense dataset: Mushroom characteristic data, which consists of 8,124 transactions, each with an average length of 23 items. 3. A sparse real dataset: TCM formula dataset, obtained from Information Institute of China Academy of Traditional Chinese Medicine, which consists of 85916 formulas with 26295 kinds of medicine involved, each with an average length of 8 kinds of medicine.

Table 4 shows the number of frequent mutually associated patterns and maximal frequent mutually associated patterns on mushroom dataset when the minimum all-confidence varies with a fix minimum support 1%. From Table 4, we can see that when the minimum all-confidence is less than 30%, the number of maximal frequent mutually associated patterns is significantly less than the number of frequent mutually associated patterns. Figure 2 (IV) shows the execution time of frequent mutually associated pattern mining and maximal frequent mutually associated pattern mining respectively on mushroom dataset when the minimum all-confidence varies with a fix minimum support 1%. From Figure 2 (IV), we can see that the execution time of maximal frequent mutually associated pattern mining is much less than the execution time of frequent mutually associated pattern mining, especially when the minimum all-confidence is low. Therefore, when the minimum all-confidence is low, maximal

frequent mutually associated pattern mining is quite a necessary approach to increasing the performance of the algorithm and reducing the number of patterns produced.

Table 2 and Table 3 show the number of maximal frequent mutually associated patterns generated on Connect-4 game state information dataset and TCM formula dataset respectively when minimum support is fixed and the minimum all-confidence varies. When the minimum all-confidence is low, even the number of maximal frequent mutually associated patterns is large.

Figure 2 (I, II, III) shows the execution time of the pre-pruning algorithm and the filter algorithm on three datasets. From Figure 2 (I, II, III), we conclude that the pre-pruning algorithm always outperforms the filter algorithm especially at a low minimum all-confidence. Furthermore the pre-pruning algorithm is much more efficient than the filter algorithm when the dataset is large and dense.

From experimental results, we have conclusions that: (1) When the dataset is very large, dense and/or the minimum all-confidence is set to be low, maximal frequent mutually associated pattern mining is quite a good means to decrease the number of the results produced and increase the mining efficiency. (2) The pre-pruning algorithm is much efficient and effective even when we mine on very large and dense datasets, such as Connect-4 game state information dataset.

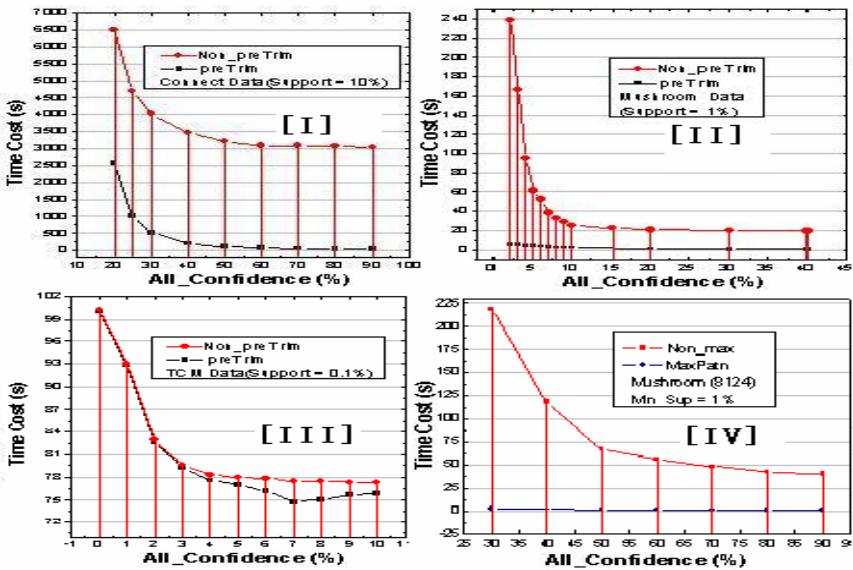


Fig. 2. Execution time on three datasets

Table 2. Number of patterns from the Connect-4 dataset (Support 10%--67557)

All-Conf.(%)	11	13	16	20	25	30	40	50
Number	66571	22805	15119	9630	5241	2313	1535	1138

Table 3. Number of patterns from the TCM formula data (Support : 0.06%)

All-Conf.(%)	1	2	3	4	5	6	7	8
Num(0.06%)	13173	6064	3924	2757	2036	1550	1145	872

Table 4. Number of patterns from the Mushroom dataset (Support 1% --8124)

All-onf.(%)	10	20	30	40	50	60	70	80
Max-Num	961	489	281	182	70	37	17	7
Num	510350	50860	5198	1220	435	208	143	123

4 Conclusions

Frequent mutually associated patterns are more useful for making business decisions than frequent patterns because any two sub-patterns of a frequent mutually associated pattern are associated. However, when the minimum association is low, it still generates a large number of frequent mutually associated patterns. Therefore, in this paper, a new task of maximal frequent mutually associated pattern mining is proposed based on the downward closure property of associations. Experimental results show that maximal frequent mutually associated pattern mining is quite a valid method to decrease the number of patterns produced without information loss. Moreover, some techniques have been developed to reduce the execution time. Our performance study shows that our algorithm is much efficient even over very large and dense database.

Acknowledgments. The work is funded by subprogram of China 973 project (NO. 2003CB317006), China NSF program (No. NSFC60503018).

References

1. R. Agrawal, R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In Proc. 1994 Int. Conf. Very Large Databases, pp. 487-499.
2. R. Agrawal, R. Srikant. Mining sequential patterns. In Proc. 1995 Int. Conf. Data Engineering, pp. 3-14.
3. J.Han, G.Dong, Y.Yin. Efficient mining of partial periodic patterns in time series database. In Proc.1999 Int. conf. Data Engineering, pp. 106-115.
4. H. Mannila, H. Toivonen, A. I. Verkamo. Efficient algorithms for discovering association rules. In Proc. AAAI'94 Workshop Knowledge Discovery in Databases, pp. 181-192.
5. H. Mannila, H. Toivonen, A. I. Verkamo. Discover of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1:259-289, 1997.
6. E. Omiecinski.: Alternative interesting measures for mining associations. *IEEE Trans. Knowledge and Data Engineering*, 15: 57-69, 2003.
7. W.-Y. Kim, Y.-K. Lee, J. Han. CCMine: Efficient Mining of Confidence-Closed Correlated Patterns. In Proc. 2004 PAKDD, pp. 569-579.
8. Y.-K. Lee, W.-Y. Kim, Y. D. Cai, J. Han. CoMine: Efficient Mining of Correlated Patterns. In Proc. 2003 Int. Conf. Data Mining, pp.581-584.

Efficiently Mining Mutually and Positively Correlated Patterns

Zhongmei Zhou^{1,2}, Zhaohui Wu¹, Chunshan Wang¹, and Yi Feng¹

¹ College of Computer Science and Technology, Zhejiang University, China

² Department of Computer Science, Zhangzhou Teacher's College, China
{zzm, wzh, cswang, fengyi}@zju.edu.cn

Abstract. Positive correlation mining can find such type of patterns, “the conditional probability that a customer purchasing A is likely to also purchase B is not only great enough, but also significantly greater than the probability that a customer purchases only B .” However, there often exist many independence relationships between items in a correlated pattern due to the definition of a correlated pattern. Therefore, we mine mutually and positively correlated patterns, whose any two sub-patterns are both associated and positively correlated. A new correlation interestingness measure is proposed for rationally evaluating the correlation degree. In order to improve the mining efficiency, we combine association with correlation and use not only the correlation measure but also the association measure in the mining process. Our experimental results show that mutually and positively correlated pattern mining is a good approach to discovering patterns which can reflect both association and positive correlation relationships between items at the same time. Meanwhile, our experimental results show that the mining combined association with correlation is quite a valid method to decrease the execution time.

1 Introduction

Data mining aims to discover useful patterns in large data sets. Although association mining [1] [2] can find many interesting patterns, the following kind of patterns are sometimes meaningless in some applications. “ A and B are associated but not correlated, that is, the conditional probability that a customer purchasing A is likely to also purchase B is great enough, but it is not significantly greater than the probability that a customer purchases only B . For instance, if $P(B)=88\%$, $P(B/A)=90\%$, the sale of A cannot increase the likelihood of the sale of B , even though the conditional probability $P(B/A)=90\%$ is much higher than the given threshold. It is the case that A and B are associated but not correlated.”

To overcome this difficulty, correlation has been adopted as an interestingness measure since most people are interested in not only association-like co-occurrences but also the correlation relationships between items. According to the definition of a correlated pattern [4], although there must exist correlation relationships between items in a correlated pattern, there usually exist many independence relationships between items in a correlated pattern which has more than two items as shown in the example and experimental results. On the other hand, negatively correlated patterns

are misleading on some occasions, especially, on making business decisions. For example, if $P(B)=90\%$ and $P(B/A)=20\%$, the sale of A cannot increase the likelihood of the sale of B , even if the purchase of B is influenced by the purchase of A . It is the case that A and B are negatively correlated. Based on these reasons, in this paper, we mine mutually and positively correlated patterns. From the definition of a mutually and positively correlated pattern, we can see that any two subsets of a mutually and positively correlated pattern are both associated and positively correlated.

Since mutually and positively correlated patterns must have high conditional probabilities, we can discover all mutually and positively correlated patterns in two steps. Firstly, we find patterns which have conditional probabilities high enough. Secondly, we test whether these patterns are mutually and positively correlated or not. Omicinski [7] introduced three alternative interestingness measures, called any-confidence, all-confidence and bond for mining associations. In terms of the definition of all-confidence [7], if a pattern has all-confidence greater than or equal to a given minimum all-confidence, this pattern must have conditional probabilities greater than or equal to the given minimum threshold. Therefore, the measure all-confidence is much suitable for discovering patterns which have conditional probabilities high enough.

A difficulty in this paper is that there are few correlation measures which not only have proper bounds for effectively evaluating the correlation degree of patterns but also are suitable for mining long correlated patterns. The most commonly employed method for correlation mining is that of two-dimensional contingency table analysis of categorical data using the chi-square statistic as a measure of significance. Brin et al. [4] analyzed contingency tables to generate correlation rules. H. Liu et al. [5] analyzed contingency tables to discover unexpected and interesting patterns that have a low lever of support and a high level of confidence. Bing Liu et al. [3] used contingency tables for pruning and summarizing the discovered correlations etc. Although the low chi-squared value (less than the cutoff value, e.g. 3.84 at the 95% significance lever [6]) efficiently indicates that all patterns $AB, \overline{AB}, A\overline{B}, \overline{A}B$ are independent, the high chi-squared value only indicates that at least one of patterns $AB, \overline{AB}, A\overline{B}, \overline{A}B$ is dependent. Therefore, it is possible that AB is independent, i.e. A and B are independent, in spite of the high chi-squared value. Thus, if we do not consider the correlation relationships of the complements of items, the chi-squared value is not reasonable for measuring the dependence degree of A and B .

For other commonly used measures, the measure $P(AB)/P(A)P(B)$ [4] does not have proper bounds. $P(AB) - P(A)P(B) / \sqrt{P(A)P(B)(1 - P(A))(1 - P(B))}$ [8] is not suitable for mining long patterns. In this paper, a new correlation measure corr-confidence is proposed. This measure not only has proper bounds for effectively evaluating the correlation degree of patterns, but also is suitable for mining long patterns.

The remainder of this paper is organized as follows: In Section 2, some related concepts are given and an algorithm is developed for discovering mutually and positively correlated patterns. Section 3 reports our experimental and performance results. Section 4 concludes the paper.

2 Mining Mutually and Positively Correlated Patterns

This section first formalizes some related concepts and then gives an algorithm for efficiently discovering all mutually and positively correlated patterns.

In statistical theory, A_1, A_2, \dots, A_n are **independent** if $\forall k$ and $\forall 1 \leq i_1 < i_2 < \dots < i_k \leq n$,

$$P(A_{i_1} A_{i_2} \dots A_{i_k}) = P(A_{i_1})P(A_{i_2}) \dots P(A_{i_k}) \quad (1)$$

Let pattern $X = \{i_1 i_2 \dots i_n\}$, the new correlation measure corr-confidence (denoted as: ρ) of pattern X is defined as follows using (1):

$$\rho(X) = \frac{P(i_1 i_2, \dots, i_n) - P(i_1)P(i_2) \dots P(i_n)}{P(i_1 i_2, \dots, i_n) + P(i_1)P(i_2) \dots P(i_n)}, \quad (n \geq 1). \quad (2)$$

From (2), we can see that ρ has two bounds, i.e. $-1 \leq \rho \leq 1$.

Let η be a given minimum corr-confidence threshold, a correlated pattern and an independent pattern can be defined using (2).

Definition 1 (a correlated pattern). Pattern A is called a **correlated pattern**, if and only if there exists a pattern B which satisfies $B \subseteq A$ and $|\rho(B)| \geq \eta$.

Definition 2 (an independent pattern). If pattern A is not a correlated pattern, then it is called an **independent pattern**.

By Definition 1 and Definition 2, we can conclude that if pattern A is a correlated pattern, any super pattern of A is a correlated pattern.

We define an associated pattern using the association measure all-confidence [7].

Let $T = \{i_1, i_2, \dots, i_m\}$ be a set of m distinct literals called *items* and D be a set of variable length transactions over T . Each transaction contains a set of items, $\{i_{j_1}, i_{j_2}, \dots, i_{j_k}\}$. A transaction also has an associated unique identifier called *TID*. A pattern X is a subset of T . Let $p(X)$ be a power set of a pattern X . The interestingness measure all-confidence (denoted as α) of pattern X is defined as follows [7]:

$$\alpha = \frac{|\{d \mid d \in D \wedge X \subseteq d\}|}{\text{MAX}\{i \mid \forall l (l \in p(X) \wedge l \neq \phi \wedge l \neq X \wedge i = |\{d \mid d \in D \wedge l \subseteq d\}|\})} \quad (3)$$

Definition 3 (an associated pattern). A pattern is called an **associated pattern** if it has all-confidence greater than or equal to the given minimum all-confidence.

Definition 4 (an associated-correlated pattern). A pattern is called an **associated-correlated pattern** if it is not only an associated pattern but also a correlated pattern.

Definition 5 (a mutually and positively correlated pattern). Pattern A is a mutually and positively correlated pattern, if and only if

$$\rho(E_1 E_2) = P(E_1 E_2) - P(E_1)P(E_2) / P(E_1 E_2) + P(E_1)P(E_2) \geq \eta \quad (4)$$

holds for any non-empty two subsets E_1 and E_2 of pattern A .

By Definition 5, we can easily get three conclusions:

- (1) There exist no independence relationships between items in a mutually and positively correlated pattern.
- (2) All mutually and positively correlated patterns are associated-correlated patterns.
- (3) Any two sub-patterns of a mutually and positively correlated pattern are associated and positively correlated.

The following example is given to illustrate our definitions.

Example. For the transaction database TDB in Table 1, we have $\alpha(AC) = 2/3$ and $\alpha(CE) = 2/3$. Since $\rho(AC) = 1/4$ and $\rho(CE) = 1/19$, if the minimum all-confidence is 0.5 and the minimum corr-confidence is 0.1, then both AC and CE are associated patterns. However, pattern AC is a mutually and positively correlated pattern and pattern CE is an independent pattern. $P(A/C) = 2/3$ and $P(A) = 2/5$, so the sale of C can significantly increase the likelihood of the sale of A . Meanwhile, $P(C/A) = 1, P(C) = 3/5$, the sale of A can also increase the likelihood of the sale of C . Since $P(C/E) = 2/3, P(C) = 3/5$, the sale of E cannot evidently increase the likelihood of the sale of C . It is easy to test that the sale of C cannot evidently increase the likelihood of the sale of E either. Since

$$\frac{P(CDE) - P(CD)P(E)}{P(CDE) + P(CD)P(E)} = \frac{1}{4} \quad \text{and} \quad \frac{P(CDE) - P(C)P(DE)}{P(CDE) + P(C)P(DE)} = \frac{1}{19},$$

CD and E are correlated, C and DE are independent by corr-confidence. Therefore, although CDE is an associated-correlated pattern by Definition 4, both C and E , C and DE , are independent. As a result, there exist independence relationships between items in associated-correlated pattern CDE .

Table 1. transaction database TDB

Transaction id	Items
10	A, B, C
20	C, D, E
30	A, C, D, E
40	D, E
50	B, F

We use the association measure and the correlation measure synchronously in the mining process and develop a level-wise algorithm for discovering all frequent mutually and positively correlated patterns.

Algorithm:

Input: a transaction database TDB , a support threshold ξ , minimum corr-confidence η and minimum all-confidence λ .

Output: the complete set of frequent mutually and positively correlated patterns.

c_k : Candidate patterns of size k

L_k : Frequent associated patterns of size k

M_k : Frequent mutually and positively correlated patterns of size k

$M_1 = \{\text{frequent items}\}$

For ($k=1; M_k \neq \emptyset; k++$) do begin

C_{k+1} = candidates generated from $M_k * M_k$

For each transaction t in database do

increment the count of all candidates in C_{k+1} that are contained in t

L_{k+1} = candidates in C_{k+1} with minimum support and minimum all-confidence

For each pattern l_{k+1} in L_{k+1}

If l_{k+1} is a mutually and positively correlated pattern insert l_{k+1} into M_{k+1}

Return $\cup M_{k+1}$

Remark: In the algorithm, the prune step is performed as follows:

For all patterns $c \in C_{k+1}$ do

For all k -subsets s of c do

If ($s \notin M_k$) delete c from C_{k+1}

3 Experiments

In this section, we report our experimental results. All experiments are performed on two kinds of datasets: 1. A dense dataset, Mushroom characteristic dataset, which consists of 8,124 transactions, each with an average length of 23 items. 2. A sparse real dataset, Traditional Chinese Medicine (TCM) formula dataset, which consists of 4,643 formulas with 21,689 kinds of medicine involved, each with an average length of 10 kinds of medicine. TCM formula dataset is obtained from Information Institute of China Academy of Traditional Chinese Medicine.

Figure 1 shows the execution time of mutually and positively correlated pattern mining and associated-correlated pattern mining respectively on mushroom dataset when the minimum support is 10% and the minimum corr-confidence is 10% with a varied minimum all-confidence. From Figure 1, we can see that when the minimum all-confidence is 30%, the execution time of mutually and positively correlated pattern mining is much less than the execution time of associated-correlated pattern mining.

Because the number of mutually and positively correlated patterns produced is much less than the number of associated-correlated patterns produced as shown in Table 3, Table 4 and Table 5 when the minimum all-confidence is low, the execution time of mutually and positively correlated pattern mining is significantly less than the execution time of associated-correlated pattern mining at a low minimum all-confidence. Consequently, mutually and positively correlated pattern mining not only increases the interestingness of patterns generated, but also saves the execution time, especially when the minimum support and the minimum all-confidence are all low.

Figure 2 (A) shows the runtime of correlated pattern mining with limit the length of patterns and without limit the length of patterns as the minimum support ascends. Figure 2 (B) shows the runtime of correlated pattern mining with limit the length of patterns and without limit the length of patterns as the minimum all-confidence ascends with the minimum support 1% . Figure 2 (A) and (B) indicate that if the maximum length of patterns produced does not exceed 5 , the runtime decreases sharply even if the minimum support and the minimum all-confidence are all low. If the length of patterns produced exceeds 5 , almost all frequent associated patterns are correlated patterns because any super pattern of a correlated is a correlated. In order to efficiently compare the mutually and positively correlated patterns mining with the associated-correlated patterns mining, we put a limit to the maximal length of patterns generated in the experiments of Table 3, Table 4 and Table 5.

Table 3 shows the number of associated-correlated patterns, associated but not correlated patterns, mutually and positively correlated patterns generated on mushroom dataset when the minimum all-confidence increases with the minimum support 1% , minimum corr-confidence 1% , minimum pattern length 2 and maximum pattern length 5 . From Table 3, we can see that for the minimum corr-confidence 1% and the minimum all-confidence 90% , there are seven associated but not correlated patterns and eight associated-correlated patterns in mushroom dataset. We can conclude that not all associated patterns are correlated even if the minimum all-confidence is much high and the minimum corr-confidence is low. Moreover, the number of mutually and positively correlated patterns is significantly less than the number of associated-correlated patterns on mushroom dataset. Therefore, we can conclude that there exist many independence relationships between items in an associated-correlated pattern. Since any two sub-patterns of a mutually and positively correlated pattern are both associated and positively correlated, only mutually and positively correlated patterns mining can discover both association and positive correlation relationships between any items at the same time.

Table 4 and Table 5 show the number of associated-correlated patterns, associated but not correlated patterns, mutually and positively correlated patterns generated in mushroom dataset and TCM dataset respectively as the minimum corr-confidence varies. To our surprise, when the minimum corr-confidence is 5% , there are only 0.28% associated but not correlated patterns of all associated patterns in TCM dataset, while there are 16% associated but not correlated patterns of all associated patterns in mushroom dataset. Therefore, we can conclude that there are more correlations in TCM dataset than in mushroom dataset. From Table 4 and Table 5, we can see that the number of mutually and positively correlated patterns is nearer to the number of associated-correlated patterns on TCM dataset than on mushroom dataset. It also indicates that there are more correlations on TCM dataset than on mushroom dataset.

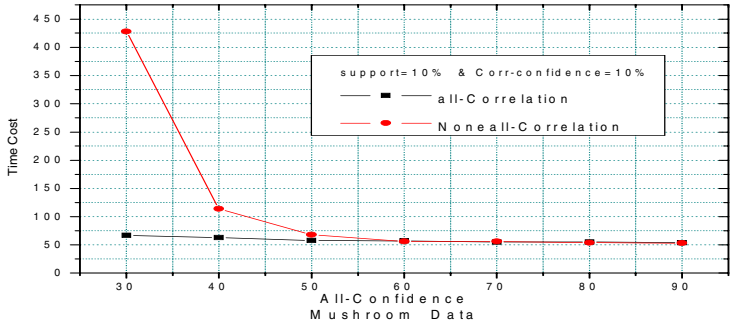


Fig. 1. Runtime of mushroom dataset

Table 3. Num. in mushroom data (min_sup 1%, min_len 2, max_len 5, c_conf 1%)

All-connfidence	Independent	correlated	Mutually correlated
30	112	3678	1051
40	90	1012	381
50	61	279	144
60	31	83	56
70	12	36	21
80	12	16	9
90	7	8	5

Table 4. Num. in TCM dataset (support1%, min_len 2, max_len 5, all_conf 10%)

Corr-confidence	independent	correlated	Mutually correlated
5	3	1058	953
10	7	1054	950
15	16	1045	940
20	31	1030	923
25	55	1006	907
30	76	985	892
35	112	949	836
40	160	901	775

Table 5. Num.of mushroom data (min_sup 1%, min_len2 max_len5, all_conf 30%)

Corr-confidence	independent	correlated	Mutually correlated
5	603	3187	810
10	1066	2724	561
15	1367	2423	406
20	1613	2177	275
25	1875	1915	209
30	2100	1690	177
35	2262	1528	131
40	2423	1367	116

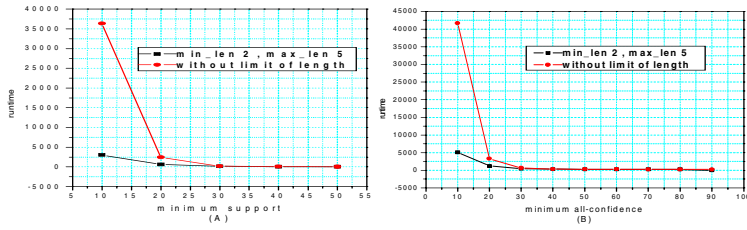


Fig. 2. Runtime of mushroom dataset

4 Conclusions

Mutually and positively correlated pattern mining can find patterns that are extraordinary useful for making business decisions, because any two sub-patterns of a mutually and positively correlated pattern are both associated and positively correlated. In this paper, a new interestingness measure for correlation mining is proposed, which is not only suitable for mining long correlated pattern, but also more rational and easier to control than the chi-squared test and other commonly used measures. Experimental results show that the algorithm developed is much efficient and effective.

Acknowledgments

The work is funded by subprogram of China 973 project (NO. 2003CB317006), China NSF program (No. NSFC60503018)

References

1. R. Agrawal, T. Imielinski, A. Swami. Mining Association Rules Between Sets of Items in Large Databases. In Proc. 1993 ACM SIGMOD Int. Conf. Management of Data (SIGMOD'93), pp. 207-216.
2. R. Agrawal, R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In Proc. 1994 VLDB Int. Conf. Very Large Databases (VLDB'94), pp. 487-499.
3. Bing Liu, Wynne Hsu, Yiming Ma. Pruning and Summarizing the Discovered Association. In Proc. 1999 ACM SIGKDD Int. Conf. Knowledge Discovery in Databases (KDD'99), pp. 15-18.
4. S. Brin, R. Motwani, C. Silverstein. Beyond Market Basket: Generalizing Association Rules to Correlations. In Proc. 1997 ACM SIGMOD Int. Conf. Management of Data (SIGMOD'97), pp. 265-276.
5. H. Liu, H. Lu, L. Feng, F. Hussain. Efficient Search of Reliable Exceptions. In Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'99), pp. 194-203.
6. F. Mills. Statistical Methods. Pitman, 1955.
7. E. Omiecinski. Alternative interesting measures for mining associations. *IEEE Trans. Knowledge and Data Engineering*, 2003(15): 57-69.
8. H. T. Reynolds. The Analysis of Cross-Classifications. The Free Press, New York, 1977.

ComEnVprs: A Novel Approach for Inducing Decision Tree Classifiers

Shuqin Wang¹, Jinmao Wei^{2,3}, Junping You², and Dayou Liu³

¹ School of Mathematics & Statistics, Northeast Normal University, Jilin, 130024, China
{wangsq562, weijm374}@nenu.edu.cn

² Institute of Computational Intelligence, Northeast Normal University, Jilin, 130024 China

³ Open Symbol Computation and Knowledge Engineering Laboratory of State Education,
Jilin University, Jilin 130024, China

Abstract. This paper presents a new approach for inducing decision trees by combining information entropy criteria with VPRS based methods. From the angle of rough set theory, when inducing decision trees, entropy based methods emphasize the effect of class distribution. Whereas the rough set based approaches emphasize the effect of certainty. The presented approach takes the advantages of both criteria for inducing decision trees. Comparisons between the presented approach and the fundamental information entropy based method on some data sets from the UCI Machine Learning Repository are also reported.

1 Introduction

From the earliest work CLS [1] to nowadays decision tree systems in data mining, decision tree learning is always one of the most widely used and practical methods for inductive inference [2], [3], [4]. When inducing a decision tree, we have to choose appropriate condition attributes as the tree nodes. Several criteria are available for selecting attributes, such as the information entropy based methods [5], Bayesian networks [6], gini index methods [7], etc.

Rough set theory, proposed by Poland mathematician Pawlak, is a new mathematic tool to deal with vagueness and uncertainty [8]. It has been widely used in many fields such as machine learning, data mining and pattern recognition [9], [10], [11], etc. In [12], the authors presented a new approach based on the rough set theory for inducing decision trees. In order to enhance its ability of tolerating possible noises in real data sets, we introduce the Variable Precision Rough Set Model (VPRSM) into the initial approach for inducing classifiers with higher generalization, which is an important problem to be handled when inducing decision trees [5], [13], [14], [15]. By further comparison, we find that, information entropy based methods emphasize the effect of overall class distribution, whereas rough set based approaches emphasize the effect of certainty, which has been similarly discussed in [16]. Consequently, we present a new approach for inducing decision trees by combining the entropy and rough set based approaches together to take the advantages of both criteria.

2 Rough Set Based Approach for Inducing Decision Trees

Given a knowledge representation system: $S = (U, Q, V, \rho)$. U is the field or universe to be learned. Q denotes the set of attributes. It is usually divided into two subsets C and D , which denote the sets of condition and decision attributes respectively.

$\rho : U \times Q \rightarrow V$, where $V = \bigcup_{a \in Q} V_a$ and V_a is the domain of attribute $a \in Q$.

For any subset G of C or D , an equivalence relation \tilde{G} on U can be defined such that a partition of U induced by it can be obtained. Denote the partition as $G^* = \{G_1, G_2, \dots, G_n\}$, where G_i is an equivalence class of \tilde{G} .

Variable Precision Rough Set Model (VPRSM)[17] is an expansion to the basic rough set model, which allows some misclassification when classifying instances. The introduction of a limit β to classification error gives it the power to consummate the theory of approximation space.

Definition 1 [17]. Assume U denotes the universe to be learned. X and Y denote the non-empty subsets of U . Let:

$$c(X, Y) = \begin{cases} 1 - \frac{|X \cap Y|}{|X|}, & |X| > 0, \\ 0, & |X| = 0. \end{cases} \quad (1)$$

$c(X, Y)$ is the relative classification error of the set X with respect to set Y .

Suppose (U, \tilde{R}) is an approximation space, $R^* = \{E_1, E_2, \dots, E_n\}$ denotes the set containing the equivalence classes in \tilde{R} . For any subset $X \subseteq U$, the β lower approximation of X with respect to \tilde{R} is defined as:

$$\underline{R}_\beta X = \bigcup \{E_i \in R^* \mid c(E_i, X) \leq \beta\}. \quad (2)$$

The β upper approximation of X with respect to \tilde{R} is defined as:

$$\overline{R}_\beta X = \bigcup \{E_i \in R^* \mid c(E_i, X) < 1 - \beta\}. \quad (3)$$

In the process of decision tree construction, we reunite the rough sets into two sets: one is the set of objects that can be definitely assigned class labels, the other is the set of objects that can't be certainly assigned class labels.

Definition 2. Let $A \subseteq C$, $B \subseteq D$. $A^* = \{X_1, X_2, \dots, X_n\}$ and $B^* = \{Y_1, Y_2, \dots, Y_m\}$ denote the partitions of U induced by equivalence relation \tilde{A} and \tilde{B} respectively. The variable precision explicit region is defined as:

$$Exp_{A\beta}(B^*) = \bigcup_{Y_i \in B^*} \underline{A}_\beta(Y_i). \quad (4)$$

where $\underline{A}_\beta(Y_i)$ is the β lower approximation of Y_i with respect to \tilde{A} .

Definition 3. Let $A \subseteq C, B \subseteq D$. $A^* = \{X_1, X_2, \dots, X_n\}$ and $B^* = \{Y_1, Y_2, \dots, Y_m\}$ denote the partitions of U induced by equivalence relation \tilde{A} and \tilde{B} respectively. The variable precision implicit region is defined as:

$$Imp_{A\beta}(B^*) = \bigcup_{Y_i \in B^*} (\overline{A}_\beta(Y_i) - \underline{A}_\beta(Y_i)). \tag{5}$$

where $\underline{A}_\beta(Y_i)$ is the β lower approximation and $\overline{A}_\beta(Y_i)$ is the β upper approximation of Y_i with respect to \tilde{A} .

The initial idea of the rough set based approaches for inducing decision trees lies in the following process:

From an original data set to the final constructed decision tree, the learned knowledge about the system tends to gradually become more and more explicit, consequently one will gradually learn more and more about the system. Hence, one attribute will be chosen if the explicit region of it is greater than that of all the others.

In the approach, when we evaluate a possible condition attribute, the data set is partitioned into two parts: one is the explicit region, *Exp* in short, in which each object has the same class label as the other objects if the values of their condition attributes are identical; the other is the implicit region, *Imp* in short. After partition we can obtain the sizes of these two parts. Similarly, we can obtain the explicit regions and implicit regions and their sizes of all other condition attributes. We choose the attribute with the greatest explicit region as the branch node and split the data set under consideration subject to the different values of the attribute. Consequently, we will learn as much knowledge as possible conveyed by the explicit region.

3 Comparisons Between Rough Set Based Approaches and Information Entropy Based Approaches

In the fundamental entropy based method, the initial idea is to observe the information gain (Info-Gain) when a data set is split by the possible values of condition attributes. Info-Gain is defined [5], [18] as:

$$Info-Gain(A, U) = Info(U) - Info(A, U)$$

Where U is the set of objects, A is a condition attribute.

If a set U is partitioned into disjoint exhaustive classes $\{Y_1, Y_2, \dots, Y_k\}$ on the basis of the value of decision attribute, the information needed to identify the class of an element of U is

$$Info(U) = I(P) = - \sum_{i=1}^k p_i \log(p_i).$$

P is the probability distribution of the partition $\{Y_1, Y_2, \dots, Y_k\}$, i.e.

$$P = \left(\frac{|Y_1|}{|U|}, \frac{|Y_2|}{|U|}, \dots, \frac{|Y_k|}{|U|} \right), \quad p_i = \frac{|Y_i|}{|U|}.$$

If a condition attribute has the greatest Info-Gain, this attribute will be chosen to split the data set. As mentioned above, when evaluating a condition attribute A_n , data

set U is split into two parts, Exp and Imp . In the fundamental entropy based method, the info-gain can be calculated as:

$$\begin{aligned} \text{Info-Gain}(A_n, U) &= \text{Info}(U) - \text{Info}(A_n, U) \\ &= \text{Info}(U) - (\text{Info}(A_n, Exp) + \text{Info}(A_n, Imp)) = \text{Info}(U) - \text{Info}(A_n, Imp). \end{aligned}$$

Here $\text{Info}(A_n, Exp)$ is zero. This implies that Exp doesn't make contribution to the information gain, or at least, Exp doesn't make contribution directly. In practice, if $\text{Info}(A_n, Imp)$ is the smallest, attribute A_n will be chosen. From this point, the entropy based methods pay attention to the distribution of classes on Imp . In contrast, the rough set based approaches pay attention to the size of Exp . If the explicit region Exp of attribute A_n , for example, is the greatest, it will be chosen at last.

In the basic rough set based approaches, no care is cast over Imp no matter what it may convey. In applications, even a small perturbation may totally reverse the classification of objects from explicit regions into implicit regions. Therefore, it is necessary to take a look at the objects within $Imps$, and reclassify the misclassified objects in the $Imps$ into the $Exps$ under the misclassification error limit β . The VPRSM based approach is presented to solve such problems.

4 ComEnVprs: A Combined Approach for Inducing Decision Trees

After some possible misclassified objects in implicit region are reclassified into explicit region, the final decision tree is likely to have the ability to tolerate some noises within the data. However, in the rough set based approaches we still neglect the effect of class distribution. In order to take both aspects into consideration, we propose a combined approach, and note it as ComEnVprs. In ComEnVprs, when evaluating a condition attribute, both explicit region and implicit region are calculated. Denote the size of explicit region as S . Since the information entropy of explicit region is zero, the information entropy of implicit region is denoted as E . In the process of inducing decision trees, we seek to find a condition attribute to partition data set to obtain as greater explicit region as possible and as less information entropy as possible. Hence, in ComEnVprs we use S/E to evaluate a candidate condition attribute. If S/E corresponding to a condition attribute is greater than that of the other condition attributes, this condition attribute is chosen as the current branch node.

In ComEnVprs, it involves three cases when calculating S/E :

$$S/E = \begin{cases} 1/E & S=0, \\ S/\varepsilon & E=0, \varepsilon \text{ is a relatively small value} \\ S/E & S \neq 0, E \neq 0. \end{cases}$$

If $S=0$, it implies that the size of explicit region is zero. Hence, we only need to consider the effect of class distribution. If the information entropy corresponding to a condition attribute is smaller than that of the other's, this condition attribute will possibly be chosen. If $E=0$, this implies all objects can be classified unambiguously. In this case, the corresponding attribute should definitely be chosen as the branch node. If $S \neq 0, E \neq 0$, we calculate S/E . From the definitions of explicit and implicit

regions, it is not difficult to notice that S and E could not be zero simultaneously. We also have $0 \leq |S| \leq S_d$, where S_d is the size or cardinality of the data set under consideration. If $|S| = S_d$, then $E=0$. Hence ε should be assigned a proper value. In ComEnVprs, ε is the reciprocal of the size of the original data set.

5 Comparisons on Some Real Data Sets

We utilize some data sets from the UCI Machine Learning Repository to test the presented approach, and compare the decision trees by ComEnVprs with that obtained by the fundamental entropy based method (ENPrePrune). Both the names of all data sets and the results are shown in Table 1.

We use 16 kinds of data set from the UCI Machine Learning Repository. With respect to data sets "balloons", there are three different such kinds of data set, we label them as "balloon1, 2, 3". As to data sets "monks", there are three kinds of monk problem, they are labeled as "1,2,3". For each problem, there are two different kinds of data set, one for test, we label it as "MonksD"; the other for training is labeled as "MonksT". There are two kinds of solar flare data set, labeled as "flare1, 2", for each data set, there are three decision attributes, they are dealt with separately, and labeled by adhering "1,2,3" to the end of each data set.

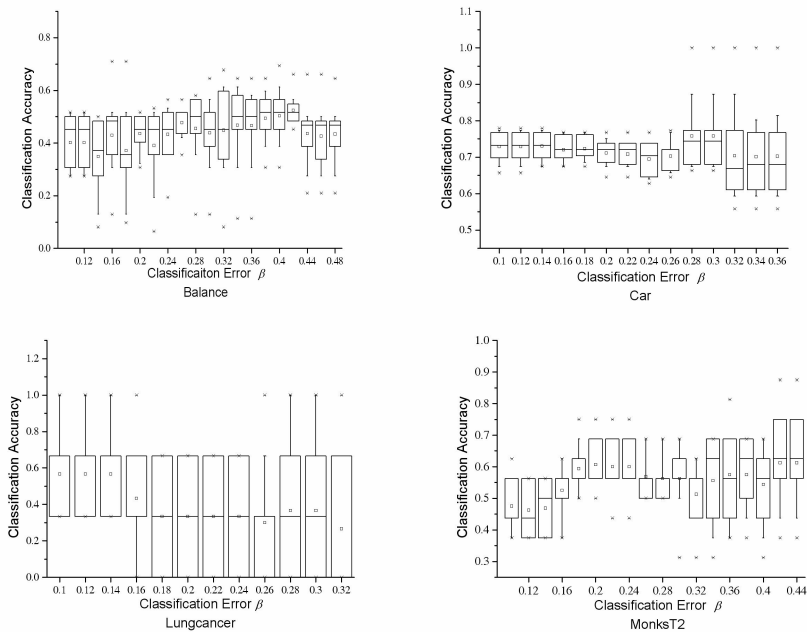


Fig. 1. Distributions of classification accuracy for different classification errors

In the table, "NC", "NT" indicate the number of condition attributes, the number of tuples or instances respectively. "NCV" indicates the number of continuous valued condition attributes. "ENPrePrune", " ComEnVprs " indicate the entropy based approach and the proposed approach respectively. "Leaves" indicates the number of the leaf nodes of the constructed decision trees.

Table 1. Comparison of decision trees

	ComEnVprs							ENPrePrune			
	NC	NCV	NT	Leaves	Tree	Accu	β	Leaves	Tree	Accu	β
Adult	14	6	16281	21	23	0.792199	0.44	97.5	139.10	0.827273	0.44
Balance	4	0	625	20.3	25.5	0.524194	0.42	24.9	31.1	0.467742	0.42
Baloon1	4	0	20	3	4	1	0	3	4	1	0
Baloon2	4	0	20	6.7	12.4	0.7	0	6.7	11.4	0.7	0
Baloon3	4	0	16	3	4	1	0	3	4	1	0
Car	6	0	1728	102	144.3	0.75814	0.28	15.9	21.9	0.744186	0.48
Cmc	9	2	1473	90.6	180.20	0.263946	0.48	223.8	504.70	0.257143	0.4
Heart disease	13	5	270	5.5	8.4	0.744444	0.26	47.2	97.3	0.744444	0.18
Iris	4	4	150	4	5	0.96	0.12	4.4	6	0.953333	0.1
Len	4	0	24	3.4	5.8	0.85	0.34	3	5	0.9	0.42
LungCancer	56	0	32	15.2	22.9	0.566667	0.1	6.3	9.2	0.3	0.42
MonksD1	7	0	432	27.8	40.7	0.972093	0.01	26.8	39.2	0.851163	0.1
MonksT1	7	0	432	4	5	0.75	0.44	4	5	0.75	0.44
MonksD2	7	0	432	3.3	4.3	0.669767	0.38	3.5	4.5	0.669767	0.4
MonksT2	7	0	432	2.7	3.7	0.6125	0.42	8.5	12.3	0.5625	0.42
MonksD3	7	0	432	12	17	1	0	12.8	16.8	1	0.1
MonksT3	7	0	432	17	25.5	0.875	0.1	17.3	26	0.883333	0.1
Mushroom	22	0	8124	23.8	28.8	1	0.01	23.8	27.8	1	0.1
Nursery	8	0	12960	18.1	25	0.77338	0.34	16.3	22	0.77338	0.36
Shuttle	6	0	15	6.8	8.8	0.7	0.26	5	6.1	0.7	0.34
Solarflare11	10	0	323	5.6	6.6	0.8875	0.24	6.2	7.6	0.8875	0.34
Solarflare12	10	0	323	6	7	0.903125	0.32	4.5	6	0.890625	0.48
Solarflare13	10	0	323	5.6	6.6	0.978125	0.18	5.6	6.6	0.978125	0.18
Solarflare21	10	0	1066	1.2	2.2	0.828302	0.18	14.8	27.8	0.816981	0.48
Solarflare22	10	0	1066	6	7	0.966038	0.14	5.2	6.2	0.966038	0.26
Solarflare23	10	0	1066	4	5	0.995283	0.16	4	5	0.995283	0.16
Tictactoe	9	0	958	3	4	0.527368	0.46	3.2	4.3	0.546316	0.48
Housevote	16	0	435	5.4	7.6	0.955814	0.2	3	4	0.955814	0.32

"Tree" indicates the complexity of the decision trees. The number in this column is the node counts of the whole trees. In experiments, ten fold cross validation was conducted on all data sets to calculate the classification accuracy of the two methods. For each data set, we first divided it into ten subsets, then for each subset we used it as test set, and the rest nine sets as train set, this resulted in a value of accuracy. That is to say, we can obtain ten results with respect to each data set. In the table, "Accu" indicates the average accuracy of the ten values of accuracy with respect to a data set and is assigned as the accuracy of the corresponding decision tree.

In ENPrePrune, all possible attributes were evaluated by calculating their Info-Gains. Attributes with the greatest Info-Gain were chosen to split data sets.

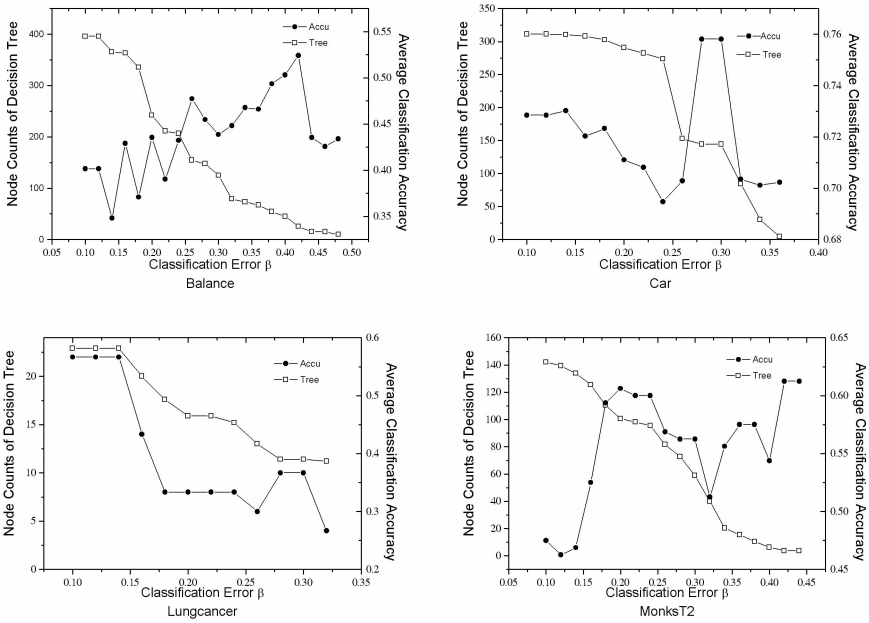


Fig. 2. Distributions of node counts and average classification accuracy of decision trees for different classification errors on data sets Balance, Car, Lungcancer, and MonksT2

In Fig.1 and 2, the reported results are for four data sets (Balance, Car, Lungcancer and MonksT2) that the decision trees are induced by the presented approach. We present the distributions of classification accuracy for different classification errors on the four data sets and the distributions of node counts and average classification accuracy of decision trees for different limits of classification error on the same four data sets.

The box charts of the classification accuracy for four data sets are presented in Fig.1. In the figure, the horizontal lines in the box denote the 25th, 50th, and 75th percentile values. The square symbol in the box denotes the mean of the column of data.

The plots of the node counts and average classification accuracy (Accu) of the whole trees for the four data sets are presented in Fig.2. Lines marked with dots denote the average classification accuracy of the trees; lines marked with squares denote the node counts of the trees.

From Table 1, we can see that ComEnVprs shows to be better than or as good as its counterpart on most data sets. Fig.1 shows that suitable limits of classification error can be found for the problems. In Fig.2, the lines of the node counts of the decision trees decline when the classification errors increase from a small value to a value not over 0.5. In fact, in the experiments, this is true to all data sets, that is to say, the size of decision trees will reduce gradually when the limits of classification error increase. The appearances of the lines of average classification accuracy turn to be different with respect to different data sets. However, with what is shown in Fig.1, we can find a suitable limit β of classification error for each data set, the result is presented in Table 1.

From the results, we can see that the sizes of decision trees decline as classification error increases. That is to say, pruning embeds in the process of inducing decision trees. However, such pruning issues for enhancement of generalization ability need further investigation. For the data sets with conflicting tuples or noise, classification was made according to majority in all methods. In the data sets with numerical attribute(s), we discretized the attribute value(s) equally (equal interval) before constructing the decision trees.

6 Concluding Remarks

In the paper, we review the fundamental idea of the rough set and variable precision rough set based approaches for inducing decision trees, and analyze the differences between the rough set based approaches and the information entropy based approaches, then present the combined method in order to take the advantages of both of them. Experiments on some data sets from the UCI Machine Learning Repository show that ComEnVprs is better than or as good as the fundamental entropy based method in classification accuracy on most data sets. By heuristically finding an appropriate limit of classification error in the training stage, the presented approach can be utilized to achieve high generalization ability of decision trees. Pruning embeds in ComEnVprs during the process of inducing decision trees, though such problems as how to enhance the generalization abilities need further investigation. ComEnVprs is a simple approach for inducing decision trees, it is certainly easy to be integrated with other scalable decision tree inducers [19], [20] in applications.

References

1. Hunt, E. B., Marin, J., Stone, P. J.: Experiments in Induction. New York: Academic Press (1966)
2. Fayyad, U. M., Weir, N., Djorgovski, S.: SKICAT: A machine learning system for automated cataloging of large scale sky surveys. Proc. the Tenth International Conference on Machine Learning, 112-119. Amherst. MA: Morgan Kaufmann (1993)

3. Michalski, R. S., Carbonell, J. G., Mitchell, T. M.: *Machine Learning-An Artificial Intelligence Approach*. Springer-Verlag, printed in Germany (1983)
4. Chen, S. C., Shyu, M. L., Chen, M., Zhang, C. C.: *A Decision Tree-based Multimodal Data Mining Framework for Soccer Goal Detection*. IEEE International Conference on Multimedia and Expo, June 27 - June 30, 2004, Taipei, Taiwan, R.O.C
5. Quinlan, J. R.: *Introduction of Decision Trees*. *Machine Learning*, (1986) Vol. 3, 81-106
6. Cheng, J., Bell, D.: *Learning bayesian networks from data: an efficient approach based on information theory*. Proc. of the sixth ACM International Conference on Information and Knowledge Management, (1997) 325-331
7. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: *Classification and regression trees*. Technical report, Wadsworth International, Monterey, CA (1984)
8. Pawlak, Z.: *Rough sets*. *International Journal of Computer and Information Science*, (1982) 11, 341-356
9. Jerzy, W., GrZymala-Busse, J. W. Ziarko, W.: *Data mining and rough set theory*. *Communications of the ACM*. (2000) 43(4) 108-109
10. Pawlak, Z.: *Rough set approach to multi-attribute decision analysis*. *European Journal of Operational Research*, (1994) 72 (3) 443-459
11. Pawlak, Z., Wang, S. K. M., Ziarko, W.: "Rough sets: probabilistic versus deterministic approach". *Int. J. Man-Machine Studies*, (1988) 29-1, 81-95
12. Wei, J. M.: *Rough Set Based Approach to Selection of Node*. *International Journal of Computational Cognition*. (2003) 1(2) 25-40.
13. Mingers, J.: *An empirical comparison of pruning methods for decision-tree induction*. *Machine Learning*, (1989) 4(2) 319-342
14. Quinlan, J. R., Rivest, R.: *Inferring decision trees using the minimum description length principle*. *Information and Computation*. (1989) 80(3) 227-248
15. Zbigniew, W. Ras, Zemanekova, M.: *Imprecise Concept Learning within a Growing Language*(314-319). Proc. the sixth inter. workshop on Machine learning, Ithaca, New York, United States (1989)
16. Fernando Berzal, Juan Carlos Cubero, Fernando Cuenca & María José Martín-Bautista: *On the quest for easy-to-understand splitting rules*. *Data & Knowledge Engineering*, (2003) Vol. 44, No. 1, pp. 31-48.
17. Ziarko, W.: *Variable precision rough set model*. *Journal of Computer and System Sciences* (1993) 46(1) 39-59
18. Quinlan, J. R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann(1993)
19. Gehrke, J, Ramakrishnan, R.&Ganti, V.: *RainForest - A Framework for Fast Decision Tree Construction of Large Datasets*. *Data Mining and Knowledge Discovery*, (2000) 4(2/3) 127-162.
20. Rastogi, R. & Shim, K.: *PUBLIC: A Decision Tree Classifier that integrates building and pruning*. *Data Mining and Knowledge Discovery*, (2000) 4(4) 315-344.

Towards a Rough Classification of Business Travelers

Rob Law, Thomas Bauer, Karin Weber, and Tony Tse

School of Hotel & Tourism Management, The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
hmroblaw@polyu.edu.hk

Abstract. The significant economic contributions of the fast growing tourism industry have drawn worldwide attention on understanding the behavioral and demographic patterns of visitors. This research makes an attempt to develop a rough sets based model that can capture the essential information from business travelers, a segment of the market that to date has been entirely overlooked by academic researchers in data mining. Utilizing the primary data collected from an Omnibus survey carried out in Hong Kong in late 2005, experimental findings showed that the induced decision rules could classify 82% of the cases in the testing set and 41% of the classified cases were correctly estimated. Most importantly, there was no statistically significant difference between the estimated values and actual values.

1 Introduction

In Hong Kong, a Special Administrative Region of the People's Republic of China (HKSAR), the tourism industry plays a major role as one of the key industries to support the city. The tourism receipts generated by tourists permeate to different sectors in society, which in turn, significantly contribute to the growth of the local economy. Various studies have shown the economic contributions of international travelers in general, and business travelers in particular, to a destination [1, 7]. During 2004, business visitors in Hong Kong (including those who visited to attend meetings, conferences, and exhibitions) stayed 40% longer than leisure visitors and spent 22% more than all other overnight visitors [4]. Likewise, statistical data from the Hong Kong Tourism Board (HKTB) showed that the number of business events increased by 86% from 2003 to 2004, and that the corresponding percentage increase for the number of visitors attending these events was 59.1% [3]. These figures highlight the importance of the business traveler market and its contribution to the economy. As a result, the high-yield business traveler segment has attracted the attention of policy makers, practitioners, and, to a much lesser extent, academic researchers in general and in particular researchers in data mining.

Despite its significance, few attempts, if any, have been made to understand the behavioral patterns of business travelers. The HKTB data only showed the number and percentages of business visitors from different major source markets but no effort has been made to mine the profile of these business travelers [4]. Hence, policy makers and practitioners have no way to distinguish between different characteristics of these important visitors. Yet, such a lack of understanding limits their ability to cater

to the needs of, and market to, this segment. In the existing data mining and tourism literature, the study of data mining of business travelers has been entirely overlooked by academic researchers. In other words the demographic and trip profiles of this high yield group of visitors remain largely unknown. Taking note of the absence of prior studies on the mining of the profiles of business travelers, this research seeks to fill the void by developing a rough sets model to capture useful information from business travelers to Hong Kong using primary data collected in a recent survey.

Having introduced the research background, the following section provides an overview of the general concept of the rough sets theory and presents the rough sets based model developed in this research to mine the useful information from a set of raw tourism data. Next, there is a methodology section which describes the data collection and implementation processes. The subsequent section analyzes the decision rules induced from the model calibration set, and their quality of estimation (forecasting) using the model testing set. The last section summarizes this research, and offers suggestions for future research.

2 Data Mining of Business Travelers

The rough sets approach deals with classification and knowledge discovery of uncertain, vague, or imprecise information which is usually presented in a form of data acquired from the real world. In general, the concept of the rough sets theory is presented in the approximation space of upper and lower boundaries of a set of knowledge [8]. The approximation space is itself a classification of the domain of interest which is further divided into disjoint categories. To present the notion formally, the classification is our knowledge about the domain concept. In other words, the concept is modeled as the ability to characterize all classes of the classification in terms of the characteristics of the entries that are part of the domain. Furthermore, objects of the same category are not discernable, meaning that their status with respect to a subset of the knowledge domain cannot be clearly distinguished. This quality turns into the definition of the lower and upper approximations of a set. The lower approximation relates to the objects that are known to be as elements of the subset with interest. In contrast, the upper approximation is for the objects that could be the elements of the subset. A rough set is a subset defined between the lower and upper approximations. The rough sets approach has been applied in different fields. For instance, Grzymala-Busse, Goodwin, and Zhang [2] showed the application of the rough sets approach in medical applications, global warming, nursing, environmental protection, natural language, and data transmission. Similarly, Tanaka and Maeda [10] demonstrated the data reduction capability in financial management. The following paragraphs formally present the rough sets notion that is developed to model the business travelers.

2.1 Approximate Classification

Let x be a set of objects which represents a concept of universe U and $x \subseteq U$. x is then put into the database (or knowledge base) $K = \langle U, R \rangle$ where R represents the elementary sets or equivalent classes. The P-lower approximation of x in $K = \langle U, R \rangle$ is defined as:

$$\underline{P}x = \cup\{y \in U \mid R: y \subseteq x\}. \quad (1)$$

In other words, $\underline{P}x$ is the set which consists of all objects that can be classified as elements of x , in the knowledge P .

In addition, the P-upper approximation of x in $K=\langle U,R \rangle$ is defined as:

$$\overline{P}x = \cup\{y \in U \mid R: y \cap x \neq \emptyset\}. \quad (2)$$

Based on the above lower and upper approximations, three regions are further introduced: i) The P-boundary region of x in $K=\langle U,R \rangle$ is $BNp(x) = \overline{P}x - \underline{P}x$, and ii) the P-positive region of x in $K=\langle U,R \rangle$ is $POSp(x) = \underline{P}x$, and iii) the P-negative region of x is

$$NEGp(x) = U - \overline{P}x. \quad (3)$$

2.2 Information Table (IT)

An *IT* is defined as a special database in which $IT = \{U,A,V,f\}$ such that U is a universe of objects in the database. Features or attributes represent the characteristics of each object. Each of these attributes can be represented by a finite number of values. A is a finite set of attributes in which $A = \{C,D\}$ where C is a set of condition attributes (variables) and D is a set of decision attributes (variables). V is the union of attribute domains, such that $V = \cup \mathcal{N}_a$ for $a \in A$, where V_a is the domain of the attribute a . Lastly, f is an information function which associates a unique value of each attribute with every object that belongs to U .

Simply put, an *IT* can appear in a table form in which rows correspond to objects that are represented by attribute values. In the case of business travelers, each row represents the demographic and trip profile of a traveler; where the decision attribute and condition attributes are spending per night and other attributes respectively.

2.3 Information Reduction

Intuitively, not all information in an attribute-value *IT* is important. More precisely, unique judgments on an *IT* can be made by omitting some attributes. To eliminate the superfluous attributes, the concepts of *reduct* and *core* sets of attributes are introduced. For every $R, R \subseteq C$, if $POS_C(D) \neq POS_R(D)$, C is defined as independent with respect to D . Otherwise, C is defined as dependent with respect to D . Discovering attribute independencies is claimed to have the primary importance in the rough sets approach in relationship modeling [6,9]. In view of the limitations of original rough sets theory which only deals with fully correct or certain classification, Ziarko [11,12,13] further proposed a generalized rough sets model by specifying attribute precision values which can handle a tolerable degree of uncertainty or misclassification. In such a generalized approach, the positive and negative regions are defined as the areas where approximate classification with an error level less than a predefined

level is possible. As such, the boundary region becomes the complement of the target event with an error level that cannot be smaller than a predefined value.

The set S is defined to be a *reduct* of C , denoted by $RED_D(C)$ if S is an independent subset of C with respect to D and $POS_S(D) = POS_C(D)$. In other words, the reduct $RED_D(C)$ is the minimal subsets of C which generates the same classification of objects into an equivalence class of knowledge D as the whole knowledge C . For a set of condition and decision attributes, C and D , $a \in C$ is defined as dispensable in D if $POS_C(D) = POS_B(D)$; where $B = C - \{a\}$. Otherwise, attribute a is said to be indispensable in D .

A *core* is defined to be the set of all indispensable attributes in P . That is, $CORE_D(C) = \{a \in C: POS_C(D) \neq POS_B(D)\}$. The *core* is a collection of the most significant attributes in an IT. In other words, one cannot eliminate a *core* attribute without destroying the ability to classify objects into an equivalence class D . However, the *core* set could be empty. Rules can then be induced from the reduced IT which contains the *reduct* set. The procedure for capturing decision rules from a set of raw data is known as induction [6, 13], which is shown in the next subsection.

2.4 Rules Induction

For an arbitrary set of objects (O_b), a set of object feature (Att), a set of attribute values (V), and a function $f: O_b \times Att \rightarrow V$, such that each element in O_b is described by the values of its associated attributes. The equivalence relation $R(A)$, $\forall A \subset Att$, and given two objects o_1 and o_2 , where $o_1, o_2 \in O_b$, if $o_1 R(a) o_2 \iff f(o_1, a) = f(o_2, a)$, $\forall a \in A$, o_1 and o_2 are indiscernible with respect to attributes in A . This relation is then used to partition the universe of objects into different equivalent classes, $\{e_1, e_2, e_3, \dots\} = R$. The pair (O_b, R) forms an approximation space with which different subsets of O_b are approximated. By applying the various concepts defined in this section, inductive learning systems in a form of decision rules are generated. In general, the description of an object in the positive region implies a positive decision class; whereas the description of an object in the negative region implies a negative decision class. The description of an object in the boundary region implies a probabilistically positive decision class.

3 Methodology

Data were collected in an Omnibus Survey carried out by the School of Hotel & Tourism Management at the Hong Kong Polytechnic University during the period from October 3, 2005 to October 22, 2005. During this period, a total of 1,282 non-transit visitors from seven major tourist generating regions were interviewed face-to-face in the restricted departure lounge of the Hong Kong International Airport. Following the practice of the Omnibus Survey [5], the questionnaire was developed in English and then translated into Chinese. The final version of the questionnaire was pilot-tested in September 2005 to ensure that questions were clearly understood. The questionnaire consisted of a common set of questions for demographic and trip profile. These demographic and trip profile data were utilized in this research for classification of business travelers. Among the 1,282 respondents, 303 identified

themselves as business travelers and provided usable demographic and trip profile data. Table 1 lists the profile of these business travelers. All attributes (variables) in Table 1 were grouped as condition attributes whereas expenses per night excluding accommodation and airfare were used as the decision attribute. An equal percentile approach was adopted to split the values in the decision attribute into three categories of High (H), Medium (M), and Low (L).

A software system was implemented that induces decision rules for the data obtained from business travelers. By randomly selecting 80% (N=243) of the cases for model calibration and the remaining ones (N=60) for model testing, the accuracy of

Table 1. Induced Decision Rules

Number	Rule
1	If [First Visit to Hong Kong = Yes] and [Region of Residence = (Australia or Western Europe or Singapore or China or Malaysia or United States)] and [$4 \leq \text{Length of Stay in Hong Kong} \leq 6$] then [Expense/night = Low]
2	If [Region of Residence = (Australia or Western Europe or Singapore or China or Malaysia or United States)] and [Length of Stay in Hong Kong ≥ 7] then [Expense/night = Low]
3	If [$4 \leq \text{Length of Stay in Hong Kong} \leq 9$] and [Region of Residence = (Malaysia or United States or China)] then [Expense/night = Medium]
4	If [$13 \leq \text{Length of Stay in Hong Kong} \leq 15$] and [Region of Residence = China] then [Expense/night = Medium]
5	If [$10 \leq \text{Length of Stay in Hong Kong} \leq 12$] and [Region of Residence = (Australia or Taiwan or United States)] then [Expense/night = Medium]
6	If [Length of Stay in Hong Kong ≤ 3] and [Region of Residence = (Australia or Taiwan or United States)] then [Expense/night = High]
7	If [First Visit to Hong Kong = No] and [Length of Stay in Hong Kong ≤ 3] and [Region of Residence = (China or Malaysia or United States)] then [Expense/night = High]
8	If [$3 \leq \text{Length of Stay in Hong Kong} \leq 6$] and [Region of Residence = Western Europe or Singapore] then [Expense/night = High]
9	If [First Visit to Hong Kong = No] and [Length of Stay in Hong Kong ≥ 10] and [Region of Residence = (Australia or Taiwan or United States)] then [Expense/night = High]

the rough sets classification was determined in two quality terms of percentage of successfully classified cases (i.e., with a decision) and percentage of correctly classified cases (the estimated value matches the actual value). Experimental results are presented in the next section.

4 Empirical Findings and Discussion

Applying the rough sets model presented in the second section to the collected data, nine decision rules were induced (Table 2). In addition to being non-redundant in terms of the required number of rules and their conditions as well as the ability to capture only the essential condition attributes that influence the classification results, the induced rules matched about 70% of the training cases. In other words, the generalization is assured. In particular, five of the eight condition attributes were excluded in the induced set of decision rules.

As far as the estimation (or forecasting) quality is concerned, 49 of the 60 cases were classified and 20 of these classified cases had the same estimated and actual values. In other words, 82% of the testing cases were classified, and 41% of these classified cases were correctly estimated. A non-parametric Wilcoxon signed ranks test showed there was no significant difference between the sets of actual and estimated values (Table 3).

Table 2. Wilcoxon Signed Ranks Test

	Estimated - Actual
Z	-0.807 ^a
Asymp. Sig. (2-tailed)	0.420

a. Based on negative ranks.

Among the classified cases, more than three-quarters were either correctly classified or the difference between the estimated and actual values was 1. Table 4 shows the frequency and percentage distribution of the differences between the actual and estimated values. Such a finding further demonstrates the close resemblance of the actual and estimated values.

Table 3. Differences between the Actual and Estimated Values

Diff.*	Frequency	Valid Percent	Cumulative Percent
-2	4	8.2	8.2
-1	9	18.4	26.5
0	20	40.8	67.3
1	9	18.4	85.7
2	7	14.3	100.0

* Diff. represents the difference between the estimated and actual values.

Negative ranks: Estimated < Actual

Positive ranks: Estimated > Actual

Neutral: Estimated = Actual

5 Conclusions

The research has shown the promising results for data mining using the rough sets model developed for business travelers. Despite its limited scale in time frame and data coverage, empirical results show that the research is heading in a promising direction, and new insights are offered for tourism researchers to look into the issue of applying advanced data mining techniques to business travelers. On the basis of research findings, a number of areas are open for future investigations. One future research opportunity is to extend the developed model to include additional attributes/data. Another future research area is to combine the rough sets model with other data mining techniques to form a hybrid mining model. Such an approach may further improve the forecasting accuracy as well as the classification percentage. As an industry which is very much applied in nature, it would also be valuable to test the applicability of the induced rules in the industrial setting with input from all stakeholders.

As a final note, the business travel sector in Hong Kong presently faces numerous challenges. In the past decade, many countries in the Asia-Pacific region have recognized the significant contributions that this industry can make to a region's economy, and consequently have invested in substantial infrastructure developments complemented by the implementation of significant policy, planning, and marketing initiatives. For instance, Singapore, Thailand, Malaysia and Japan have become important players in recent years, and Mainland China is expected to show significant growth in the near future. Further compounding these competitive challenges is the impact of recent crises such as SARS, terrorist attacks, and avian flu that resulted in cancellations and/or postponements of business events with the various associated negative impacts on the tourism industry of the host destination. Hence, Hong Kong is no longer assured its position as one of the leading destinations for business travelers in Asia and has to take active measures to address these competitive issues. In view of the challenges that the Hong Kong inbound business traveling sector is facing, there is an urgent need for policy makers and practitioners to better understand the demand for business visitor arrivals, and to carry out more accurate planning at strategic, tactical, and operational levels.

References

1. Braun, B.M., Rungeling, B.: The relative economic impact of convention and tourist on a regional economy: a case study. *International Journal Hospitality Management* 11(1) (1992) 65-71
2. Grzymala-Busse, J.W., Goodwin, L.K., Zhang, X.: Increasing sensitivity of preterm birth by changing rule strengths. *Pattern Recognition Letters*. 24 (2003) 903-910.
3. Hong Kong Tourism Board: Statistics on Conventions & Exhibitions 2004. Available online at <http://partnet.hktourismboard.com/>. Accessed on February 3, 2006. (2005a).
4. Hong Kong Tourism Board: Visitor Profile Report 2004. Available online at <http://partnet.hktourismboard.com/>. Accessed on February 3, 2006. (2005b)
5. Hui, E.L.L., McKercher.: Operational Issues in Marketing Research: An Example of the Omnibus Tourism Survey. *Pacific Tourism Review* 5(1/2) (2001) 5-13

6. Katzberg, J., Ziarko, W.: Variable precision rough sets with asymmetric bounds. In: Ziarko W. (ed.): *Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD'93)*. Springer-Verlag, Berlin (1994) 167-177
7. Lawson, F. R.: Trends in business tourism management. *Tourism Management* 3(4) (1982) 298-302
8. Pawlak, Z.: Rough Set Elements. In: Polkowski, L., Skowron A. (eds.): *Rough Sets in Knowledge Discovery 1*. Physica-Verlag Heidelberg, New York (1998) 10-30
9. Slowinski, R., Zopounidis, C.: Application of the Rough Set Approach to Evaluation of Bankruptcy Risk. *Intelligent Systems in Accounting, Finance and Management* 4 (1995) 27-41
10. Tanaka, H., Maeda, Y.: Reduction Methods for Medical Data. In: Polkowski, L., Skowron, A. (Eds.): *Rough Sets in Knowledge Discovery, Vol. 2*. Physica-Verlag, Warsaw. (1998) 295-306
11. Ziarko, W.: Rough Sets. *Journal of Computer and Systems Sciences* 46 (1993a) 39-59
12. Ziarko, W.: Variable Prevision Rough Set Model. *Journal of Computer and Systems Sciences* 46(1) (1993b) 39-59
13. Ziarko, W.: Rough Sets as a Methodology for Data Mining. In: Polkowski L., Skowron A. (eds.): *Rough Sets in Knowledge Discovery 1*. Physica-Verlag Heidelberg, New York. (1998) 554-576

Feature Extraction Based on Optimal Discrimination Plane in ECG Signal Classification*

Dingfei Ge and Xiao Qu

School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310012, P.R.C.
gedingfei@vip.163.com

Abstract. In order to improve the classification results on electrocardiogram (ECG) signals, Optimal Discrimination Plane (ODP) approach is introduced. Features are extracted from time-series data using the ODP that is developed by Fisher's criterion method. ECG patterns are projected onto two orthogonal vectors, and the two-dimensional feature vectors are used as features to represent the ECG segments. Two types of ECG signals are obtained from MIT-BIH database, namely normal sinus rhythm and premature ventricular contraction. A quadratic discriminant function based classifier and a threshold vector based classifier are employed to classify these ECG beats, respectively. The results show the proposed technique can achieve better classification results compared to that of some recently published on arrhythmia classification.

1 Introduction

Due to the large number of patients in intensive care unit (ICU) and the need for continuous observation, numerous methods for cardiac arrhythmias classification have been proposed. Most of them base on various transform methods like wavelet transform [1], Fourier transform [2], Lyapunov transform [3] etc. Other methods include AR analysis [4], complexity measure [5], adaptive threshold method [6], direct electrocardiogram (ECG) feature extraction [7] etc. However, these methods seem to lose some of classification information [3]. Thus, the best feature extraction methods can be remained to present as a study for classification of various arrhythmias.

The purpose of the work is to extract ECG features from time-series data to improve classification accuracy on cardiac arrhythmias. Each sample point in a segment is weighted and fused according to the weighting factors. The procedures of the proposed method in the paper include redundancy removed by Principle Component Analysis (PCA), data normalized by whitening transform, feature extraction by Optimal Discrimination Plane (ODP) approach developed by Fisher's criterion method, and classification based on the threshold vector method and quadratic discriminant function (QDF). Whitening transform makes the within-class dispersion spheric. Two-dimensional features are extracted to represent ECG segments. The distribution of the feature vectors of the class NSR is near circular and close, while PVC distribution is wide relatively based on current method. The two classification methods,

* Supported by Zhejiang Province Natural Science Foundation, P. R. C. (Grant No. Y104284).

especially QDF are suitable for the classification under this case. In addition, the results also could be improved as soon as possible by subjectively adjusting the classifier parameters used in current research. In this study, two types of ECGs including normal sinus rhythm (NSR) and premature ventricular contraction (PVC) were used.

2 Methods

2.1 ECG Data, Filtering and Segmentation

The selected data including NSR and PVC with frequency 360HZ is taken from MIT-BIH arrhythmia database. Four patient's ECGs are selected from the database shown in Table 1. A band-pass filter with lower frequency passband 2Hz and upper frequency passband 20 Hz is utilized to filter the ECG signals. In current study, the sample size of the various segments is 0.9 seconds (325 sample points), which 0.3 seconds before R peak and 0.6 seconds after R peak are picked. A normal ECG refers to the usual case in the healthy adults where the heart rate is 60-100 beats per minute, which the RR intervals are observed in the range of 0.6- 0.9 seconds. Thus, a particular cardiac cycle is adequate to capture the most of ECG information.

Table 1. Evaluation data from the MIT-BIH arrhythmia database

Identification Number	Number of NSR (N_1)	Number of PVC (N_2)
Record 106	1507	520
Record 210	2423	194
Record 233	2230	831
Record 221	2029	394

2.2 Procedures of the Feature Extraction

The PCA is used to reduce the redundancy of time-series data, white transform is applied to the data in order to normalize them into spheric distribution. The features are extracted by ODP approach that is a linear technique and involved two projecting vectors based on Fisher's criterion. The ODP approach also finds another projecting vector that is orthogonal to the Fisher's vector.

Redundancy of the Data Reduced by PCA. (1) Calculate the covariance matrixes Σ_i of each class and the within-class scatter matrix S_w of the classes.

$$S_w = \sum_{i=1}^2 P_i \Sigma_i \quad (1)$$

where P_i is the prior probability of ω_i , $P_i=0.5$ in current study. (2) Calculate the eigenvalues and eigenvectors of S_w . (3) In order to select the d eigenvectors corresponding to the d largest eigenvalues of S_w , the separability criterion based on standard deviation and Euclidean center distance (*SDECD*) is used. The *SDECD* can be expressed as

$$J = \frac{\sqrt{\sum_{i=1}^d (\mu_{1i} - \mu_{2i})^2}}{3 \left(\frac{1}{d} \sum_{i=1}^d \sigma_{1ii} + \frac{1}{d} \sum_{i=1}^d \sigma_{2ii} \right)} \quad (2)$$

where σ_{1ii} and σ_{2ii} represent the standard deviations of individual variables of each class, respectively. $\mu_1=[\mu_{11}, \mu_{12}, \dots, \mu_{1d}]^T$ and $\mu_2=[\mu_{21}, \mu_{22}, \dots, \mu_{2d}]^T$ are the expected vectors for the classes, respectively. The criterion to select the d eigenvectors is to make the $J \geq 1.00$, which is calculated based on the reduced data. (4) Projecting each pattern onto these chosen eigenvectors to generate the sample vector x_d 's.

The Data Normalized by Whitening Transform. After redundancy of the data is reduced by PCA, the within-class dispersion of each class is normalized to spheric distribution by whitening transform.

The within-class scatter matrix of the reduced data (S_{wr}) is computed based on the class of NSR in this research. Suppose the eigenvalues and eigenvectors of S_{wr} are expressed by λ_k and p_k ($k=1,2,3,\dots,d$), respectively, p_k is a d -dimensional column vector, then the white transform is given by [9]

$$y_d = P^T x_d \quad (3)$$

where $P = (p_1 / \sqrt{\lambda_1}, p_2 / \sqrt{\lambda_2}, \dots, p_d / \sqrt{\lambda_d})$ is $d \times d$ matrix, y_d is a d -dimensional column vector.

2.3 ECG Feature Extraction

After the data is normalized, the ODP method is applied to the data to extract the two-dimensional features. The feature vector Z 's are obtained by projecting y_d 's onto two projecting vectors that are orthogonal between them, and used to represent the ECG segments. Suppose ; S_{wn} denotes the within-class scatter matrix of the normalized data y_d 's. First projecting vector v_1 called Fisher's vector is given by [9]

$$v_1 = S_{wn}^{-1}(m_1 - m_2) \quad (4)$$

Then another projecting vector v_2 can be found according to the ODP. The v_2 also maximizes the Fisher criterion and orthogonalizes with v_1 , which can be expressed as

$$v_2 = [S_{wn}^{-1} - \frac{(m_1 - m_2)^{-T} (S_{wn}^{-1})^2 (m_1 - m_2)}{(m_1 - m_2)^T (S_{wn}^{-1})^3 (m_1 - m_2)} * (S_{wn}^{-1})^2] (m_1 - m_2) \quad (5)$$

where m_1 and m_2 are the expected vectors for the classes, which are computed based on the normalized data, respectively.

2.4 Classification Based on QDF and Threshold Vector

After ODP process, the ECG segments represented by $Z=[Z_{v1}, Z_{v2}]$'s are classified using the QDF-Based algorithm and threshold vector $Z_0=[v_{1o}, v_{2o}]$. Suppose the

class ω_1 is NSR, ω_2 is PVC, m_1' and m_2' are the expected vectors of NSR and PVC, respectively, Σ_1' is the covariance matrix of NSR, then the QDF can be expressed as [10]

$$g(z) = k^2 - (z - m_1') \Sigma_1'^{-1} (z - m_1')^T \tag{6}$$

where k is the variable to be selected according to the classification goals. The decision-making rule is:

If $g(Z) > 0$, then Z belongs to PVC, otherwise Z belongs to NSR

A threshold vector $Z_0 = [v_{1o}, v_{2o}]$ is specified as

$$z_0 = \frac{N_1 m_1' + N_2 m_2'}{N_1 + N_2} \tag{7}$$

The decision-making rule is:

If $Z_{v1} < v_{1o}$ and $Z_{v2} < v_{2o}$, then Z belongs to NSR, If $Z_{v1} > v_{1o}$ or $Z_{v1} > v_{2o}$, then Z belongs to PVC.

During the training phase, m_1' and Σ_1' are computed using the random selected samples of the class NSR, the sample mean m_2' is computed using the random selected samples of the class PVC, then the QDF based classifier and the threshold vector Z_0 can be determined. During the testing phase, the value $g(z)$ is computed as the equation of (6) with a pre-selected k . The above decision-making rules were used to classify the rest testing data as belonging to a class.

In this research, individual ECG record is trained and tested, respectively. Then the overall training and testing for universal ECG records are performed.

3 Results

Forty largest eigenvalues and corresponding eigenvectors of the within-class scatter matrix S_w are selected based on the criterion *SDECD*. Thus, 40-dimensional feature vectors are used to represent the ECG segments after PCA. Two-dimensional feature vectors are extracted to represent the ECG segments by projecting the 40-dimensional feature vectors onto the Fisher's vector and its orthogonal vector. A mapping result of the testing data on ODP is shown in Fig.1.

In training phase, one hundred cases each from the two classes in individual ECG record are random selected for training, and the remaining is used for testing. Table 2 and 3 give the individual ECG record classification results on testing data based on the QDF algorithm and the threshold vector. The universal ECG record classification results are shown in table 4 based on QDF classifier and the training data set including 400 samples.

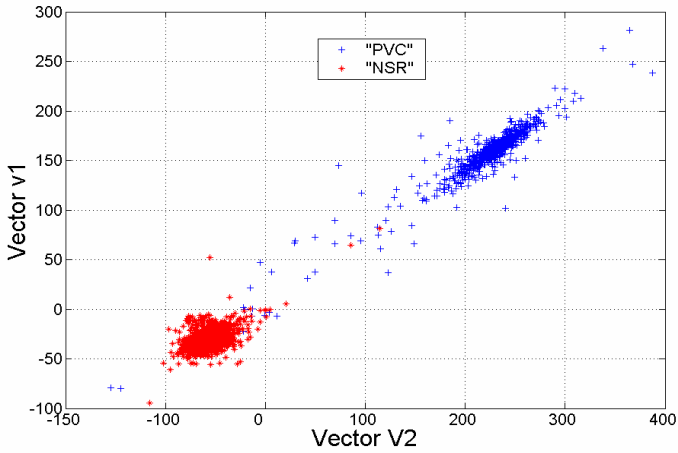


Fig. 1. A mapping result of the testing data on ODP

Table 2. Classification accuracy based on QDF algorithm

Classes	ECG Records	106	210	233	221
NSR	Accuracy	99.35%	97.82%	97.22%	99.37%
PVC	Accuracy	98.32%	91.98%	99.72%	99.82%
	Average	98.83%	94.90%	98.47%	99.59%

Table 3. Classification accuracy based on threshold vector

Classes	ECG Records	106	210	233	221
NSR	Accuracy	99.61%	98.45%	99.60%	99.84%
PVC	Accuracy	99.61%	91.44%	98.35%	99.48%
	Average	99.61%	94.94%	98.97%	99.66%

Table 4. Overall classification accuracy for universal ECG records based on QDF

Classes	Universal ECG Records	106, 210, 233, 221
NSR	accuracy	99.01%
PVC	accuracy	95.08%
	Average	97.04%

4 Discussions

The objective of this study is to extract ECG features from hyperdimensional time-series data in order to improve the classification results using the QDF and threshold vector based classifiers. ODP uses two orthogonal vectors including Fisher's vector. A good classification performance with average accuracy of 97.04 % has been achieved based on the extracted features and proposed classifiers. In general, class separability not only depends on the class distributions but also depends on the classifiers to be used. One can see from Table 2 and 3 that the classification accuracy is almost the same using the two different classifiers. In view of this, the class separability exhibits the less dependency on the classifiers, which does the good class distribution usually hold.

Our experimental results show the distribution of NSR is closer than that of PVC. So we only used the within-class scatter matrix of class NSR in order to make the distribution of NSR became more circular. One also can see from Fig.1 such a distribution is more suitable for the QDF based classification between NSR and PVC.

The proposed classification results were comparable to some recently published results on arrhythmias classification, for example, classify decimated ECG data including PVC and NSR using artificial neural network, an overall accuracy of 93% was obtained[11].

In addition, we used AR modeling technique to classify the same ECG data shown in Table1. The model order was 4, and the 4 AR coefficients were used as ECG features to represent ECG segments. The overall accuracy of detecting PVC and NSR is 84.83% and 92.05% based on AR modeling and QDF based classifier, respectively. The experimental results show that the ODP based method can better capture the information from ECG time-series data compared to AR modeling technique.

5 Conclusions

The ECG classification results could be improved using the features extracted from hyperdimensional time-series data by ODP based method and are suitable for real-time implementation for diagnosis purpose.

References

1. Khadra L., al-Fahoum A. S., al-Nashash H.: Detection of life-threatening cardiac arrhythmias using the wavelet transformation. *Med. Biol. Eng. Comput.* 35 (1997) 626-32.
2. Mroczka T., Lewandowski P., Maniewski R., et al.: Effectiveness of high resolution ECG spectral analysis in discrimination of patients prone to ventricular tachycardia and fibrillation. *Med. Sci. Monit.* 6 (2000) 1018-26.
3. Mohamed I. O., Abou-Zied Ahmed, H., M. Youssef Abou-Bakr et al.: Study of features based on nonlinear dynamical modeling in ECG arrhythmia detection and classification. *IEEE Trans. Biomed. Eng.* 49 (2002) 733-736.
4. Ge D. F., Srinivasan N., and Krishnan S. M.: Cardiac arrhythmia classification using autoregressive modeling. *Biomedical Engineering Online*, (2002) 1:5.

5. Sun Y., Chan K. L., Krishnan S. M.: Life-threatening ventricular arrhythmia recognition by nonlinear descriptor. *Biomed Eng Online*. (2005) 4:6.
6. Christon I. I.: Real time electrocardiogram QRS detection using combined adaptive threshold. *Biomed Eng Online*. (2004) 1:28.
7. Zhou S. H., Rautaharju P. M., Calhoun H. P.: Selection of a reduced set of parameters for classification of ventricular conduction defects by cluster analysis. *Proc. Comp. Cardiol*. (1993) 879-882.
8. Ge D. F., Shao Y. Q., Jiang H. Z.: An algorithm study on telecardiogram diagnosis based on multivariate autoregressive model and two-lead ECG signals. *Space Med. Med. Eng*. 17 (2004) 355-9.
9. Fukunaga K.: *Introduction to statistical pattern recognition*, Academic Press Limited, United States of America, New York, 1990.
10. Duda R. O., Hart P. E.: *Pattern classification*. Wiley Interscience Publication, New York, 2001.
11. Melo S. L., Caloba L. P., Nadal J.: Arrhythmia analysis using artificial neural network and decimated electrocardiographic data. *Comp. Cardiol*. 27 (2000) 73-76.

Music Style Classification with a Novel Bayesian Model

Yatong Zhou¹, Taiyi Zhang¹, and Jiancheng Sun²

¹ Dept. Information and Communication Engineering,
Xi'an Jiaotong University,
710049 Xi'an, P.R. China
{zytong, tyzhang}@mailst.xjtu.edu.cn

² Dept. Communication Engineering,
Jiangxi University of Finance and Economics,
330013 Nachang, P.R. China
sunjc@jxufe.edu.cn

Abstract. Music style classification by mean of computers is very useful to music indexing, content-based music retrieval and other multimedia applications. This paper presents a new method for music style classification with a novel Bayesian-inference-based decision tree (BDT) model. A database of total 320 music staffs collected from CDs and the Internet is used for the experiment. For classification three features including the number of sharp octave (NSO), the number of simple meters (NSM), and the music playing speed (MPS) are extracted. Following that, a comparative evaluation between BDT and traditional decision tree (DT) model is carried out on the database. The results show that the classification accuracy rate of BDT far superior to existing DT model.

1 Introduction

Music as a carrier of human emotion is one of the most important sources distributed by the Internet. However, it is still difficult for a computer to automatically analyze music content, especially to classify music style. Since style provides important information in the music, it would be very useful to music indexing, content-based music retrieval and other multimedia applications if we could automatically classify or discriminate music style.

Music style classification (MSC) has been receiving an increasing attention in recent years. Qin introduced a MSC system that taken MIDI as data source and mined frequent patterns of different music [1]. Kuo developed the multi-type melody style classification system to recommend the music objects [2]. In Ma's work, a music style classification algorithm was proposed to measure music style through melody mutual Information [3]. Hsu classified MIDI objects by fast discovering nontrivial repeating patterns [4]. Zhang presented a study on music classification using short-time analysis together with data mining techniques to distinguish between five music styles [5]. Word presented a method for content-based audio music files classification, in which they used duration, pitch, amplitude, brightness and bandwidth as features [6]. Xu proposed effective algorithms to automatically classify music into pure music and vocal music [7]. Above-mentioned works can be primarily separated into two classes, one takes MIDI as data source, and the other takes audio as source. However, current

work has addressed little on taking music staff as source. In this paper we will take music staff as source since it is quite general.

In music style classification, music is classified into different categories based on different style. These categories, such as pure music and vocal music, or symphony and Beijing opera, can be previously defined. In this paper we focus on automatically classifying music into the pre-defined categories—pleasurable music and sorrowful music. It is well known that the classifiers play a crucial role in such a binary classification problem. Based on traditional decision tree (DT), we propose a Bayesian-inference-based decision tree (BDT) model and employ it as the classifier.

The DT model automatically constructed from data that have been used successfully in many real-world situations such as knowledge and information extraction from databases or web [8]. Despite it’s many successes, from a Bayesian perspective, the DT model lacks of a probabilistic background. One way to tackle this problem is applying Bayesian approach to it. When applied to the DT, the Bayesian approach allows ready in corporation of prior knowledge and the seamless combination of such knowledge with observed data. Moreover, because of the Bayesian nature, the approach can treat uncertainty uniformly at all levels of the modeling process. Thus not only builds the ability to infer the model parameters in Bayesian approach but also provides posterior class probability. Based on the reasons mentioned above, we apply Bayesian approach to DT and seek to incorporate the Bayesian inference into the learning procedure of the model. As a result, the BDT model is proposed in this paper. It is anticipated that the performance of BDT superior to existing DT model.

2 Classification Framework and Feature Extraction

Our classification framework is illustrated in Fig.1. The first step is to construct a music staff database. For this purpose we collected 320 music staves including pianos, symphonies, popular songs, and Chinese folk songs from music CDs and the Internet. After that, to convert the music staves to text files that the computer could be read, we define the music staff conversion rules shown in Table.1. For example, a segment of music staff given in Fig.2 could be converted into the text file shown in Fig.3.

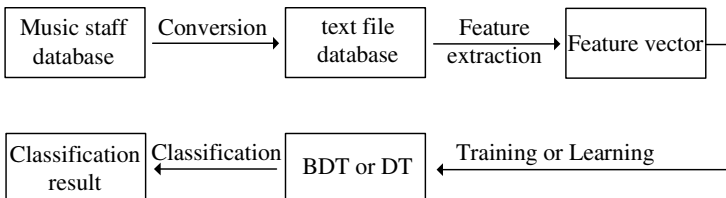


Fig. 1. Our classification framework

It is a challenge to extract the most common and salient features to characterize the music style from unstructured text files. The music belonging to a same style has some commonness. People usually define the commonness by some high-level perceptive features, such as melody and rhythm. However it is difficult to give a deter-

minate description for these high-level features. Fortunately, the high-level features can be reflected through some low-level features. The first low-level feature we extracted is the number of sharp octave (NSO) that reflects the music’s melody. There are more chances with ascending tune in pleasurable music, where sharp octave comes out more often and NSO is large. On the contrary, NSO is small for sorrowful music. The second feature, the number of simple meters (NSM), is extracted to reflect the music’s rhythm. Since pleasurable music usually owns rapid or strong rhythm, correspondingly, NSM is large. The third feature is the music playing speed (MPS).

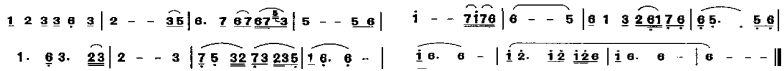


Fig. 2. Example of staff segmentation for a piece of music

```

1_2_3_3_6#_3_|2--3_5_|6_7_6_7_6_7_3_|5--5_6|$ 1^--7_1^_7_6_|6--
5|6_1_3_2_6#_|1_7#_|6#_|6#_5#_|5#_6#_|$ 1. 6#_3. 2_3_|2--3|7#_5#_3_2_
7#_3_2_3_5#_|1_6#_|6#-|$ 1^_6. 6-|1^_2^_|1^_2^_|6|1^_6. 6-|6--||$
    
```

Fig. 3. Example of text file for a segment of music staff

After feature extraction three features are sent into the BDT and DT model. They will be further utilized for training model. Finally, for music with unknown style, the classification result is produced when it sent into the trained model.

Table 1. The music staff conversion rule

Symbol	Meaning	Symbol	Meaning
1-7	Pitch	—	Simple meter
^	Sharp octave	—	Duple meter
#	Flat octave	.	Dotted note
0	Null meter		Bar
	Music cadence	\$	Text file end

3 Proposed BDT Model for Classification

3.1 A Review on DT Model

The DT model addresses a classification problem by building a binary tree. The tree consisting of nodes and branches is a recursive structure for expressing classification rules. The classification boundary obtained by DT can be represented as

$$f(\mathbf{x}) = \sum_{i=1}^k \beta_i B_i(\mathbf{x}), \tag{1}$$

where β_i are the coefficients of the basis $B_i(\mathbf{x})$ and k is the number of leaf nodes in the model. The basis functions is the product of J_i Heavisine functions defined as

$$B_i(\mathbf{x}) = \prod_{j=1}^{J_i} H \left[S_{ji} \left(x^{v(ji)} - r_{ji} \right) \right], \tag{2}$$

where the sign indicators S_{ji} is equal ± 1 . The knot points r_{ji} give the positions of the splits and $v(ji)$ give the index of the variable which is being split on the r_{ji} .

In the DT model, the parameters $k, J_i, S_{ji}, r_{ji}, v(ji)$ and β_i ($j=1,2,\dots,J_i, i=1,2,\dots,k$) are set to single optimal values. This optimization is achieved gradually in the model’s learning procedure that includes a series of splitting and pruning operations. For convenience, we take these parameters the vector $\boldsymbol{\theta}^{(k)}$ as a whole.

3.2 Implemental Process of BDT Model

The BDT model is proposed when applying Bayesian approach to the DT. Similarly with DT, the classification boundary obtained by BDT can also be represented as

$$f(\mathbf{x}, k, \boldsymbol{\theta}^{(k)}) = \sum_{i=1}^k \beta_i B_i(\mathbf{x}), \tag{3}$$

where the vector $\boldsymbol{\theta}^{(k)}$ is written out explicitly. Then the parameterized output of BDT can be represented as

$$P(t | \mathbf{x}, k, \boldsymbol{\theta}^{(k)}) = \frac{1}{1 + \exp(-f(\mathbf{x}, k, \boldsymbol{\theta}^{(k)}))}. \tag{4}$$

By marginalization which integrates the vector $\boldsymbol{\theta}^{(k)}$ out from Eq. (4), we obtain the output of BDT, the posterior probability that sample \mathbf{x} belongs to a particular class t

$$P(t | \mathbf{x}) = \sum_k \int P(t | \mathbf{x}, k, \boldsymbol{\theta}^{(k)}) P(k, \boldsymbol{\theta}^{(k)} | D) d\boldsymbol{\theta}^{(k)}, \tag{5}$$

where $P(k, \boldsymbol{\theta}^{(k)} | D)$ is the posterior distribution and D is the learning samples set. Clearly the integral in Eq. (5) is computationally intractable and some approximation method is required. An elegant solution is provided by MCMC [10] which allows one to draw N_c samples $(k_n, \boldsymbol{\theta}_n^{(k_n)})$, $n=1,2,\dots,N_c$, from the posterior $P(k, \boldsymbol{\theta}^{(k)} | D)$ and then approximate Eq. (5) by

$$P(t | \mathbf{x}) \approx \frac{1}{N_c} \sum_{n=1}^{N_c} P(t | \mathbf{x}, k_n, \boldsymbol{\theta}_n^{(k_n)}). \tag{6}$$

The next step is to calculate the posterior $P(k, \boldsymbol{\theta}^{(k)} | D)$. It can be obtained with the prior $P(k, \boldsymbol{\theta}^{(k)})$ and likelihood $P(D | k, \boldsymbol{\theta}^{(k)})$ based on Bayesian inference

$$P(k, \boldsymbol{\theta}^{(k)} | D) = P(\boldsymbol{\theta}^{(k)}, k) P(D | k, \boldsymbol{\theta}^{(k)}) / P(D) \quad . \quad (7)$$

The prior is factorized as

$$P(k, \boldsymbol{\theta}^{(k)}) = P(k) P(\boldsymbol{\theta}^{(k)} | k) \quad . \quad (8)$$

In literature [9], Denson presented a Bayesian version of the classification and regression tree (CART), resulting in BCART model. Similar with BCART, a Poisson distribution with parameter λ is used to specify the prior for the k , giving

$$P(k) = \lambda^k / (e^\lambda - 1) k! \quad . \quad (9)$$

The conditional distribution $P(\boldsymbol{\theta}^{(k)} | k)$ can be factorized as the product of the probability over each element of the vector $\boldsymbol{\theta}^{(k)}$. On the other hand, we assume the likelihood with the form of

$$P(D | k, \boldsymbol{\theta}^{(k)}) = \prod_{n=1}^N \left(P(t_n | \mathbf{x}_n, k, \boldsymbol{\theta}^{(k)}) \right)^{t_n} \left(1 - P(t_n | \mathbf{x}_n, k, \boldsymbol{\theta}^{(k)}) \right)^{1-t_n} \quad . \quad (10)$$

3.3 Sampling Via RJMCMC

Now the key to BDT is how to draw N_c samples from the posterior $P(k, \boldsymbol{\theta}^{(k)} | D)$. However, the parameter k is unknown and the dimension of the posterior is varying. Therefore we will generate samples from the posterior by RJMCMC [10]. For the BDT model, supposing that the current number of leaf nodes equals to k_m , i.e. $k = k_m$, the corresponding model posterior distribution is $P(k_m, \boldsymbol{\theta}^{(k_m)} | D)$. We construct ergodic Markov chains admitting $P(k_m, \boldsymbol{\theta}^{(k_m)} | D)$ as the invariant distribution. However, the Markov chains would admit $P(k_n, \boldsymbol{\theta}^{(k_n)} | D)$ as the invariant distribution when the number of leaf nodes changes to k_n . The parameters $\boldsymbol{\theta}^{(k_m)} \in R^{N_m}$ and $\boldsymbol{\theta}^{(k_n)} \in R^{N_n}$ are model dependent and the dimension of subspaces R^{N_m} and R^{N_n} are different.

RJMCMC allows the sampler to jump between the different subspaces. To ensure a common measure, it requires the extension of each pair of communicating subspaces, R^{N_m} and R^{N_n} . It also requires the definition of deterministic, differential, invertible dimension matching functions $\varphi_{n \rightarrow m}$ and $\varphi_{m \rightarrow n}$ between the extended subspaces

$$\left(\boldsymbol{\theta}^{(k_m)}, \mathbf{u}_{m,n}\right) = \varphi_{n \rightarrow m}\left(\boldsymbol{\theta}^{(k_n)}, \mathbf{u}_{n,m}\right), \left(\boldsymbol{\theta}^{(k_n)}, \mathbf{u}_{n,m}\right) = \varphi_{m \rightarrow n}\left(\boldsymbol{\theta}^{(k_m)}, \mathbf{u}_{m,n}\right), \quad (11)$$

where $\mathbf{u}_{m,n} \in U_{m,n}$ and $\mathbf{u}_{n,m} \in U_{n,m}$. If the current state of the chain is $(k_n, \boldsymbol{\theta}^{(k_n)})$, we jump to $(k_m, \boldsymbol{\theta}^{(k_m)})$ by generating $\mathbf{u}_{n,m} \sim q_{n \rightarrow m}(\cdot | n, \boldsymbol{\theta}^{(k_n)})$, ensuring that Eq. (11) holds, and accepting the jump according to some probability ratio. The RJMCMC sampler is iterated until enough samples have been collected. An initial portion is discarded to allow time for the chain to converge sufficiently closely to its invariant distribution. In the end N_c samples can be obtained.

4 Experiments

The experiment is simulated in Matlab circumstance. The computer used is Celeron (2.8GHZ) PC with 256 RAM. In the experiment, 100 pleasurable and sorrowful music samples are selected respectively from the database to form the training set. The remaining samples are put into the test set. That means we will use totally 200 samples for the training and 120 samples for the testing. A comparative evaluation between DT and BDT is carried out and the results shown in Fig. 4 and Table 2 are promising.

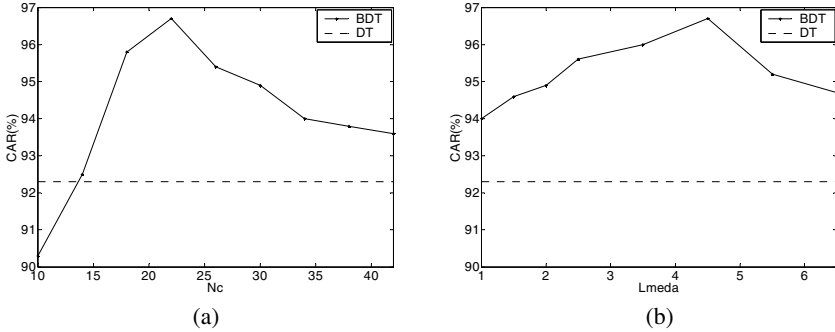


Fig. 4. Comparison of classification accuracy rate (CAR) between BDT and DT. The left graph shows the dependence of CAR on the parameter N_c with the fixed value of $\lambda = 4.5$. The right graph shows the dependence of CAR on the parameter λ with $N_c = 22$.

The BDT model has two main user set parameters N_c and λ . Fig.4 illustrates the variation of classification accuracy rate (CAR) with two parameters. From the figure (a) we can see that BDT outperforms DT in all the cases except for $N_c = 10$. Especially, for the BDT model there is an obvious improvement on CAR when $N_c = 22$. The curve in Fig.4 (b) is quite different from that of Fig.4 (a). It's found that BDT outperforms DT in all the cases no matter what value of λ taken.

To give a further investigation on model's computational efficiency, we compare the CPU time in seconds consumed for learning between BDT and DT. With view of

Table.2, the CPU time for BDT is little longer than DT, whereas CAR improves a lot if the parameter N_c selected appropriately (such as $N_c = 22$ or $N_c = 26$). In other words, BDT takes little longer time and yields quite better performance.

Table 2. Classification results on music database with the fixed value of $\lambda = 4.5$. Time denotes the CPU time in seconds consumed for learning.

Items	DT	BDT					
N_c	---	10	18	26	34	50	100
Time	81.9	79.4	102.5	124.7	158.3	220.2	551.2
CAR	92.3%	90.3%	95.8%	95.4%	94.0%	93.7%	93.2%

5 Conclusions

In this paper, a new method for music style classification using proposed BDT model is presented. When applying Bayesian approach to traditional DT model and seeking to incorporate the Bayesian inference into the learning procedure, the BDT model is built. In the experiment, a database of total 320 music staves is used for a comparative evaluation between BDT and DT. From experimental results, we observe that the performance of BDT far superior to DT model.

References

1. Qin D., Ma G. Z.: Music style identification system based on mining technology. *Computer Engineering and Design*. 26(2005) 3094–3096
2. Ma G. Z., Qin D.: Music style classification using mutual information. *Computer Applications*. 25(2005) 1116–1118 (In Chinese)
3. Kuo F. F., Shan M. K.: A personalized music filtering system based on melody style classification. In: Blum (eds.): *Proc. IEEE Int. Conf. Data Mining*. (2002) 649–652
4. Hsu J., Lin C., Chen A. L.: Discovering Nontrivial Repeating Patterns in Music Data. *IEEE Trans. Multimedia*. 3(2001) 311–325
5. Zhang Y. B., Zhou J.: A study on content-based music classification. In: Jordan (eds.): *Proc. 7th Int. Sym. Signal Processing and Its Applications*, Paris, France. 2(2003) 113–116
6. Word E., Blum T., Keislar D.: Content-Based Classification, Search, and Retrieval of Audio. *IEEE Trans. MultiMedia*. 3(1996) 27–36
7. Xu C. S., Maddage N. C., Shao X.: Automatic music classification and summarization. *IEEE Trans. Speech and Audio Processing*. 3(2005) 441–450
8. Lee S. K.: On generalized multivariate decision tree by using GEE. *Computational Statistics & Data Analysis* 49(2005) 1105–1119
9. Denson D. G. T.: Simulation based Bayesian nonparametric regression methods. Ph.D Dissertation. Imperial College, London University, 2001
10. Green P. J.: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*. 82(1995) 711–732

Classification of Polarimetric SAR Data Based on Multidimensional Watershed Clustering

Wen Yang^{1,2}, Hao Wang¹, Yongfeng Cao¹, and Haijian Zhang¹

¹ School of Electronic Information, Wuhan University, Luoyu Road 129#, Wuhan 430079, Hubei Province, China
yw@eis.whu.edu.cn

² SOA Key Laboratory for Polar Science, Polar Research Institute of China, Shanghai 200136, China

Abstract. This paper proposes a polarimetric synthetic aperture radar (PolSAR) data classification method which applies multi-dimensional transform to identify density peaks and valleys for polarimetric signatures clustering. The new approach firstly introduces an improved maximum homogeneous region filter which can effectively preserve structure feature and polarimetric signatures. Then polarimetric signatures are extracted based on Freeman-Durden three-component composition. Finally, we obtain the classification results by multi-dimensional watershed clustering on the extracted polarimetric signatures. The effectiveness of this classification scheme is demonstrated using the full polarimetric L-band SAR imagery.

1 Introduction

PolSAR is a well-established multidimensional SAR technique based on acquiring earth's surface information by means of using a pair of orthogonal polarizations for the transmitted and received electromagnetic fields. With radar polarization, the textural fine structure, target orientation and shape, symmetries and material constituents of the Earth surface can be covered with considerable improvements above of that standard "amplitude-only radar" [1]. During the last decade, multi-frequency and polarimetric SAR imaging has been investigated with respect to classification of terrain types, many supervised and unsupervised classification methods for PolSAR data have been proposed. When the ground truth is not available, supervised classification does not work well since there are many difficulties to select significant training sample sets [2]. In this paper, we propose a multidimensional watershed based unsupervised classification algorithm which classifies automatically the PolSAR data by finding the clusters using multidimensional watershed transform.

The organization of the rest of the paper is as follows: In the next section, the extraction of polarimetric signatures by applying the Freeman-Durden decomposition [3] and an improved speckle reduction process ahead will be presented. The effective multidimensional watershed clustering algorithm is proposed for classifying the pixels according to their polarimetric signatures in Section 3. And the experimental results and analysis of utilizing this classification scheme are presented in Section 4. Finally, some discussions and conclusions are presented in Section 5. L-band Pi-SAR (The

Pi-SAR sensor is an airborne POLSAR system developed by NICT and JAXA of Japan. The resolution in the L-band image is 3 m×3m) data are used for illustration.

2 Polarimetric Signatures Extraction

Polarimetric target decomposition theories decompose the polarimetric signatures into several elementary scattering mechanisms. These decomposition methods can benefit from polarimetric preservation of speckle filtering, and improve class definition the statistical variation in stochastic data due to speckle needs to be reduced. Therefore, a speckle reduction process will be applied before the extraction of polarimetric signatures.

The speckle reduction problem is more complicated for polarimetric SAR than a single polarization SAR due to the difficulties of preserving polarimetric properties and of dealing with the cross-product terms. To preserve the polarimetric signature, each element of the covariance matrix of polarimetric SAR image should be filtered independently in the same local homogenous region. A fundamental problem with speckle reduction techniques that use averaging, is the reduction of resolution and the attendant smearing of line features in the data. Some other methods adopt the fixed-size sliding window, which leads to that the size of homogenous region can't be adapted to the variance of local texture, and thus it's very hard to make a good trade-off between structure feature preserving and speckle smoothing. As for these problem, we proposes an improved approach- Maximum Homogeneous Region (MHR) Polfilter based on J.S.Lee filter [4], which uses an adaptive window to search a maximum homogeneous region, and the size of the maximum homogeneous region can be adjusted depending on the statistics of local texture. The latter experimental results demonstrate that this approach can effectively preserve structure feature and polarimetric properties, and meanwhile obtains a rather good de-speckling performance.

The purpose of target decomposition is to provide means for interpretation and optimum utilization of polarimetric scattering data based on sensible physical constraints. Since the initial work of J.R.Huynen, there are many proposed decomposition theorems, and we focus here on the "model based" decomposition of the covariance matrix usually called Freeman-Durden decomposition. This decomposition uses simple scattering processes to model the scattering mechanism of the Earth surface and describes the polarimetric backscatter from naturally occurring scatters. An advantage of this model is that the scattering mechanisms can be estimated for clusters of pixels in polarimetric SAR images. According to this model, backscattering from terrain can be regarded as the superposition of three single scattering processes: surface, dihedral and volume scattering. So pixels are divided into three scattering categories: surface, double bounce, and volume, respectively.

Here we follow the scheme by Freeman and Durden [3] and utilize the covariance matrix to re-derive the decomposition, the measured covariance matrix is written as

$$\langle [C] \rangle = \begin{bmatrix} \langle |S_{HH}|^2 \rangle & \sqrt{2} \langle S_{HH} S_{HV}^* \rangle & \langle S_{HH} S_{VV}^* \rangle \\ \sqrt{2} \langle S_{HV} S_{HH}^* \rangle & 2 \langle |S_{HV}|^2 \rangle & \sqrt{2} \langle S_{HV} S_{VV}^* \rangle \\ \langle S_{VV} S_{HH}^* \rangle & \sqrt{2} \langle S_{VV} S_{HV}^* \rangle & \langle |S_{VV}|^2 \rangle \end{bmatrix} \quad (1)$$

Using the Freeman-Durden approach, we can rewrite the covariance matrix as formula (2),

$$\begin{aligned}
 \langle [C] \rangle &= f_s \langle [C] \rangle_{surface} + f_d \langle [C] \rangle_{double} + f_v \langle [C] \rangle_{vol} \\
 &= f_s \begin{bmatrix} \beta^2 & 0 & \beta \\ 0 & 0 & 0 \\ \beta & 0 & 1 \end{bmatrix} + f_d \begin{bmatrix} \alpha^2 & 0 & -\alpha \\ 0 & 0 & 0 \\ -\alpha & 0 & 1 \end{bmatrix} + f_v \begin{bmatrix} 1 & 0 & 1/3 \\ 0 & 2/3 & 0 \\ 1/3 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} f_s \beta^2 + f_d \alpha^2 + f_v & 0 & f_s \beta - f_d \alpha + \frac{f_v}{3} \\ 0 & \frac{2f_v}{3} & 0 \\ f_s \beta - f_d \alpha + \frac{f_v}{3} & 0 & f_s + f_d + f_v \end{bmatrix}
 \end{aligned} \tag{2}$$

We can estimate the contribution on the dominance in scattering powers of P_s, P_d, P_v , corresponding to surface, double bounce and volume contributions, respectively, are obtained as

$$P_s = f_s(1 + \beta^2) \quad P_d = f_d(1 + \alpha^2) \quad P_v = 8f_v/3 \tag{3}$$

3 Multidimensional Clustering Based on Watershed Transform

Our approach is density-based which considers that clusters are high-density regions separated by low density regions. Traditional density-based methods do not work well when clusters are closed to each other, both the cluster centers and cluster boundaries (as the peaks and valleys of the density distribution) become fuzzy and difficult to determine [5]. In this paper, multidimensional watershed transform is applied to identify density peaks and valleys in density landscape for overcoming this problem.

Watershed transform is usually used as a region-based segmentation approach, and it produces a complete division of the image in separated regions even if the contrast is poor, and does not usually destroy the edges of ideal segmentation. The intuitive idea underlying this method comes from geography: it is that of a landscape or topographic relief which is flooded by water, watersheds being the divide lines of the domains of attraction of rain falling over the region [6]. Multi-dimension watershed clustering algorithm is based on simulating this process for data clustering [7]. Firstly, regarding the data density space (here is the feature vector space) as a multi-dimensional topographic relief. Then applying the ‘‘rainfall’’ or ‘‘flooding’’ operation on it. Finally, when the process is stopped, the density space is partitioned into regions or basins separated by dams, called watershed surfaces or simply watersheds. Data in the same watershed can be assigned a type. It has a better partition for clusters which have arbitrary shapes comparing with some other algorithms, and the

computing cost only depends on the size of density space and the dimension of attributes. The major steps of our method are as follows.

3.1 Data Pre-processing

We consider a dataset consisting of data points $x_i = (x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{ip}) \in P$ in attribute space P , where $i = 1, 2, \dots, N$, and each component $x_{ij} \in P_j$ is a numerical or nominal categorical attribute (here are the decomposed polarimetric signatures, each pixel is described by a set of polarimetric signatures.). The ultimate goal of clustering is to assign points to a finite system of k subsets, or clusters. Since the attributes at each point may have different units (length), this would be make ad hoc transformations or normalizations of the attributes necessary, so here we take a normalization transformation [7].

3.2 Construct Density Space

In density-based approaches, we must need to estimate the density of data. The usual influence functions include a gaussian influence function, and a square wave function. After pre-processing, we set up P -dimensional density space D^P by utilizing density functions. A density function can be considered as the superposition of several influence functions. For a dataset $D = \{x_i\}_{i=1}^N \subset R^P$, the density on x_i is defined as

$$f_B^D(x_i) = \sum_{i=1}^N f_B^{x_i}(x_i) \quad (4)$$

Here, we select the gaussian influence function. For convenience, we denote density space as D_1^P , and every point in D_1^P equals to subtracting the corresponding point from the maximum in D^P .

3.3 Determine the Initial Flooding Points

We can determine the initial flooding points by basin dynamic which was first proposed by Grimaud [8] as a measure for basin's saliency. Suppose M is a basin in density space, we can define basin dynamic by associating the dynamic of a minimum to its attraction basin. The dynamic of a minimum M is defined as below,

$$\min(\max_{s \in [0,1]} (f(\gamma(s)) - f(\gamma(1)) | \gamma: [0,1] \rightarrow R^2, f(\gamma(1)) < f(\gamma(0)), \gamma(0) \in M)) \quad (5)$$

where γ is a path linking two points. In practice, we can define the basin dynamic of M as the minimum of difference of all paths linking bottom point of basin M to the closest basin's bottom point having a lower value. The lowest basin has no dynamic, so we give it the maximum value. The algorithm can also be described in the following way: When two basins meet at an altitude of immersion h , we compare their respective minima. The basin with the higher minimum pours into the second one, and we evaluate the dynamic of the former by the formula:

$$\text{Dyn}(M) = h - h_M \quad (6)$$

The basins which have the maximum dynamic value are the initial flooding point. Sometimes manual selection is also a good choice for determining the initial flooding points [9].

3.4 Watershed Transform on Density Space

An implementation of the watershed transform on density space D_1^p can be conducted after determining the initial flooding points. Denote h_{\min} and h_{\max} are the lowest valley and highest peak of the density distribution, respectively. Sorting density space points to assume that the sorted array enables a direct access to the points at a given density level h ($h_{\min} \leq h \leq h_{\max}$). When points have been sorted, the progressive flooding of the catchment basins of the density space is executed. Suppose the flooding has been done up to a given level h . Each catchment basin whose corresponding minimum has a level lower or equal to h is supposed to have a unique label. Due to the initial sorting, we now access the points of level $h+1$ directly, and those points among which have labeled neighbors are put into FIFO stack. Pop-up points in FIFO stack orderly, and label them by the label of neighbor points or dam value whose neighbors has a different label number. After this step, only the points that at level $h+1$ and unconnected with any of the labeled catchment basins have not been reached. Therefore, scanning of the points at level $h+1$ again is necessary to detect the points which are still unlabelled. Repeat this step until FIFO is empty, and all the points at level $h+1$ are labeled. The stepwise flooding process will be done when the h reach the highest peak h_{\max} .

4 Classification of PolSAR Images

There are two approaches to scene classification: unsupervised or supervised. Here we take the former. Unsupervised classification leads to an understanding of the class separability in the scene that is supported by the polarimetric signatures of the data [10]. A difficulty with unsupervised classification is that convergence depends on the initial seeding of candidate classes, but our algorithms can avoid this problem. Fig.1 gives our whole classification scheme, and two groups of classification are given in this section to illustrate the effectiveness of the classification algorithm.

Fig.2 shows the original and filtered pseudo-color image of L-band Pi-SAR polarimetric SAR data. We can find that all the important point like targets are preserved, lines are brighter and boundaries are sharper, and meanwhile it also preserve the polarimetric signatures and statistical correlation. Fig.3 shows the classification results of the original PolSAR data in Fig.2 with K-means algorithm and our approach, and Fig.4 give the classification results for the filtered PolSAR data based on MHR Polfilter. The scene includes cropland, water, woods and some buildings. On the basis of the classified image and a set of regions representing test areas, the confusion matrices are calculated for checking the accuracy of classification. Table 1 and 2

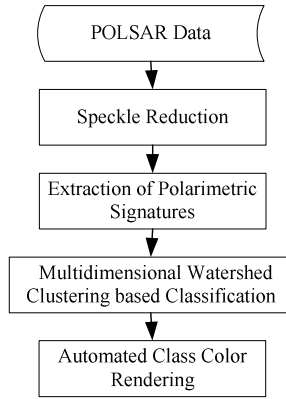


Fig. 1. Flow chart of multi-dimensional clustering based PolSAR data classification

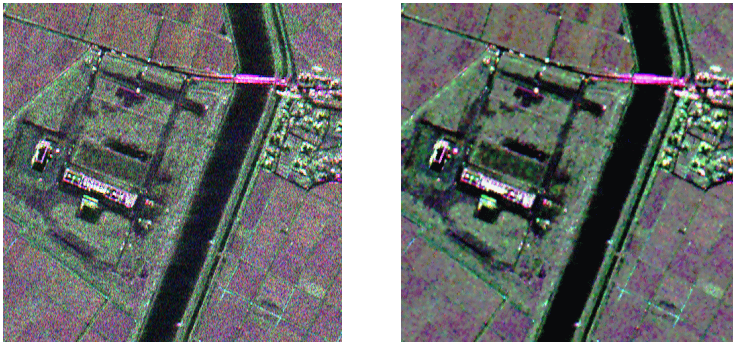


Fig. 2. Color composite image of original and filtered PolSAR data (HH: red, HV: green, VV: blue)

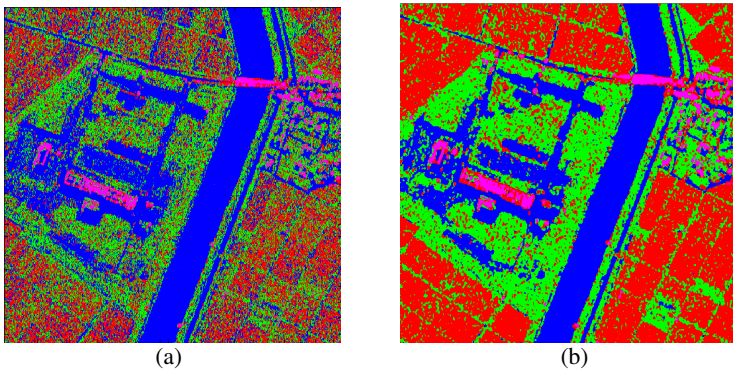


Fig. 3. Classification results without speckle reduction: (a) by K-means (b) by our approach

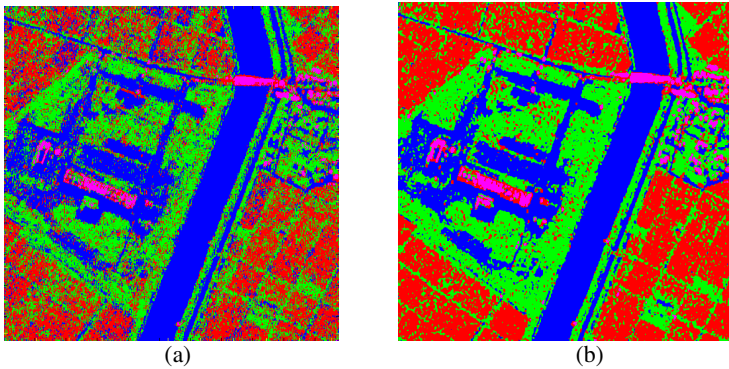


Fig. 4. Classification results with speckle reduction: (a) by K-means (b) by our approach

Table 1. Confusion matrix of classification result in Fig.4.(a)

	Cropland	Water	Woods	Building
Cropland	0.5866	0.1905	0.2114	0.0115
Water	0.0000	0.9995	0.0000	0.0005
Woods	0.2255	0.2107	0.5629	0.0009
Building	0.1078	0.0948	0.1171	0.6803

Table 2. Confusion matrix for classification result in Fig.4.(b)

	Cropland	Water	Woods	Building
Cropland	0.9069	0.0031	0.0893	0.0007
Water	0.0004	0.9996	0.0000	0.0000
Woods	0.1468	0.0450	0.8082	0.0000
Building	0.2066	0.0103	0.0103	0.7728

gives the confusion matrix for classified image in Fig.4(a) and (b) , respectively. Comparing with K-means clustering algorithm, our approach is more robust in noise environment, and obtains rather better performance, especially in cropland and woods.

5 Conclusions

In this paper we present a novel classification algorithm of PolSAR data based on multi-dimensional watershed clustering which is shown to be more robust and reliable than traditional methods that perform peak or valley seeking on density functions. Further work will focus on utilizing the approach with more refined features, and assessing quantitative classification accuracy by supervised segmentation.

Acknowledgement

The authors would like to thank the anonymous referees for the detailed, valuable suggestions. This work has been supported by Funds of LIESMARS, SOA Key Laboratory for Polar Science (KP200509), and NSFC project (60372057, 40376051). We also thank to Signal Processing Laboratory of Wuhan University for providing the facility to test and evaluate our algorithms over their software platform about interpretation of polarimetric SAR imagery.

References

1. Touzi, R., Boerner, W.M., Lee, J.S., Lueneburg, E. A Review of Polarimetry in the Context of Synthetic Aperture Radar: Concepts and Information Extraction, *Can. J. Remote Sensing*, Vol. 30, No. 3, pp. 380–407, 2004
2. Lee, J.S., Grues, M.R., Pottier, E., Ferro-Famil, L. Unsupervised Terrain Classification Preserving Polarimetric Scattering Characteristics, *IEEE Trans.Geosci.Remote Sensing*, Vol. 42, No. 4, pp. 722–731, 2004
3. Freeman, A., Durden, S.L. A Three-component Scattering Model for Polarimetric SAR Data. *IEEE Trans.Geosci.Remote Sensing*, Vol. 36, No. 3, pp. 963–973, 1998
4. Lee, J.S., Grues, M.R., Grandi, G. Polarimetric SAR speckle filtering and its implication for classification. *IEEE Trans.Geosci.Remote Sensing*, Vol. 37, No. 5, pp. 2363–2373, 1999
5. Andy, M.Y., Chris, D., Tony, F.C., Dynamic Cluster Formation Using Level Set Methods, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 28, No. 6, pp. 877–889, 2006
6. Najman, L., Schmitt, M., Geodesic Saliency of Watershed Edges and Hierarchical Segmentation, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 18, No. 12, pp. 1163–1173, 1996
7. Hao, W., Yongfeng C., Hong,S., Clustering Analysis Based on Watershed Transform, *Proceedings of CCSP'2005*, pp.375-378, August 2005
8. Grimaud, M., A New Measure of Contrast: Dynamics, in *Proc. Image Algebra and Morphological Processing III*, Vol. SPIE 1769, San Diego, pp. 292–305, July 1992.
9. Yongfeng C., Hong,S., Xin,X., One Novel and Efficient Multi-level Thresholding Method, *Proceedings of SPIE*, Vol 5286, No.1, pp.330-333, 2003
10. Lee, J. S., Grunes, M. R., Pottier, E., Quantitative Comparison of Classification Capability: Fully Polarimetric versus Dual and Single-polarisation SAR, *IEEE Trans.Geosci.Remote Sensing*, Vol.39, No.11,pp.2343-2351, 2001

An Effective Combination Based on Class-Wise Expertise of Diverse Classifiers for Predictive Toxicology Data Mining

Daniel Neagu¹, Gongde Guo^{1,2}, and Shanshan Wang³

¹ Dept. of Computing, Univ. of Bradford, Bradford, BD7 1DP, UK
{D.Neagu, G.Guo}@Bradford.ac.uk

² Dept. of Computer Science, Fujian Normal Univ., Fuzhou, 350007, China

³ Dept. of Computer Science, Nanjing Univ. of Aeronautics and Astronautics, 210016, China
Shshwang@Nuaa.edu.cn

Abstract. This paper presents a study on the combination of different classifiers for toxicity prediction. Two combination operators for the Multiple-Classifer System definition are also proposed. The classification methods used to generate classifiers for combination are chosen in terms of their representability and diversity and include the Instance-based Learning algorithm (IBL), Decision Tree learning algorithm (DT), Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Multi-Layer Perceptrons (MLPs) and Support Vector Machine (SVM). An effective approach of combining class-wise expertise of diverse classifiers is proposed and evaluated on seven toxicity data sets. The experimental results show that the performance of the combined classifier based on our approach over seven data sets can achieve 69.24% classification accuracy on average, which is better than that of the best classifier (generated by MLP) and four combination schemes studied.

1 Introduction

Decision making occurs in a wide range of human activities. At its broadest, the term could cover any activity in which some decision or forecast is made on the basis of currently available information, and a classifier is then some formal method for repeatedly making such judgments in new situations [1]. Various approaches to classification have been developed and applied to real-world applications for decision making. Examples include probabilistic decision theory, discriminant analysis, fuzzy-neural networks, belief networks, non-parametric methods, tree-structured classifiers, and rough sets.

Unfortunately, no dominant classifier exists for all data distributions, and the data distribution of the task at hand is usually unknown. A single classifier cannot be discriminative enough if the number of classes is huge or if the available data show complex correlation. For applications where the classes of content are numerous, unlimited and unpredictable, one specific classifier cannot solve the problem with a good accuracy.

A Multiple Classifier System (MCS) is a powerful solution to difficult decision making problems involving large sets and noisy input because it allows simultaneous use of arbitrary feature descriptors and classification procedures [2]. The ultimate goal of designing multiple classifier systems is to achieve the best possible classification performance for the task at hand. Empirical studies have observed that different classifiers potentially offer complementary information about patterns to be classified, which could be harnessed to improve the overall performance [3].

Many different approaches have been developed for classifier combination. Examples include majority voting [4], entropy-based combination [5], Dempster-Shafer theory-based combination [6], [7], Bayesian classifier combination [8], similarity-based classifier combination [9], fuzzy inference [10], gating networks [11] and statistical models [2]. However, empirical studies on classifier combination have observed that most of the combined classifiers inherit both, the strength and weakness of each individual classifier, resulting in a small improvement of classification accuracy being made in such cases. An example is the combined classifier based on Dempster rule of combination which pools all the information no matter it is reliable or unreliable from different classifiers together for classification [6]. This will neutralize the efforts in improving the performance. This situation also happens to majority voting-based classifier combination as synthesizing various opinions from different classifiers takes both, positive and negative information into account. Such drawbacks motivate us to seek an effective solution for tackling the MCS problem.

2 The Proposed Effective Combination Scheme

In literature, most combination schemes focus on either final classification results (classes) or intermediate classification results (class-wise possibilities or similarities) of individual classifiers, thus inheriting both strength and weakness of each classifier. It is expected to enhance their strength and weaken their weakness when integrating them into a combined system. This motivates us to seek a way of using only the class-wise expertise of each classifier for building model for each class.

With this consideration, we propose a hybrid classifier combination scheme (see Figure 1 for more details) which makes use of class-wise expertise of diverse classifiers – a kind of priori knowledge obtained from the training set - to achieve potentially better performance. The two new combination operators (\otimes, \oplus) for the Multiple-Classifier System definition (Figure 1) are defined as follows:

- **Operator \otimes :**

$$\otimes_j = \arg \max_{i, M_i} \left\{ \frac{TP_j^i}{TP_j^i + FP_j^i} \mid i = 1, 2, \dots, m \right\}, \quad j = 1, 2, \dots, L. \quad (1)$$

where TP is an abbreviation of ‘True Positive’; TP_j^i represents the number of instances with class c_j correctly classified by classifier A_i ; FP stands for ‘False Positive’, and FP_j^i represents the number of instances incorrectly classified as c_j by classifier A_i . M_i is a model built by A_i on a given training set.

• **Operator \oplus :**

$$\oplus = \arg \max_{i, M_i} \{CA_i \mid i = 1, 2, \dots, m\} . \quad (2)$$

where CA_i stands for Classification Accuracy obtained by running classifier A_i on a given training set.

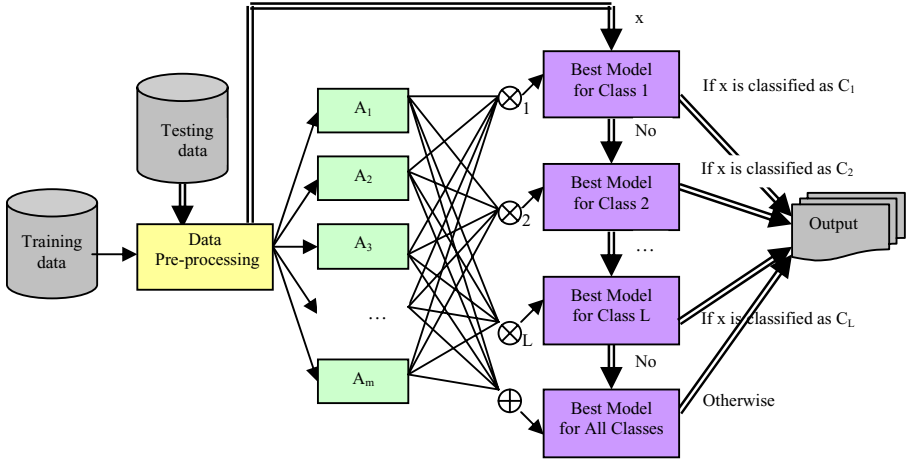


Fig. 1. Architecture of the Effective Multiple Classifier System

Based on the integrating scheme of multiple classifiers depicted in Fig. 1, the model construction and classification algorithms are described as follows:

• **Model construction algorithm:**

- 1) Given a data set D with L classes, divide it into a training set D_1 and a testing set D_2 .
- 2) For each chosen algorithm A_i ($i=1, 2, \dots, m$)
 - a) Evaluate it on D_1 using 10-fold cross validation method.

b) Calculate average accuracy CA_i and precision $P_j^i = \frac{TP_j^i}{TP_j^i + FP_j^i}$ for each class on D_1 , where $j=1, 2, \dots, L$.

c) Build a model M_i on D_1 .

- 3) Choose the best model for each class:

$$\otimes_j = \arg \max_{i, M_i} \{P_j^i \mid i = 1, 2, \dots, m\}, \quad j = 1, 2, \dots, L$$

- 4) Sort \otimes_j ($j=1, 2, \dots, L$) by descent order in terms of P_j^i values. The final order is assumed to be $\otimes'_1, \otimes'_2, \dots, \otimes'_L$ with associated classes c'_1, c'_2, \dots, c'_L respectively.

- 5) Choose the best model: $\oplus = \arg \max_{i, M_i} \{CA_i \mid i = 1, 2, \dots, m\}$
- 6) A set of obtained models $\{\otimes'_1, \otimes'_2, \dots, \otimes'_L, \oplus\}$ defines the final model.

• **Classification Algorithm:**

Given a set of models $\{\otimes'_1, \otimes'_2, \dots, \otimes'_L, \oplus\}$ obtained in the model construction stage and a new instance $x \in D_2$ to be classified, the classification process of x is described as follows:

- 1) Try the models one by one in the order of $\{\otimes'_1, \otimes'_2, \dots, \otimes'_L\}$ to find the first model \otimes'_i which classifies x as its model associated class c'_i . x is finally classified as c'_i .
- 2) If no any model classifies x as its model associated class c'_i , then let model \oplus decide the class of x .

3 Description of Case Studies and Experimental Results

In this section, a brief description of data sets from predictive toxicology used in our case studies to prove the validity of our approach is given. We also provided an introduction of the chosen classifier algorithms and the description of the combination schemes used to compare with the results of our proposed approach. The experimental results are discussed at the end of this section.

3.1 Data Sets

For the purpose of evaluation of the proposed combination scheme, seven data sets from predictive toxicology domain are chosen for evaluation. Among these data sets, five of them, i.e. TROUT, ORAL_QUAIL, DAPHNIA, DIETARY_QUAIL and BEE have been developed by the DEMETRA project [12]; APC data set is proposed by CSL [13]; Phenols data set comes from TETRATOX database [14]. Random division of each data set into a training set (around 70%) and a testing set (around 30%), evenly for each class, was carried out before evaluation. General information about these data sets is given in Table 1.

Table 1. General information about toxicity data sets

Data sets	NI	NF_FS	NC	CD	CD_TR	CD_TE
TROUT	282	22	3	129:89:64	109:74:53	20:15:11
ORAL_QUAIL	116	8	4	4:28:24:60	3:24:19:51	1:4:5:9
DAPHNIA	264	20	4	122:65:52:25	105:53:43:21	17:12:9:4
DIETARY_QUAIL	123	12	5	8:37:34:34:10	7:31:28:29:8	1:6:6:5:2
BEE	105	11	5	13:23:13:42:14	12:18:11:35:12	1:5:2:7:2
PHENOLS	250	11	3	61:152:37	43:106:26	18:46:11
APC	60	6	4	17:16:16:11	12:12:12:9	5:4:4:2

In Table 1, the meaning of the title in each column is as follows: NI - Number of Instances, NF_FS - Number of Features after Feature Selection using CfsSubsetEval method [15]; NC - Number of Classes; CD - Class Distribution; CD_TR - Class Distribution of TRaining set, and CD_TE - Class Distribution of TEsting set.

3.2 Classifiers

Five different types of classifiers were evaluated in this study: Instance-based Learner (IBL), Decision Tree (DT), Repeated Incremental Pruning to Produce Error Reduction (RIPPER), Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM). These classifiers, implemented in Weka software package [15], are chosen in terms of their representability and diversity.

3.3 Combination Schemes

Four combination schemes have been considered in this study as terms of comparison for the proposed MCS: Majority Voting-based Combination (MVC), Maximal Probability-based Combination (MPC), Average Probability-based Combination (APC) and Classifier Combination based on Dempster Rule of Combination (DRC). A brief introduction on MVC, MPC, and APC is provided; further details on DRC can be found in [6], [7].

1) Majority Voting-based Combination

Given a new instance x to be classified, whose true class label is $t_x \in C = \{c_1, c_2, \dots, c_L\}$ and m predefined classifiers are denoted as A_1, A_2, \dots, A_m respectively, the classifier A_i approximates a discrete-valued function $f_{A_i} : \mathcal{X}^n \rightarrow C$. The final class label of x , obtained by majority voting-based classifier combination, is described as follows:

$$f(x) \leftarrow \arg \max_{c \in C} \sum_{i=1}^m \delta(c, f_{A_i}(x)). \quad (3)$$

where $\delta(a, b) = 1$ if $a = b$, and $\delta(a, b) = 0$ otherwise.

With the same aforementioned assumption, the classification result of x classified by A_j is given by a vector of probability values of x to each class $P = \langle P_{j1}, P_{j2}, \dots, P_{jL} \rangle$, where $j = 1, 2, \dots, m$. The final class label of x can be obtained in two different ways:

2) Maximal Probability-based Combination (MPC)

$$f_1(x) \leftarrow \arg \max_{c_v \in C} \{ \max \{ P_{uv} \mid u = 1, \dots, m \} \mid v = 1, \dots, L \}. \quad (4)$$

3) Average Probability-based Combination (APC)

$$f_2(x) \leftarrow \arg \max_{c_v \in C} \left\{ \sum_{u=1}^m (P_{uv} / m) \mid v = 1, 2, \dots, L \right\}. \quad (5)$$

3.4 Experimental Results

Experimental results of the five classifiers evaluated on the aforementioned seven data sets are presented in Table 2 and 3, where C_i stands for class i and M_i stands for the chosen best model for class i . The parameter LR for MLP in Table 3 stands for learning rate; the parameter k for IBL stands for the number of nearest neighbours used for classifying new instances and the parameter C for SVM stands for the cost of SVM. Table 2 presents the best models chosen for each class over seven data sets. The classification accuracies of models created by each algorithm vary for each data set: some accuracy results are relatively poor when compared to ‘benchmark’ data sets from UCI machine learning repository using the same algorithms. The unique nature of the seven chemical data sets used in this paper can make accurate class predictions difficult, as data is often noisy and unevenly distributed across the multi-dimensional attribute space.

Table 2. The best models chosen for each class over seven data sets

Data sets	C_i	M_i	C_i	M_i	C_i	M_i	C_i	M_i	C_i	M_i
TROUT	3	MLP	2	IBL	1	MLP	/	IBL		
ORAL_QUAIL	3	IBL	2	J48	1	MLP	/	MLP		
DAPHNIA	4	MLP	2	IBL	1	JRIP	3	IBL	/	IBL
DIETARY_QUAIL	3	SVM	2	MLP	4	MLP	1	IBL	5	J48 / MLP
BEE	3	IBL	5	IBL	4	MLP	2	IBL	/	J48
PHENOLS	3	MLP	1	SVM	2	MLP	/	MLP		
APC	1	IBL	3	MLP	2	IBL	4	MLP	/	IBL

Table 3. Performance of different algorithms on seven data sets (TR:70%, TE:30%)

Data sets	MLP	LR	IBL	K	DT	RIPPER	SVM	C
TROUT	65.22	0.3	63.04	5	56.52	54.35	60.87	100
ORAL_QUAIL	47.37	0.3	47.37	5	47.37	47.37	47.37	1.0
DAPHNIA	54.76	0.3	64.29	5	45.24	57.14	52.38	100
DIETARY_QUAIL	70.00	0.9	60.00	10	47.62	40.00	55.00	1.0
BEE	58.82	0.9	70.59	1	58.82	58.82	58.82	1.0
PHENOLS	86.67	0.3	73.33	5	77.33	72.00	78.67	100
APC	53.33	0.9	53.33	5	53.33	46.67	46.67	100
Average	62.31	/	61.71	/	55.18	53.76	57.11	/

These properties of the data sets can make creating accurate models difficult. As shown in Table 3, some algorithms appear more suitable for particular data sets (IBL for BEE and DAPHNIA, MLP for PHENOLS and DIETARY_QUAIL); exhibiting higher than average accuracy compared to their performance across all seven data sets. This implies that careful selection of algorithms can make the creation of accurate models more successful. However, there is not a general approach to identify most appropriate models to particular data sets, especially when data can be updated and change its range and also imply further model development.

Table 4. Comparison of performance of combination schemes on seven data sets

Data sets	CSCEDC	MVC	MPC	APC	DSC
TROUT	76.09	67.39	52.17	65.22	60.87
ORAL_QUAIL	52.63	47.37	52.63	47.37	47.37
DAPHNIA	66.67	54.76	52.38	52.38	52.38
DIETARY_QUAIL	65.00	60.00	45.00	55.00	50.00
BEE	70.59	52.94	70.59	64.71	64.71
PHENOLS	86.67	84.00	81.33	84.00	81.33
APC	67.00	46.67	53.33	46.67	46.67
Average	69.24	59.02	58.20	59.34	57.62

Compared to individual classifiers, the performances of traditional multi-classifier approaches MVC, MPC, APC and DSC (Table 4) have not been significantly improved. This result emphasizes the drawbacks of existing combination schemes as they inherit both the strength and weakness of each classifier, thus neutralizing the efforts in improving performance of global models. In contrary, our proposed combination scheme CSCEDC (Combination Scheme based on Class-wise Expertise of Diverse Classifiers) not only makes use of the expertise of best individual classifiers but removes their negative influences as well, therefore results presented in Table 4 show significant improvement of global performance.

4 Conclusions

Formal definition of combination operators and a methodology to develop effective Multiple-Classifier System based on individual class-wise expertise of best models following studies performed using different Machine Learning approaches have been proposed. The outcome of this study on predictive toxicology data sets proved that single classifier-based models can not be discriminative well enough on all data sets considered. The Multi-Classifier System based on the proposed combination scheme obtains for seven toxicity data sets best performance compared with any individual classifier or commonly used combination schemes studied.

The use of class-wise expertise of diverse classifiers effectively reduces the negative influence of each classifier to the combined system, thus leading to significant improvements of performance. The proposed combination scheme applied to predictive toxicology data mining justifies our hypothesis. To consolidate this work, more experiments on some benchmark data sets will be carried out.

Acknowledgment

This work is partially funded by the EPSRC project PYTHIA GR/T02508/01 (<http://pythia.inf.brad.ac.uk>). The authors acknowledge the support of the EU FP5 project DEMETRA, Dr. Qasim Chaudhry (Central Science Laboratory York, UK) and Dr. Mark Cronin (Liverpool John Moores University, UK).

References

1. Michie, D., Spiegelhalter, D. J. and Taylor, C. C. Machine Learning, Neural and Statistical Classification, Ellis Horwood, (1994)
2. Ho, T.K., Hull, J.J. and Srihari, S. N. Decision Combination in Multiple Classifier Systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 16, Issue 1, (1994) 66 - 75
3. Baykut, A. and Ercil, A. Towards Automated Classifier Combination for Pattern Recognition. Multiple Classifier Systems, Springer Verlag, 2003, T. Wideatt, Fabio Roli (eds.), (2003) 94-105
4. Nadal, C., Legault, R. and Suen, C. Y. Complementary Algorithms for the Recognition of Totally Unconstrained Hand Written Numeral. In Proc. of the 10th International Conference on Pattern Recognition, Volume A (1990) 434-449
5. Saerens, M. and Fous. F. Yet Another Method for Combining Classifiers Outputs: A Maximum Entropy Approach. In Proc. of MCS'04, the 5th International Workshop on Multiple Classifier Systems, LNCS 3077, (2004) 82-91
6. Zhang, B. and Srihari, S.N. Class-Wise Multi-Classifier Combination Based on Dempster-Shafer Theory. In Proc. of ICARV'02, (2002)
7. Bi, Y., Bell, D., Wang, H., Guo, G. and Greer, K. Combining Multiple Classifiers Using Dempster-Shafer's Rule for Text Categorization. In Proc. of MDAI'04, LNCS 313/2004, (2004) pp.127-138
8. Xu, L., Krzyzak, A. and Suen, C. Methods of Combination Multiple Classifiers and Their Applications to Handwritten Recognition. IEEE Transactions on Systems, Man and Cybernetics, SMC-22(3): 418-435, May/June 1992.
9. Guo, G. and Neagu, D. Similarity-based Classifier Combination for Decision Making, IEEE International Conference on Systems, Man and Cybernetics (SMC'05) Hawaii, USA, (2005) 176-181
10. Neagu, D. and Palade, V. Modular Neuro-Fuzzy Networks: An Overview of Explicit and Implicit Knowledge Integration. In Proc. of the 15th International FLAIRS-02 Conference, Special Track on Integrated Intelligent Systems, 14-16 May 2002, Pensacola, Florida, USA, AAAI Press, (2002) 277-281
11. Kuncheva, L. I. Combining Classifiers: Soft Computing Solutions. In: S.K. Pal (Eds.) Pattern Recognition: From Classical to Modern Approaches, World Scientific, Singapore, (2001) 427-452
12. EU FP5 Quality of Life DEMETRA QLRT-2001-00691: Development of Environmental Modules for Evaluation of Toxicity of pesticide Residues in Agriculture (<http://www.demetra-tox.net>).
13. CSL (Central Science Laboratory York): Development of Artificial Intelligence-based In-silico Toxicity Models for Use in Pesticide Risk Assessment, 2004-2007 (<http://www.csl.gov.uk>).
14. Schultz, T.W. TETRATOX: Tetrahymena Pyriformis Population Growth Impairment Endpoint - A Surrogate for Fish Lethality. Toxicol. Methods 7: (1997) 289-309
15. Witten, I.H. and Frank, G. Data Mining: Practical Machine Learning Tools with Java Implementations, Morgan Kaufmann, San Francisco (2000)

Robust Collective Classification with Contextual Dependency Network Models[★]

Yonghong Tian¹, Tiejun Huang^{1,2}, and Wen Gao^{1,2}

¹ Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China

² Digital Media Institute, Peking University, Beijing 100080, China

{yhtian, tjhuang, wgao}@jd1.ac.cn

Abstract. In order to exploit the dependencies in relational data to improve predictions, relational classification models often need to make simultaneous statistical judgments about the class labels for a set of related objects. Robustness has always been an important concern for such collective classification models since many real-world relational data such as Web pages are often accompanied with much noisy information. In this paper, we propose a contextual dependency network (CDN) model for classifying linked objects in the presence of noisy and irrelevant links. The CDN model makes use of a dependency function to characterize the contextual dependencies among linked objects so that it can effectively reduce the effect of irrelevant links on the classification. We show how to use the Gibbs inference framework over the CDN model for collective classification of multiple linked objects. The experiments show that the CDN model demonstrates relatively high robustness on datasets containing irrelevant links.

1 Introduction

Many real-world datasets are characterized by the presence of complex relational structure: Web, bibliographic data, social networks, epidemiological records, etc. In such relational data, entities are related to each other via different types of relations (e.g., hyperlinks, citations, friendships). For classification of relational data, the relational structure can be exploited to achieve better predictions. Often, relational classification models need to make simultaneous statistical judgments about the class labels for a set of related objects, rather than classifying them separately. Clearly, such collective classification models are capable of significantly improving probabilistic inference in relational data [1].

Recently, some combinative relational classification (CRC) algorithms (e.g., [4][5][6][16][17]) have been proposed for classification of link data by integrating relational feature generation into traditional machine learning algorithms. Due to the implementation simplicity, CRC algorithms are often used as the baselines for classification of link data. For example, this paper uses neighborhood iterative classification (NIC)[4] and linkage logistic regression (LLR)[6] as baseline link-based models. Several researchers also proposed statistical relational models (SRMs) to characterize the correlation between link data, e.g., probabilistic relational models (PRMs)[7], probabilistic entity-relationship models (PERs)[15], relational Markov networks (RMNs)[2], Markov logic

[★] This work is supported by China-America Digital Academic Library project (grant No. CADAL2004002).

networks (MLNs)[10] and relational dependency networks (RDNs)[8][9]. These models allow the specification of a probability model for types of objects, and also allow attributes of an object to depend probabilistically on attributes of other related objects. Thereinto, RDNs offer several advantages over the other models, including the interpretable representation that facilitates knowledge discovery in relational data, the ability to represent cyclic dependencies, and the simple but efficient methods for learning model structure and parameters [9].

However, the link structure in real world is more complex. Links can be from an object to another object of the same topic, or they can point at objects of different topics. The latter are sometimes referred to as "noise" when they do not provide useful and predictive information for categorization. To perform robust reasons in such "noisy" data sets, this paper proposes a contextual dependency network (CDN) model. Similar to RDNs, the CDN model also uses dependency networks (DNs) [11] for modeling relational data. On top of the DN framework, we introduce additional parameters called dependency functions to directly capture the strengths of dependencies among linked objects. In this way, CDNs can effectively reduce the effect of the irrelevant links on the classification. Moreover, we also show how to use the Gibbs inference framework over the learned CDN model for collective classification of multiple linked objects. Experiments were performed on Cora and WebKB to compare the classification performance of CDNs with RDNs and two baseline link-based models. The experimental results indicate that the CDNs can scale well in the presence of noise.

We present the formulation, learning and inference of CDNs in Section 2. Experiments and results are presented in Section 3. We conclude the paper in Section 4.

2 Contextual Dependency Network Model

2.1 Relational Data

In general, link data can be viewed as an instantiation of a relational schema \mathcal{S} where entities are interconnected. A schema specifies a set of object types \mathbf{T} . Each object type $t \in \mathbf{T}$ is associated with a set of attributes. Moreover, a link dataset can be represented as a directed (or undirected) graph $\mathcal{G}_D = (\mathcal{O}_D, \mathcal{L}_D)$, where $o_i \in \mathcal{O}_D$ the node denotes an object (e.g., authors, papers) and the edge $o_i \rightarrow o_j \in \mathcal{L}_D$ denotes a link from o_i to o_j (e.g., author-of, citation). Clearly, attributes of an object can depend probabilistically on its other attributes, and on attributes of other linked objects in \mathcal{G}_D .

The link regularities in many real-world data are very complex. That is, many real-world link data such as Web pages may well exhibit the "partial co-referencing" regularity, i.e., objects with the same class tend to link to objects that are semantically related to each others, but also link to a wide variety of other objects without semantic reason [3]. Clearly, links are less informative in this case, but sometimes also provide additional information for the classification of the objects in question. Instead of eliminating these links outright, the approach that we take in CDN is to weigh these links differently through dependency functions that can be learn from the training data set.

2.2 Model Formulation

Dependency networks (DNs) [11] are probabilistic graphical models that are similar to Bayesian Networks (BNs). They differ in that the graphical structures of DNs are not required to be acyclic. A DN $\mathcal{D}=(\mathcal{G}, \mathbf{P})$ encodes the conditional independence constraints that each variable is independent of all other variables in \mathbf{X} given its parents, where the direct graph \mathcal{G} encodes the dependency structure and \mathbf{P} is a set of conditional probability distributions (CPDs) satisfying $p(X_i|\mathbf{Pa}_i) = p(X_i|\mathbf{X}\setminus X_i)$ for each $X_i \in \mathbf{X}$ (\mathbf{Pa}_i denotes the parents of X_i). An advantage of DNs is that for both structure learning and parameter estimation, the CPD for each variable can be learned independently using any standard classification or regression algorithm [11].

By simply extending DNs to a relational setting, RDNs [8][9] use a bidirected model graph \mathcal{G}_M with a set of CPDs \mathbf{P} . Each node in \mathcal{G}_M corresponds to an attribute A_i^t and is associated with a CPD $p(a_i^t|\mathbf{Pa}(a_i^t))$. The RDN learning algorithm is much like the DN learning algorithm, except it uses relational probability trees (RPTs) to learn CPDs [9]. However, the link structure is not a part of the probabilistic model of RDNs, thus we cannot predict links and more importantly use the links to improve prediction about other attributes in the model [7].

Instead of specifying a single CPD for the class label of an object with type given both other attributes of that object and attributes of other related objects (as in RDNs), CDNs define two CPDs: one for capturing *intrinsic dependency* and the other for capturing *relational dependency*. More formally,

$$P(C_i|\mathbf{Pa}(C_i)) = \alpha_t P(C_i|\mathbf{Pa}^{(L)}(C_i)) + (1 - \alpha_t) P(C_i|\mathbf{Pa}^{(N)}(C_i)), \quad (1)$$

where $\mathbf{Pa}^{(L)}(C_i)$ denotes the ‘‘local’’ parents of C_i (i.e., attributes in $\mathbf{Pa}^{(L)}(C_i)$ are associated with object o_i), $\mathbf{Pa}^{(N)}(C_i)$ denotes the ‘‘networked’’ parents of (C_i) (i.e., attributes in $\mathbf{Pa}^{(N)}(C_i)$ are associated with objects in \mathcal{G}_D that are related to o_i). For convenience, we refer to $\mathbf{Pa}^{(L)}(C_i)$ as *intrinsic* CPDs, $\mathbf{Pa}^{(N)}(C_i)$ as *relational* CPDs, and accordingly $\mathbf{Pa}(C_i)$ as *full* CPDs or directly CPDs for short. Parameter α_t is a scalar with $0 \leq \alpha_t \leq 1$ to capture the strength of the intrinsic dependency for objects of each type $t \in \mathbf{T}$. Moreover, CDNs introduce some parameters, called *dependency functions*, to directly capture the different strengths of contextual dependencies among linked objects such that the relational CPD $\mathbf{Pa}^{(N)}(C_i)$ is expressed as

$$P(C_i|\mathbf{Pa}^{(N)}(C_i)) = \sum_{o_{ik} \in \mathbf{Pa}(o_i)} \sigma_{i,ik} P(C_i|\mathbf{Pa}_{ik}^{(N)}(C_i)), \quad (2)$$

where $\mathbf{Pa}_{ik}^{(N)}(C_i)$ is the parent set of C_i in attributes of object $o_{ik} \in \mathbf{Pa}(o_i)$, and $\sigma_{i,ik}$ is the dependency function of o_i on o_{ik} , which is used to measure how much $\mathbf{Pa}_{ik}^{(N)}(C_i)$ affects the distribution of C_i . Here we assume that a function $\sigma_{i,ik}$ is called a *dependency function* of object o_i on object $o_{ik} \in \mathcal{O}_D$ if it satisfies: (1) $\sigma_{i,ik} \geq 0$; (2) $\sum_{o_{ik} \in \mathbf{Pa}(o_i)} \sigma_{i,ik} = 1$; (3) The function $\sigma_{i,ik}$ consists of at least two components: the mutual information $I(o_i; o_{ik})$ and the linkage kernel $K(o_i, o_{ik}) = f(\varphi_{i,ik})$. Several oft-used kernel functions (e.g., polynomial, exponential, or sigmoid functions) can be adopted to construct linkage kernels from the link features $f(\varphi_{i,ik})$ between o_i and o_{ik} [14], given a parameter β_i for each type

$t \in \mathbf{T}$ of objects. Here we use mutual information $I(o_i; o_{ik})$ to measure the statistically semantic correlation among o_i and o_{ik} . The higher the $I(o_i; o_{ik})$, the easier it is to estimate one object given the other, or vice versa. Thus we have the following definition:

Definition 1. For the relational schema \mathcal{S} , a CDN model $\mathcal{M}=(\mathcal{G}_M, \mathbf{P}, \theta)$ defines:

1. a directed model graph \mathcal{G}_M in which each node corresponds to an attribute of objects with type $t \in \mathbf{T}$ and each edge represents the dependency among attributes,
2. a set of template CPDs $\mathbf{P}=\mathbf{P}^{(L)} \cup \mathbf{P}^{(N)}$ where $\mathbf{P}^{(L)}$ and $\mathbf{P}^{(N)}$ are the intrinsic and relational CPDs respectively, and
3. a parameter set $\theta=\{\alpha_t, \beta_t, \pi_t, a_{i,j}^{t,t'}\}_{t \in \mathbf{T}}$ that are used to specify dependency functions among linked objects in any link graph that is defined by the schema \mathcal{S} , where $\pi_t=\{p_i^t = P(c_i^t)\}$ are the priors, and $\{p(c_i^t|c_j^t) \mid t' \in \mathbf{T}\}$ are the transition probabilities.

For a given link graph \mathcal{G}_D , a CDN model uses the \mathcal{G}_M and \mathcal{G}_D to instantiate an inference graph $\mathcal{G}_I=(\mathcal{V}_I, \mathcal{E}_I)$ during inference so as to represent the probabilistic dependencies among all variables in a test set [9]. Figure 1 shows an example of \mathcal{G}_M and \mathcal{G}_I . Given a CDN model \mathcal{M} , the full joint distribution over the unknown label variables in \mathcal{G}_D can be approximately expressed as follows:

$$\begin{aligned}
 P(\mathcal{G}_D|\mathcal{M}) &= \prod_{t \in \mathbf{T}} \prod_{o_i \in \mathbf{I}(t)} P(C_i | \mathbf{Pa}(C_i)) \\
 &= \prod_{t \in \mathbf{T}} \prod_{o_i \in \mathbf{I}(t)} \left[\alpha_t P(C_i | \mathbf{Pa}^{(L)}(C_i)) + (1 - \alpha_t) P(C_i | \mathbf{Pa}_{ik}^{(N)}(C_i)) \right] \\
 &= \prod_{t \in \mathbf{T}} \prod_{o_i \in \mathbf{I}(t)} \left[\sum_{o_{ik} \in \{o_i\} \cup \mathbf{Pa}(o_i)} \tilde{\sigma}_{i,ik} P(C_i | \mathbf{Pa}_{ik}^*(C_i)) \right], \tag{3}
 \end{aligned}$$

where

$$\tilde{\sigma}_{i,ik} = \begin{cases} \alpha_t, & o_{ik} = o_i, \\ (1 - \alpha_t)\sigma_{i,ik}, & o_{ik} \in \mathbf{Pa}(o_i), \\ 0, & \text{otherwise.} \end{cases}$$

and

$$\mathbf{Pa}_{ik}^*(C_i) = \begin{cases} \mathbf{Pa}_{ik}^{(L)}(C_i), & o_{ik} = o_i, \\ \mathbf{Pa}_{ik}^{(N)}(C_i), & o_{ik} \in \mathbf{Pa}(o_i). \end{cases}$$

CDNs first approximate the full joint distribution for a collection of related objects with a set of CPDs. Then each CPD can be further modeled as a linear combination of an intrinsic CPD and a set of relational CPDs with the weights represented by dependency functions. This would facilitate ease of knowledge acquisition and domain modeling, and provide computational savings in the inference process.

2.3 Learning

Like DNs, both the structure and parameters of CDNs are determined through learning a set of CPDs. For a CDN model, the parameter-estimation task is to learn a set of

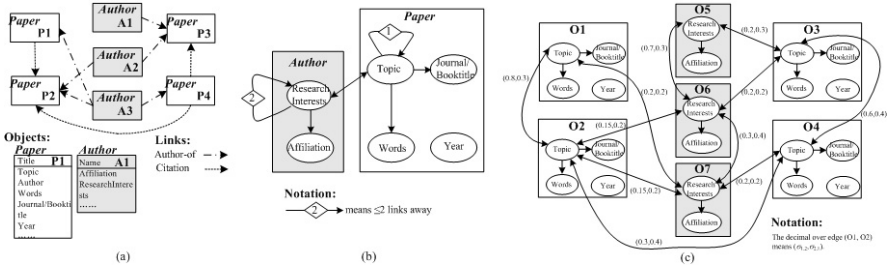


Fig. 1. (a) A link graph, (b) the model graph and (c) the inference graph

model parameters $\{\mathbf{P}_t^{(L)}, \mathbf{P}_t^{(N)}, \alpha_t, \beta_t, \pi_t\}_{t \in \mathbf{T}}$ from a training set $\mathcal{G}'_D = (\mathcal{O}'_D, \mathcal{L}'_D)$, where $\mathbf{P}^{(L)}$ and $\mathbf{P}^{(N)}$ are the intrinsic CPDs and relational CPDs respectively, α_t is the self-reliant factor, β_t is the parameter for linkage kernels, $\pi_t = \{p_t^i = P(c_i^t)\}$ are the priors. The learned parameters are then applied to a separate testing set \mathcal{G}_D . Note that the transition probabilities $a_{i,j}^{t,t'}$ can be obtained by using the intrinsic and relational CPDs.

The CDN learning algorithm is in principle based on pseudo-likelihood techniques, which avoids the complexities of estimating a full joint distribution. With the assumption that the objects in \mathcal{O}'_D are independent, the prior p_t^i can be estimated by the relative frequencies of objects with the class label c_i^t in \mathcal{O}'_D , and the intrinsic CPDs $\mathbf{P}_t^{(L)}$ can be estimated by any probabilistic classification or regression techniques (called *intrinsic models*) such as naïve Bayes (NB) (e.g., [4]), logistic regression (e.g., [6]), or probabilistic support vector machine (SVM) (e.g., [12]). For the relational CPDs $\mathbf{P}^{(N)}$, however, we cannot directly use the standard statistical learning methods since the labels of objects are correlated. By modeling the learning of $\mathbf{P}^{(N)}$ as a dynamic interacting process of multiple Markov chains, here we use the self-mapping transformation algorithm [13] to learn $\mathbf{P}^{(N)}$. That is, the graph \mathcal{G}'_D is partitioned into N' subgraphs, each of which contains an object o_i and its parents $\mathbf{Pa}(o_i)$. For each subgraph $\mathcal{G}'_{D_i} = (\mathcal{O}'_{D_i}, \mathcal{L}'_{D_i})$, the relational CPD parameter can be learned by using the self-mapping transformation [13]. This process is repeated for all subgraphs until convergence. Lastly, for the parameters α_t and β_t , we can set the appropriate values empirically or by the cross-validation method. For example, α_t is set to be 0.7~0.8 for type=paper and 0.4~0.5 for type=author in the citation data.

2.4 Inference

During inference, a CDN model uses the \mathcal{G}_M and \mathcal{G}_D to instantiate an inference graph \mathcal{G}_I . This process includes two operations: (1) Each object-attribute pair gets a separate, local copy of the appropriate CPD (including an intrinsic CPD and a relational CPD). (2) Calculate the dependency functions using the parameter set θ of the CDN model.

In general, the CDN inference graph can be fairly complex. Clearly, exact inference over this complex network is impractical, so we must resort to approximate inference. As in DNs and RDNs, we also use ordered Gibbs sampling for approximate inference over CDN inference graphs. First, a bootstrap step is used to assign an initial label for each unlabeled object using only the intrinsic models. That is, $p(C_i | \mathcal{M})$ can be initial-

ized as $p(C_i|\mathbf{Pa}^{(L)}(C_i))$ and an initial CDN inference graph $\mathcal{G}_1^{(0)}$ can be constructed over the link graph \mathcal{G}_D . Gibbs inference then proceeds iteratively to estimate the joint posterior distribution over the class variables of all unlabeled objects. For each variable, the *influence propagation* step is performed to return a refined posterior probability $p(C_i|\mathcal{M})$ given both other attributes of that object (i.e., $\mathbf{Pa}^{(L)}(C_i)$) and attributes of other related objects (i.e., $\mathbf{Pa}^{(N)}(C_i)$). This process is repeated for each unknown variable in the graph \mathcal{G}_1 . After a sufficient number of iterations, the values will be drawn from a stationary distribution [11]. This paper adopts a mixed iteration convergence criteria for Gibbs inference, including the convergence of the log-likelihood over all unobserved label variables, the consistency of the maximum a posterior (MAP) estimates among two consecutive iterations, and a predefined iteration upper bound.

3 Experiments

In this paper, we used two real-world datasets, i.e., Cora and WebKB, each of which can be viewed as a link graph. The whole Cora dataset consists of about 37,000 papers. In common with many other works (e.g., [6]), we use the subset (denote by Cora_0) with 4331 papers of Machine Learning and 11,873 citations. Moreover, several extended datasets, denoted by Cora_δ , are constructed by adding into Cora_0 different amounts of miscellaneous links that point from Cora_0 to papers with other topics. On the other hand, the WebKB dataset contains approximately 4100 pages from four computer science departments, with a five-valued type attribute (i.e., faculty, student, project, course and other), and 10,400 links between pages. Similarly, the base subset of pages with the four labels is denoted by WebKB_0 . We construct several extended sets WebKB_δ by appending some links that point to other pages. With different δ values, the Cora_δ and WebKB_δ datasets may exhibit different link regularities. For simplicity, we set $\{0, 0.05, 0.10, 0.15, 0.18\}$ to values in for the two datasets.

In [14], we have shown that noisy links have high influence on link-based classification. Here our main goal is to demonstrate the robustness of our CDN model in collective classification on noisy datasets. We also use NBs and SVMs as the baseline intrinsic models, and use NICs and LLRs as the baseline link-based models. In addition, we re-construct all the link-based models respectively with NBs and SVMs as their intrinsic models. For convenience, they are denoted by NIC_{NB} , NIC_{SVM} , LLR_{NB} , LLR_{SVM} , RDN_{NB} , RDN_{SVM} , CDN_{NB} and CDN_{SVM} respectively. The experimental results are shown in figure 2.

On average, CDN_{NB} outperformed NIC_{NB} , LLR_{NB} and RDN_{NB} respectively by about 6.15%, 6.08% and 5.62% on WebKB, and about 1.46%, 1.73% and 1.13% on Cora; CDN_{SVM} outperformed NIC_{SVM} , LLR_{SVM} and RDN_{SVM} respectively by about 3.33%, 3.94% and 2.71% on WebKB, and about 2.32%, 2.51% and 1.57% on Cora. More importantly, the relative accuracies of CDNs do not decline along with increasing the parameter δ for the two datasets. In other words, CDNs can effectively exploit the miscellaneous links to improve the classification performance. Comparatively, although RDNs can use the selective relational classification algorithms (e.g., RPTs) to learn a set of CPDs, their performance is also affected by the noisy links in the inference phase. This enlightens us that the selectivity of link features should be directly encoded in the

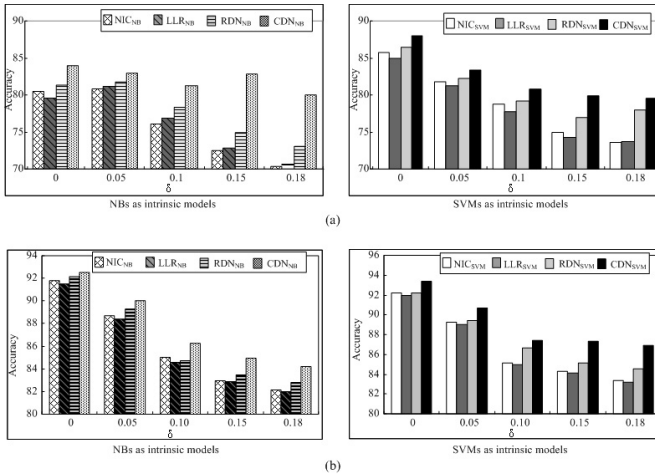


Fig. 2. Comparison of classification accuracies of all link-based models on (a) WebKB and (b) Cora

relational model itself such that the learned model can keep robust in different link data. We also noted one exception that the CDN models performed poorly on WebKB₅. This indicates that the accuracy improvements of CDNs might be not significant when the datasets have only fewer noisy links.

The differences in accuracy between the NIC, LLR, RDN and CDN models may indicate that the improvements are not significant. To investigate this possibility we performed two-tailed, paired t-tests to assess the significance of the results obtained from the four-validation tests. With a few exceptions, CDNs outperform NICs and LLRs at the 90% (averagely 97.7%) significance level on both WebKB and Cora, and outperform RDNs at the 80% (averagely 93.6%) significance level on the two datasets. These results support our conclusions that the classification performance of CDNs is significantly better than NICs, LLRs and RDNs.

Currently, we are experimenting with Web image classification tasks to explore more interesting applications of the RDN models. Our basic premise is that Web images which are co-contained in the same pages or contained in co-cited pages are likely to be related to the same topic. We thus can build a robust image classification model by using visual, textual and link information. On a sports Web image collection crawled from Yahoo!, the CDN model obtained about 14% improvements in the average classification accuracy over the SVM classifier that uses visual and textual features.

In summary, the experimental results are generally positive, but in some cases the improvements are not so significant. However, we can safely conclude that the CDN models show relatively high robustness in the link data with a few noisy links.

4 Conclusion

Many link data such as Web pages are often accompanied with a few noisy links. Such noisy links do not provide the predictive information for categorization. To capture

such complex regularities in link data, this paper proposes a robust model for collective classification, i.e., contextual dependency network (CDN) model. Experimental results showed that the CDN model can demonstrate high robustness in the noisy link datasets, and provide good prediction for the attributes of linked objects.

References

1. Jensen, D., Neville, J. and Gallagher, B.: Why collective inference improves relational classification. Proc. 10th ACM Int'l Conf. on Knowledge Discovery and Data Mining (2004) 593–598
2. Taskar, B., Abbeel, P., Koller, D.: Discriminative Probabilistic Models for Relational Classification. Proc. of Uncertainty on Artificial Intelligence, Edmonton, Canada (2002) 485–492
3. Yang, Y., Slattery, S. and Ghani, R.: A Study of Approaches to Hypertext Categorization. J. Intelligent Information system **2/3** (2002) 219–241
4. Chakrabarti, S., Dom, B. and Indyk, P.: Enhanced Hypertext Categorization Using Hyperlinks. Proc. of SIGMOD'98 (1998) 307–318
5. Neville, J., Jensen, D., Friedland, L. and Hay, M.: Learning relational probability trees. Proc. 9th ACM Int'l Conf. on Knowledge Discovery and Data Mining (2003) 625–630
6. Lu Q. and Getoor, L.: Link-based Classification. Proc. 12th Int'l Conf. on Machine Learning (2003) 496–503
7. Friedman, N., Koller, D. and Taskar, B.: Learning Probabilistic Models of Relational Structure. J. Machine Learning Research (2002) 679–707
8. Neville, J. and Jensen, D.: Collective Classification with Relational Dependency Networks. Proc. 2nd Multi-Relational Data Mining Workshop in KDD-2003 (2003)
9. Neville, J. and Jensen, D.: Dependency Networks for Relational Data. Proc. IEEE Int'l Conf. on Data Mining (2004) 170–177
10. Richardson, M. and Domingos, P.: Markov Logic Networks. Machine Learning **26(1-2)** (2005) 107–136
11. Heckerman, D., Chickering, D., Meek, C., Rounthwaite, R. and Kadie, C.: Dependency Networks for Inference, Collaborative Filtering, and Data Visualization. J. Machine Learning Research **1** (2001) 49–75
12. Sollich, P.: Probabilistic methods for Support Vector Machines. Proc. Advances in Neural Information Processing Systems 12 (2000), MIT Press, 349–355
13. Zhong, S. and Ghosh, J.: A New Formulation of Coupled Hidden Markov Models. Tech. Report, Dept. of Electronic and Computer Engineering, U. of Texas at Austin, USA, (2001)
14. Tian, Y. H., Huang, T. J., Gao, W.: Latent linkage semantic kernels for collective classification of link data. J. Intelligent Information Systems, In Press, (2006)
15. Heckerman, D., Meek, C., Koller, D.: Probabilistic Models for Relational Data, Tech. Report, MSR-TR-2004-30, Microsoft Research (2004)
16. Uwents, W. and Blockeel, H.: Classifying relational data with neural networks. Proc. 15th Int'l Conf. on Inductive Logic Programming, Bonn, Germany (2005) 384–396
17. Neville, J., Jensen, D. and Gallagher, B.: Simple estimators for relational Bayesian classifiers. Proc. 3rd IEEE Int'l Conf. on Data Mining (2003) 609–612

User-Centered Image Semantics Classification

Hongli Xu, De Xu, and Fangshi Wang

School of Computer & Information Technology, Beijing Jiaotong University,
Beijing, China 100044

hlxu@center.njtu.edu.cn

xd@computer.njtu.edu.cn

wfs@computer.njtu.edu.cn

Abstract. In this paper, we propose a multiple-level image semantics classification method. The multiple-level image semantics classifier is constructed according to a hierarchical semantics tree. A semantics tree is defined according to the individual user's habit of managing files. So it is personalized. The classification features are selected by calculating information entropy of images. The hierarchical classifier is constructed according to a class correlation measure. This measure considers both the relation of the classifiers between different hierarchical levels and the relation between the classifiers at the same level. The unlabelled pictures can be classified top-down and assigned to corresponding class and semantic labels. In our experiment binary SVM is used. The hierarchical classifier is built by selecting meta-classifiers with the combinations that have better performance. The result shows that the hierarchical classifier is more effective than a flat method.

1 Introduction

Increasing use of Internet and digital cameras leads to people having large personal collections of digital pictures. People may want to build computer-based systems to manage their pictures for easy browsing and retrieval. When people access their multimedia collections by expressing certain knowledge, the main problem is how to close the gap between low-level features and human knowledge. Image semantics annotation has been the key to bridge the gap between low-level features and high-level semantics. In the methods of image semantics annotation, most of the studies have been focusing on expert-centered model [1], which assumed domain expert defined image meanings. However, cultural difference and individual demands are not considered. Multimedia systems should use cultural clues during interaction, as well as during analysis [2]. However, expert-centered model may not always fit to every user. Brahmi [3] proposed a model of the interactions between user and meta-data. It is used to find users' preferences by analyzing their answers through a Relevance Feedback process. Gosselin [4] presented an approach to apply knowledge extracted through user interaction processes. The semantic annotations were integrated into the similarity matrix of the database images.

In our proposed method, image meaning is defined by user according to his/her own acknowledge and habit of managing pictures. The outcome of this process is

constructed as the multi-level semantics tree (MLST) [5]. Based on MLST, features are selected by calculating information entropy of images under a parent node. A combined classifier is constructed by a class-correlation measure. Thus classifier is personalized. In our experiment, binary SVM is used as meta-classifier. The combined classifier is built through selecting meta-classifiers with the correlation performance analysis.

The paper is organized as follows. Section 2 describes the construction of a Hierarchical Classifying System, and the definition of a multi-level semantics tree from user. Section 3 discusses the preprocessing steps including the feature selection process. In Section 4, SVM based multi-level semantics classifier is presented. The experimental results in building hierarchical semantics classifier are also presented. Finally, we conclude the application of the multi-level semantics classification and discuss the further work.

2 Multi-Level Semantics Tree from User (MLST)

Studies in cognition and human vision have shown that humans have been accustomed to managing files in term of the category hierarchy. If we analyze the hierarchical structure, we can discover the relation between levels: lower level category inherits higher-level category meaning, and only the same level category need be classified. For example, inflorescence and flower possess the flowerage' meaning. Therefore, when we extract lower level category semantics, we may discard the redundant features, and select the best discriminating features for classifier. User can build MLST by himself, so that the classifier based on MLST is suitable for semantic varieties and user' demand.

A Multi-level Semantics Tree is a rooted unbalance B-tree. Every node $ND(Cid, SAs, OP, P)$ has four fields:

Cid is unique symbol.

SAs is a semantic attribute set (W, F, CR) including the descriptive words W , the low-level feature F , and CR is composed of the class center CC of the training set and the class correlation CSs between the children nodes of the same parent node.

OP is semantic operation, executing the classification algorithm.

P is a pointer vector, which describes the relation between the parent-child nodes.

In the Algorithm 1, we have illustrated the basic steps for constructing the MLST.

Algorithm 1. Create Multi-level Semantics Tree (MLST)

Input: the category hierarchy and the image set $\{I_{ij}\}$

Output: MLST.

$ND_{ij}(Cid, SAs_{ij}, OP_{ij}, P_{ij})$ at every node.

Initialize Cid, SAs, P , feature subset;

for each node ND_{ij} **do** // from root down to leaf nodes at every level of the MLST

input the descriptive words W_{ij} ;

set P_{ij} the node pointer;

if the node of ND_{ij} is leaf node **Then**

```

input the image sample set  $I_{ij}$  ;
calculate  $CC_{ij}$  the center of the image set  $I_{ij}$  ;
calculate  $CSS_{ij}$  the class correlation between the
        children nodes of the same parent node;
 $F_{ij} \leftarrow$  Feature Selection() between child nodes of the
        same parent at the level- $i$ ;
 $OP_{ij} \leftarrow$  Train Classifier() using the class correla-
        tion measure;
End for

```

For the MLST, every node represents an image semantic class; user defines the descriptive words of every node. Its inserting and deleting node is similar to B-tree. But, CSS_{ij} need recalculate.

3 Feature Selection Based on Information Entropy

The different semantic image should be represented by different feature collection [5]. According to different evaluation criteria, feature selection algorithms mainly include the filter model, the wrapper model, and the hybrid model [6]. The wrapper model tends to give superior performance as it find features better suited to the predetermined learning algorithm, but it also is computationally more expensive. When the training data set becomes very large, the filter model is usually a good choice due to its computational efficiency and neutral bias toward any learning algorithm. In our proposed classification, the filter model is adopted. The information measure is used to the independent criterion in the filter algorithms. We calculate the difference of the entropy of same kind feature between two classes, and select the bigger. The difference of image entropy is defined as:

$$Dist(F_{ij}, F_{i,j+1}) = \left| H(F_{ij}) - H(F_{i,j+1}) \right| \quad (1)$$

$$H(F_{ij}) = - \sum_l P(f_l) \log_2(P(f_l)) \quad (2)$$

Algorithm 2. Feature Selection

```

Input: training set  $TS(F_1, F_2 \dots F_n, C)$ , kinds of selected fea-
        ture subset
Output: the selected feature subset  $FS(F_1, F_2 \dots F_m, C)$ 
Initialize the kind of selected feature subset  $M$ ,
        the selected feature subset  $FS \leftarrow \{\}$ 
Set the undetermined feature set  $F \leftarrow F_1, F_2 \dots F_n$ 
for each kind feature  $F_i \in F$  do
Calculate the entropy difference  $Dist_{ij}$  between nodes of
        the same parent using the greedy algorithm;
Set the threshold value  $\gamma$ ;

```

Set $F \leftarrow F - \{F_i\}$ if $Dist_{ij} > \gamma$.
end for

The selected feature subset $FS(F_1, F_2 \dots F_m, C)$ for C_i

4 Combining Classifier Based on Class Correlation Measure

In the classification, the most commonly used performance measures are the classic information retrieval notions of Precision and Recall. Neither precision nor recall is useful as a performance measure separately. Therefore, the performance of the classification has often been measured by the combination of the two measures. The popular combinations need some parameters that user inputs [7]. The accuracy and error also are commonly used performance measures, denoted by AC_i and Er_i respectively. Because accuracy plus error equals 1, AC_i is enough for selecting the classifier. The contingency table for the class C_i from the node space $\{C_1, C_2, \dots, C_m\}$ is shown in Table 1. Let TP_i be the set of images correctly classified into C_i ; FP_i be the set of pictures wrongly accepted; FN_i be the set of pictures wrongly rejected and TN_i be the set of pictures correctly rejected. The standard accuracy and error are defined below:

$$Ar_i = \frac{|TP_i| + |TN_i|}{|TP_i| + |FP_i| + |FN_i| + |TN_i|} \tag{3}$$

In all the formulas presented in this paper, $|\bullet|$ is the number of elements in corresponding classifying.

Table 1. Contingency table for the class C_i

Category C_i		Expert Judgments	
		Yes	No
Classifier Judgments	Yes	TP_i	FP_i
	No	FN_i	TN_i

In the traditional performance measure, if a picture is misclassified, it is discarded. So, in the hierarchical classification, one of its obvious problems is that a misclassification at a parent node may force a picture to be discarded before it can be classified into the children nodes. For degrading misclassified ratio, we define the class similarity between two classes C_i and C_j , denoted by $CS(C_i, C_j)$, which reflects how “close” the two classes are in terms of semantic. Let d_k be a test image, if d_k is misclassified by C_i and is accepted by C_r , $CS(d_k, C_i)$ can be calculated as the following.

$$CS(d_k, C_i) = \frac{ACS_i - CS(C_r, C_i)}{ACS_i} \quad (4)$$

ACS_i is the Average Class Similarity about C_i . The accuracy based on class similarity is defined below:

$$Ar_i = \frac{|TP_i| + |TN_i| + |CS_i|}{|TP_i| + |FP_i| + |FP_i| + |FN_i|} \quad (5)$$

$$|CS_i| = \sum_k CS(d_k, C_i) \quad (6)$$

Because the value of $CS(d_k, C_i)$ is between -1 and 1 , $|CS_i|$ may be less than 1 . In our method, when user inputs the categories, the class centers are computed firstly. $CS(C_i, C_k)$ is defined by the distance between C_i and C_k .

In image classification based on SVM, given SVM_Model_{ij} is *OP* about C_{ij} , the node C_{ij} is described as: $ND(Cid, SAS_{ij}, SVM_Model_{ij}, P_{ij})$. SAS_{ij} includes W_{ij} , F_{ij} , and CR_{ij} . W_{ij} is semantic words, such as flower, forest, city etc. SVM_Model_{ij} is a trained SVM classifier with corresponding semantics, and F_{ij} is low-level features of W_{ij} . F_{ij} is represented as, $\{f_1, f_2, \dots, f_m\}$ $f_i \in \{color, texture, shape, \dots, etc.\}$, and m is the sum of feature dimension. Our method is a mapping from high-level semantics to low-level feature by using SVM classifier to model SVM_Model_{ij} . The construction procedure is shown in the Fig.1.

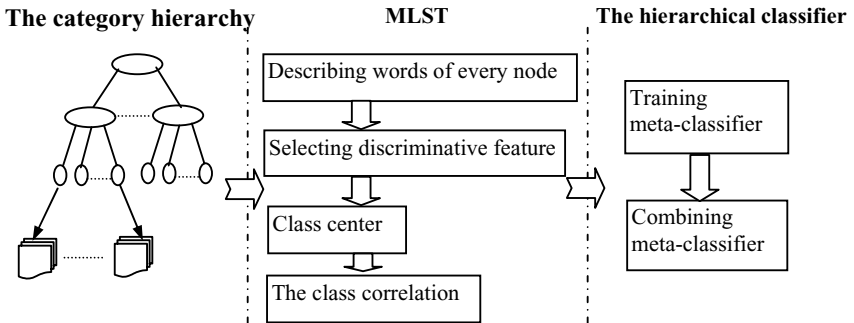


Fig. 1. Construction of a hierarchical classifying system

First, when user input a set of picture categories and the image document set, the MLST is constructed. Often, this process is done manually, but it might be as automatic procedure. Second, we train the node meta-classifiers one by one using

different features set or different classification. Finally, these meta-classifiers are combined using the new class similarity measures at every level. Actually, the final procedure is repeated until the best combining classifier is found at each level through a validation process of applying our new measures.

5 Experiment Results

5.1 Experiment Environment

In our experiments, the image database consists of 1500 images. Multi-level flower semantics samples are shown in the Fig.2. There are inflorescence, Ikebana and flower under flowerage category. There are rose, tulip, water lily and sunflower under flower category. Experimental environment includes Intel (R) 4 PC, 256M memory, Windows 2000, Microsoft Access 2000, VC++6.0.

The training and testing number of every class image are Ikebana(40,40), Flower(50,100), Inflorescence(50,80), Sky(20,20), Forest(30,45), Mountain(30,30). Table 2 shows image features in our experiment.



Fig. 2. Multi-level flower semantics of flowerage

Table 2. Abbreviations of image features

Feature	abr.	Feature	abr.
HSV Cumulative Histogram Training	HCH	RGB Moment	RM
GRAYCumulative Histogram	GCH	Co-occurrence	COO
COLOR Histogram	CH	COLOR Cumulative Histogram	CCH

Table 3. Experiment results of meta-classifiers by the classical measures

Accuracy	COO	RM/COO	HCH/COO	HCH/RM/COO	GCH/RM/COO
Inflorescence	0.89	0.85	0.81	0.85	0.70
Ikebana	0.73	0.82	0.87	0.87	0.79
Flower	0.87	0.86	0.80	0.79	0.72
Rose	0.45	0.64	0.65	0.13	0.53
Tulip	0.34	0.56	0.67	0.72	0.70
Sunflower	0.50	0.72	0.75	0.76	0.76
Water lily	0.43	0.73	0.75	0.78	0.69

5.2 Training Meta-classifiers

In order to process automatically in the future, we use the accuracy described in section 4. The different semantic classifier should be represented by different feature collection. There are three kernel functions: Polynomial kernel function, Gaussian kernel function and Sigmoid kernel function in the SVM model. We select the Gaussian kernel ($\sigma=0.5$) to train respectively every node meta-classifier under the same parent node in the MLST. The one is positive; the others are negative. The results are show in Table 3.

5.3 Constructing Hierarchical Classifier

We select different trained meta-classifiers to build the combining classifier using the class similarity measures. Table 4 and Table 5 show the experiment result of combining classifiers. For obtaining the better performance of hierarchical classifier, we arrange meta-classifiers in order of the proposed accuracy, and select the best combining. MLSC improves the traditional flat classification as shown in Table 6.

Table 4. Part experiment results of combining classifiers under flowerage node

Accuracy (Feature sets)	Inflorescence	Ikebana	Flower
1	0.83 (COO)	0.82 (HCH/COO)	0.85 (RM/COO)
2	0.76 (COO)	0.77 (HCH/COO)	0.76 (COO)
3	0.81 (COO)	0.62 (COO)	0.76 (RM/COO)
4	0.81 (COO)	0.59 (COO)	0.66 (COO)
5	0.77 (RM/COO)	0.62 (HM/RM)	0.51(RM/COO)

Table 5. Part experiment results of combining classifier under flower node

Accuracy (Feature sets)	Rose	Tulip	Sunflower	Water lily
1	0.68 (HCH/COO)	0.65 (HCH/COO)	0.82 (HCH/COO)	0.77 (GCH/CCH)
2	0.62 (CCH/RM)	0.65 (GCH/CCH)	0.78 (RM/COO)	0.67 (GCH/CCH)
3	0.23 (RM/COO)	0.65(HCH/RM/COO)	0.56 (COO)	0.67 (GCH/CCH)

Table 6. MLSC comparing with the flat classification

Accuracy (Feature sets)	Flowerage	Inflorescence	Flower	Sunflower
Flat classification	0.83 (HCH/COO)	0.44 (COO)	0.54 (RM/COO)	0.24 (HCH/COO)
MLSC	0.83 (HCH/COO)	0.83 (COO)	0.85 (RM/COO)	0.82 (HCH/COO)

6 Conclusions

In this paper, we propose an approach of multi-level semantics classification. According to multi-level semantics tree, we construct multi-level semantics classifier. Be-

cause the MLST is from user, the multi-level semantics classifier based on the MLST is individual and applicable. At every level of the MLST, Combining classifier is build using the class correlation, which guarantee the maximum precision and recall at the leaf node classifiers.

Experiment results show this method is effective and semantic retrieval can be implemented by the method. Our further work include three aspects: a) to implement the MLST for more classes of image, b) to improve the method of the low-level feature selected and consider what MPEG-7 has defined, and c) to select the different classifier model, for example, SVM classifiers with different kernels and Bayesian classifier. Finally, the hierarchical classifier should be build automatically.

References

1. Ana B. Benitez, Shih-Fu Chang. Automatic Multimedia Knowledge Discovery, Summarization and Evaluation. *IEEE Transactions on Multimedia*, 2003
2. Alejandro Jaimes Human-Centered Multimedia: Culture, Deployment, and Access. *IEEE MultiMedia* Volume 13, Issue 1(January 2006) pp.12-19
3. D. Brahmi, D. Ziou, Improving CBIR Systems by Integrating Semantic Features, 1st Canadian Conference on Computer and Robot Vision (CRV'04) May 17 - 19, 2004 pp. 233-240
4. Philippe H. Gosselin, Matthieu Cord, Semantic Kernel Updating for Content-Based Image Retrieval *IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE'04)* Miami, Florida December 13 - 15, 2004 pp. 537-544
5. Hongli XU, XU De and Sun Zhijie An Approach of Multi-Level Semantics Abstraction, *Knowledge-Based Intelligent Information and Engineering Systems: 9th International Conference, KES 2005, Melbourne, Australia, September 14-16, 2005, Proceedings, Part II*, p. 1190
6. Aleksandra Mojsilovic and Bernice Rogowitz,. Capture image semantics with low-level descriptors. *Image Processing, 2001. Proceedings. 2001 International Conference on* , Volume: 1 , 7-10 Oct. 2001 .Pages:18 - 21 vol.1
7. Aixin Sun, Ee-Peng Lim, Wee-Keong Ng, "Performance Measurement Framework for Hierarchical Text Classification," *Journal of the American Society for Information Science and Technology (JASIST)*, Vol.54, No.11, page 1014-1028, 2003.

A Performance Study of Gaussian Kernel Classifiers for Data Mining Applications

Miyoung Shin

School of Electrical Engineering and Computer Science, Kyungpook National University,
1370 Sankyuk-dong, Buk-gu, Daegu 702-701, Korea
shinmy@knu.ac.kr

Abstract. Radial basis function (RBF) models have been successfully employed to study a broad range of data mining problems and benchmark data sets for real world scientific and engineering applications. In this paper we investigate RBF models with Gaussian kernels by developing classifiers in a systematic way. In particular, we employ our newly developed RBF design algorithm for a detailed performance study and sensitivity analysis of the classification models for the popular Monk's problems. The results show that the accuracy of our classifiers is very impressive while our classification approach is systematic and easy to implement. In addition, differing complexity of the three Monk's problems is clearly reflected in the classification error surfaces for test data. By exploring these surfaces, we acquire better understanding of the data mining classification problems. Finally, we study the error surfaces to investigate trade-offs between different choices of model parameters to develop efficient and parsimonious models for a given application.

1 Introduction

The classification task is one of the major data-mining activities in many scientific and engineering applications. Gaussian kernels, in radial basis function (RBF) models and support vector machines, have shown very impressive performance in recent years. To develop such classifiers in a systematic way, Shin and Goel [1] recently proposed an innovative new algorithm for RBF model development, called the *Shin-Goel (SG) algorithm*. In their previous studies [1-3] they demonstrated its superior performance on many benchmark data sets and real world problems. In this paper we use this algorithm for a performance and sensitivity study of typical data mining classification problems. Specifically, we employ the three popular Monk's problems [4], which are of relatively small size but are representative of the modeling issues that arise in data mining applications.

The Monk's problems are binary classification problems of varying complexity based on robot datasets. Originally, these data sets were used to compare several learning techniques at the Second European Summer School on Machine Learning, as discussed in [4]. Since then, many authors [5-10] have studied their classification performance. In this paper we report the results of a detailed performance study of these problems using Gaussian kernel RBF models and the SG algorithm for

determining their design parameters. Our focus is on performance and sensitivity analysis for the selection of model parameters in developing classifiers, rather than on a comparative investigation with other classification methods and algorithms.

The paper is organized as follows. A description of the Monk's problems and related research is given in Section 2. In Section 3 we present the Gaussian RBF model and the Shin-Goel algorithm. Results of the performance study and sensitivity analysis are presented in Section 4. Some concluding remarks are presented in Section 5.

2 Monk's Problems

The Monk's problems [4] are derived from an artificial robot domain in which each training example is represented by six discrete-valued attributes. Each of the three problems involves learning a binary function defined over these attributes based on the given training data. The three problems are of differing complexity. The six attributes for each problem provide robot description and are listed in Table 1 along with the values each can take. Each problem is given by a logical description of a class, and robots may or may not belong to this class. The six attributes in Table 1 take $3+3+2+3+4+2 = 17$ values, and there are $3*3*2*3*4*2 = 432$ possible attribute combinations. The training data provided for classification is a subset of these 432 combinations. The learning task is to develop a classifier from the provided training data and then to evaluate its generalization performance on the entire data set. That is, in each case, all 432 examples are used as test set. A description of the three problems taken from [4] is given below.

1) [*Problem Monks 1*] (*head_shape = body_shape*) or (*jacket_color = red*). From the 432 possible examples, 124 were randomly selected for the training set. In this case, there were no misclassifications.

2) [*Problem Monks 2*] *Exactly two of the six attributes have their first value*. For example, if exactly two attributes (*body_shape* and *head_shape*) have their first values, i.e., *body_shape = head_shape = round*, this implies that the robot is not smiling, is holding no sword, the *jacket_color* is not red and it has no tie. From the 432 possible examples, 169 were randomly selected for training data. Again, there was no noise in this data set.

3) [*Problem Monks 3*] (*jacket_color is green and holding a sword*) or (*jacket_color is not blue and body_shape is not octagonal*). From the 432 examples, 122 were selected randomly for training, and among these 5% of them were misclassifications, i.e., there is noise in the training set.

Monks1 is in standard disjunctive normal form and is supposed to be easily learnable through the use of symbolic learning algorithms. Monks 2 is similar to parity problems. It combines different attributes in a way that makes it complicated to describe in DNF or CNF using the given attributes only. Monks 3 is again in DNF and serves to evaluate an algorithm in the presence of noise.

Table 1. Attributes and their values for THE Monk's problems

Attribute	Values
x ₁ : head_shape	round, square, octagon
x ₂ : body_shape	round, square, octagon
x ₃ : is_smiling	yes, no
x ₄ : holding	sword, balloon, flag
x ₅ : jacket_color	red, yellow, green, blue
x ₆ : has_tie	yes, no

3 Gaussian Radial Basis Function Model

3.1 Model Description

In general, a typical RBF model consists of m kernels in the hidden layer, input data \mathbf{x} and output \mathbf{y} . The input layer takes d -dimensional data vectors as input $\{\mathbf{x}_i \in R^d, i=1, \dots, n\}$, where n is the number of input vectors. Each input vector \mathbf{x} is transformed by the m kernels or basis functions in the hidden layer, producing outputs $\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x}), \dots, \Phi_m(\mathbf{x})$, where $\Phi_i(\mathbf{x})$ is the output from the i^{th} kernel function. These outputs are then weighted by w_i 's and summed to produce the model output $y = \sum w_i \Phi_i(\mathbf{x})$. For the kernel functions, there are several different choices that can be used in the RBF model. Among these, Gaussian is the most popular and is used in this study. The Gaussian kernels are obtained by taking $\Phi_i(\mathbf{x}) = \exp(-\|\mathbf{x} - \boldsymbol{\mu}_i\|^2 / 2\sigma_i^2)$ for $i=1, \dots, m$, where $\boldsymbol{\mu}_i$ and σ_i are the kernel parameters, called the center and width, respectively. Thus, in this case, the final model output \hat{y} is given below, where $\|\cdot\|$ represents the Euclidean norm.

$$\hat{y} = f(\mathbf{x}) = \sum_{j=1}^k w_j \exp(-\|\mathbf{x} - \boldsymbol{\mu}_j\|^2 / 2\sigma^2)$$

Consequently, the RBF model with Gaussian kernels is defined by the number of kernels (m), centers ($\boldsymbol{\mu}$), widths (σ) and weights (w). The first three parameters specify the hidden layer and are responsible for defining the non-linear transformation of the data. The weights w specify the linear mapping from the hidden to the output layer.

3.2 Shin-Goel Algorithm

When developing an RBF model for a problem, an important consideration is to find model parameters that represent a good compromise between under-fitting and over-fitting. Generally, too simple models tend to incur under-fitting and too complicated models tend to incur over-fitting, both of which eventually lead to poor classification performance on new data. The recently developed SG algorithm for RBF design [1, 2] seeks a good balance between these two extremes. It introduces a *Representational Capacity* (RC) criterion to choose kernel parameters in a principled way. One feature

of this algorithm is that it has been shown to almost always produce the best or near-best model in a systematic way. A summary of its implementation algorithm is given in the below; details can be found in [1, 2].

- Step 1:** For a given data matrix X of size $n \times d$, select a global width σ in the range of $0 < \sigma < \sqrt{d}/2$ and a RC criterion δ as $0 < \delta < 1$. Empirically, we chose $\delta=0.01, 0.005$, and 0.001 .
- Step 2:** Construct an interpolation matrix. $\mathbf{D} = [d_{ij}]_{i=1, \dots, n, j=1, \dots, n}$ such that
- $$d_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2).$$
- Step 3:** Determine the number of kernels (m) that satisfies the specified δ criterion. This step involves the computation of the rank of the interpolation matrix \mathbf{D} with a tolerance $\varepsilon = s_1 \times \delta$ where s_1 is the first singular value of \mathbf{D} .
- Step 4:** Determine kernel parameters $(\mu_1, \mu_2, \dots, \mu_m)$ in such a way that they maximize structural stability provided by the selected model complexity, m . This step involves the use of QR factorization with column pivoting of right singular vectors of \mathbf{D} .
- Step 5:** Compute weight $\mathbf{w} \equiv (w_1, w_2, \dots, w_m)$ by pseudo inverse and estimate output values.

4 Classification Results for Monk's Problems

4.1 Data Preparation

The data consists of six input attributes, as discussed above, and is available from the UCI machine learning data repository [11]. Each attribute is represented by a binary string of length k , where k is the number of distinct values an attribute can take. Thus, the attribute x_1 (head_shape) which can take three values (round, square and octagonal) is represented by a binary string of length three. The first value is one if head_shape is round, and the second and third values are zero. Similar interpretations hold for the other attributes. Using this conversion scheme, the input data are converted into 17 dimensional ($3 + 3 + 2 + 3 + 4 + 2$) vectors. Thus, the training data for Monks1, Monks2, and Monks3 are 124×17 , 169×17 , 122×17 matrices, respectively. The test data in each case is a 432×17 matrix. The outputs are one-dimensional vectors of binary variables, each of the same size as the input matrix.

4.2 Results for Monks1

The Shin-Goel algorithm chooses a model for each δ with the minimum classification error on the test set (432 examples). The errors on training and testing data sets and the corresponding models are summarized in Table 2 for the three δ values. Amongst these, the minimum training error is 0% while the testing error is 0.014 (=1.4%). To get a deeper understanding of the performance, we show the surface of training errors in Fig. 1(a). Here it is seen that the error goes to zero with increasing m as expected.

Also, larger σ requires fewer number of kernels m . The corresponding plot for testing error is shown in Fig. 1(b). Here the error continuously decreases with m , which is not a typical behavior. This seems to result from the fact that the training data for this problem are also included in the test data, unlike most practical cases. The sensitivity of the training and test errors also can be studied from their surfaces in Fig. 1 (a) and (b). To summarize, our best model for Monks 1 yields a testing error of 1.4% while keeping a training error of 0%.

Table 2. Classification Models for Monks1 Data

δ	σ	m	Training CE	Testing CE
0.01	1.4	79	0.0	0.025
0.005	1.4	107	0.0	0.023
0.001	1.6	122	0.0	0.014

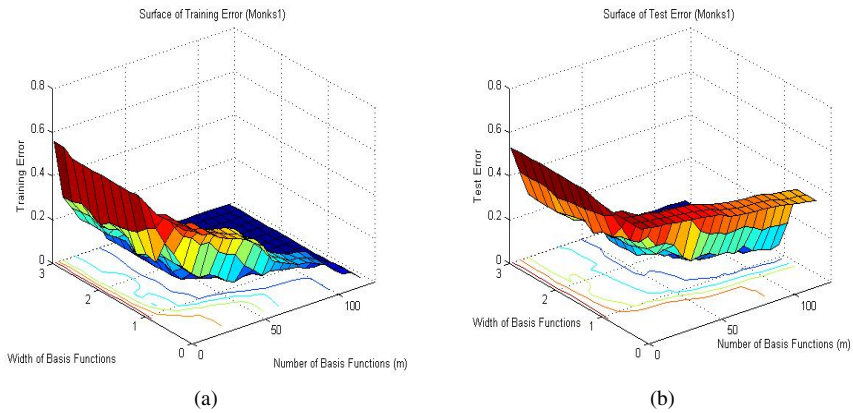


Fig. 1. Surfaces of (a) Training Error and (b) Test Error for Monks1

4.3 Results for Monks2

Regarding the Monks2 problem, almost all reported classification models [5-10] have shown a high error on testing data for this problem; only the back-propagation algorithm had 100% classification accuracy [4]. In this study, we followed the same procedure as used for Monks1. Our final model parameters and the error values are summarized in Table 3. The best performing model has a testing error of 18.5% and the training error is 1.2%. Plots of error surfaces for training and testing errors are shown in Figs. 2 (a) and (b), respectively. As expected, the training error goes to zero with increasing m . On the other hand, the testing error surface is very flat over a wide range of m and σ values, which indicates that increasing classifier complexity hardly affects testing errors in that range. However, as seen in Fig. 2(b), there is a noticeable decrease in test error for large m and small σ . This is also indicated in Table 3.

Table 3. Classification Models for Monks2 Data

δ	σ	m	Training CE	Testing CE
0.01	1.0	169	0.0	0.289
0.005	0.4	161	0.012	0.185
0.001	1.0	169	0.0	0.289

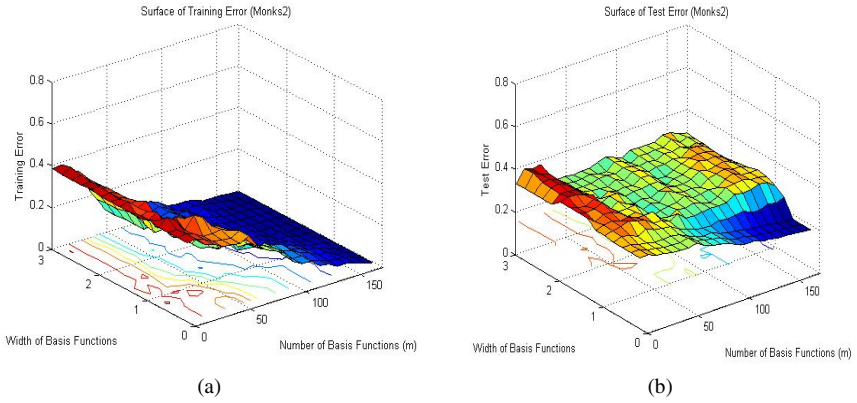


Fig. 2. Surfaces of (a) Training Error and (b) Test Error for Monks2

Table 4. Classification Models for Monks3 Data

δ	σ	m	Training CE	Testing CE
0.01	2.4	12	0.065	0.028
0.005	2.8	12	0.065	0.028
0.001	2.4	59	0.024	0.034

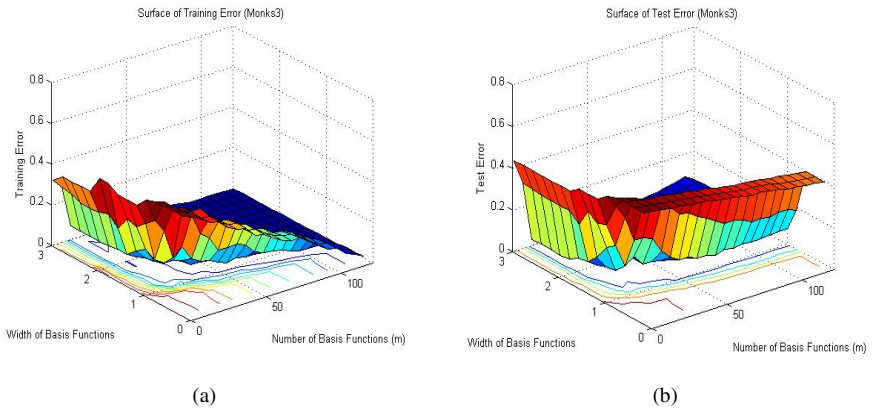


Fig. 3. Surfaces of (a) Training Error and (b) Test Error for Monks3

4.4 Results for Monks3

As noted earlier, the Monks3 data set contains noise, while the other two sets had no noise. The classification results for this problem are given in Table 4. The model with the lowest classification error on testing data is obtained for $\delta = 0.005$ with $\sigma = 2.8$ and $m = 12$. The corresponding testing and training errors are 2.8% and 6.5%, respectively. These error rates are higher than for Monks1 and seem to reflect the noise effect. The surfaces for the two errors are given in Figs. 3 (a) and (b). As expected, the training error in Fig. 3(a) continues to decrease with increasing m . However, the testing error for this case exhibits the usual behavior. It decreases first with m and then begins to increase. A minimum value occurs at $m = 12$ and $\sigma = 2.8$, but around this region of (m, σ) the error surface is somewhat flat.

5 Discussion and Concluding Remarks

Our best models for each Monk's dataset are shown in Table 5. As seen earlier, however, the error values are quite reasonable over a region of m and σ values. That is, there are other possible models that yield similar training and test error results. By considering a trade-off between model complexity and performance, appropriate other models can be chosen for the problem.

Table 5. Final models for monk's problems and their classification performance

Dataset	σ	m	Train CE	Accuracy%	Test CE	Accuracy%
Monks1	1.6	122	0.0	100	0.014	98.6
Monks2	0.4	161	0.012	98.8	0.185	81.5
Monks3	2.8	12	0.065	93.44	0.028	97.2

Though only the best models for the three cases are given in Table V, it is believed that the error surfaces provide very useful insights about the errors and their sensitivity for the three Monk's problems for selecting other models if so desired. In particular, we note that the test error surface for Monks1 continues to decrease with increasing model complexity. One explanation for this could be due to the inclusion of training data in the test set. In the usual classification studies this is not a common practice. For Monks2, the test error remains high in a wide range of classification models, which indicates the difficulty of learning this problem. Finally, for Monks3, the presence of noise seems to cause the test error to first decrease with model complexity and then increase. Because of this a relatively simple and parsimonious classification model yields very good results.

In this paper we have presented a detailed performance and sensitivity study of the Gaussian kernel RBF models for data mining applications using the Shin-Goel algorithm for the popular Monk's problems. The performance in the published literature concerning these problems varies over a wide range. Also, data selection and algorithmic details differ widely. In light of this, our emphasis here has not been on a comparative study but on getting a deeper insight into the nature of the problems by

investigating error surfaces and their sensitivity to model parameters. We expect that the results of this study would be of considerable interest to practitioners and researchers of data mining and its applications.

Acknowledgement

This research was supported by Kyungpook National University Research Fund, 2005.

References

1. A. L. Goel and Miyoung Shin, "Radial basis functions: an algebraic approach (with data mining applications)," *Tutorial notes; European conference on Machine Learning, Pisa, Italy*, September 2004.
2. M. Shin and Amrit L. Goel, "Empirical data modeling in software engineering using radial basis functions," *IEEE Trans. on Software Engineering*, vol.26, no.6, pp.567-576, June 2000.
3. M. Shin and Amrit L. Goel, "Modeling software component criticality using a machine learning approach," *Lecture Notes in Computer Science*, LNAI 3397, pp. 440-448, 2004.
4. S.B. Thrun et al, "The Monk's problems: a performance comparison of different learning algorithms," *Technical Report CMU-CS-91-197*, Carnegie Mellon University. 1991.
5. S. Saxon and A. Barry, "XCS and the Monk's problems in learning classifier systems: from foundations to applications," P. L. Lanzi et al, Ed., *Lecture Notes in Computer Science*, vol. 1813, pp. 440-448, 2000.
6. M. Casey and K. Ahmad, "In-situ learning in multi-net systems," *Lecture Notes in Computer Science*, vol. 3177, pp. 752-757, 2004.
7. H. Xiong, M.N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Trans. on Neural Networks*. March 2005.
8. M. W. Mitchell, "An architecture for situated learning agents," Ph.D. Dissertation, Monash University, Australia, 2003.
9. S. H. Huang, "Dimensionality reduction in automatic knowledge acquisition:a simple greedy search approach," *IEEE Trans. on Knowledge and Data Engineeingr.* vol.16, no. 6, pp.1364-1373, 2003.
10. K. Toh, Q-L Tran and O. Srinivasan, "Benchmarking a reduced multivariate polynomial pattern classifier," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no.2, pp. 460-474, 2005.
11. UCI machine learning data repository.
<http://www.ics.uci.edu/~mllearn/MLRepository.html>

TTLSC – Transductive Total Least Square Model for Classification and Its Application in Medicine

Qun Song, Tian Min Ma, and Nikola Kasabov

Knowledge Engineering & Discovery Research Institute
Auckland University of Technology
Private Bag 92006, Auckland 1020, New Zealand
{qsong, mmaa, nkasabov}@aut.ac.nz

Abstract. This paper introduces a novel classification method-transductive total least square classification method (TTLSC). While inductive approaches are concerned with the development of a model to approximate data in the whole problem space (induction), and consecutively – using this model to calculate the output value(s) for a new input vector (deduction), in transductive systems a local model is developed for every new input vector, based on some closest data to this vector from the training data set. The total least square method (TLS) is one of the optimal fitting methods that can be used for curve and surface fitting and outperform the commonly used least square fitting methods in resisting both normal noise and outlier. The TTLSC is illustrated by a case study: a real medical decision support problem of estimating the survival of haemodialysis patients. This personalized modelling can also be applied to solve other classification or clustering problems.

1 Introduction: Transductive Model and Total Least Square Method

Most of learning models and systems in artificial intelligence developed and implemented so far are based on inductive methods, where a model (a function) is derived from data representing the problem space and subsequently applied on new data. The derivation of the model in this manner therefore may not optimally account for all of the specific information related to a given new vector in the test data. An error is measured to estimate how well the new data fits into the model. The inductive learning and inference approach is useful when a global model (“the big picture”) of the problem is needed. In contrast, transductive methods estimate the value of a potential model (function) only in a single point of the space (the new data vector) utilizing additional information related to this point. This approach seems to be more appropriate for medical applications, where the focus is not on the model, but on the individual patient. Each individual data vector (e.g.: a patient in the medical area; a future time moment for predicting time series; or a target day for predicting a stock index) may need an individual, local model that fits the new data better than a global model, in which the new data is matched without taking any specific information about this data into account [3, 9, 15].

Transductive inference is concerned with the estimation of a function in single point of the space only. For every new input vector x_i that needs to be processed for a prognostic task, the N_i nearest neighbours, which form a sub-dataset D_i , are derived from an existing data set D and, if necessary, generated from an existing model M . A new model M_i is dynamically created from these samples to approximate the function in the point x_i - Fig. 1 and Fig. 2. Then the system is used to calculate the output value y_i for this input vector x_i (Fig. 1 and 2).

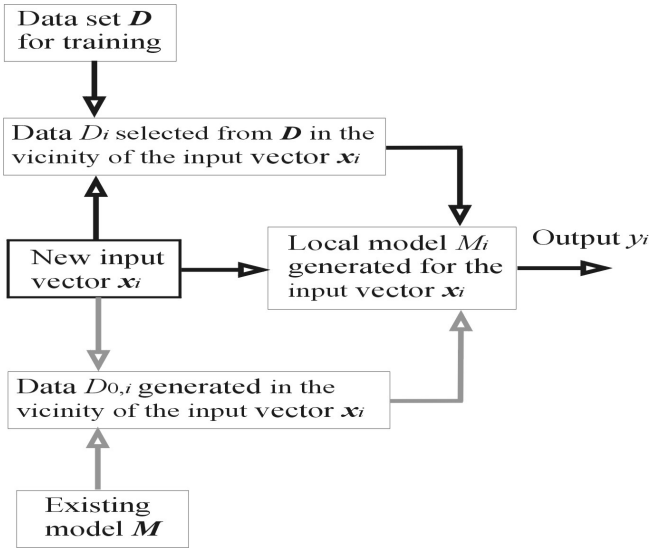
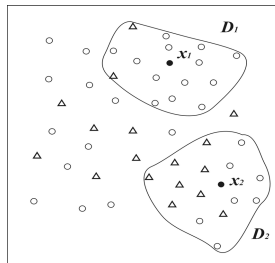


Fig. 1. A block diagram of a transductive reasoning system. An individual model M_i is trained for every new input vector x_i with data use of samples D_i selected from a data set D , and data samples $D_{0,i}$ generated from an existing model (formula) M (if such a model is existing). Data samples in both D_i and $D_{0,i}$ are similar to the new vector x_i according to defined similarity criteria.



● – a new data vector; ○ – a sample from D ; △ – a sample from M

Fig. 2. In the centre of a transductive reasoning system is the new data vector (here illustrated with two of them – x_1 and x_2), surrounded by a fixed number of nearest data samples selected from the training data D and generated from an existing model M

The problems of using a model of line (curve), plane (surface), or hyper-plane (hyper-surface) to fit a given data set are often encountered in many engineering applications. For solving such classical statistical problems, the conventional method is the Least Square (LS) fitting method. However, in many cases, LS is suboptimal. The optimal least square method is the so called Total Least Square (TLS) method [4, 14, 17]. Different from usually used LS methods the basic idea of the TLS method is to find a function, which can be a line (curve), plane (surface), or hyper-plane (hyper-surface), on the given data set and to minimize the sum of the distances between each data point to the estimated function.

Comparing with the usual LS method, to obtain the solution of TSL is generally quite burdensome. In the case of linear fitting, however, the problem of optimal fitting in the TLS sense is not so intricate. When the linear models are expressed as

$$b_0 + b_1 x_1 + b_2 x_2 + \dots + b_m x_m = 0. \quad (1)$$

where $x_j, j = 1, 2, \dots, m,$ are variables and b_0 is an arbitrary constant. For Eq. 1, the TLS fitting problem is to minimize the following total least square error E

$$E = \sum_{i=1}^n r_i^2, \quad r_i = \frac{|b_0 + b_1 x_{i1} + b_2 x_{i2} + \dots + b_m x_{im}|}{\sqrt{b_1^2 + b_2^2 + \dots + b_m^2}} = \left| b_0 + \sum_{j=1}^m b_j x_{ij} \right| / \left(\sum_{j=1}^m b_j^2 \right)^{1/2} \quad (2)$$

where n is the number of vectors in the data set.

Either a linear neural network using a constrained Hebbian learning rule [17] or a steepest descent algorithm can be used to solve such a problem. In our current research, we use the latter.

We apply the transductive technology to the TLS method for solving the classification problem better: for each class, one TLS function is created on the local area that is based on the position of the new data in the training data space, and the new data belongs to such a class – the related TLS function has the shortest distance to the new data point.

The paper is organized as follows: Section 2 presents the structure and algorithm of the TTLSC method for classification. Section 3 illustrates the approach on a case study example. Conclusions are drawn in Section 4.

2 Transductive Total Least Square Method for Classification: Structure and Learning Algorithm

TTLSC is a TLS method using the transductive technology for solving classification problems. The distance between vectors \mathbf{x} and \mathbf{y} is measured in TTLSC in normalized Euclidean distance defined as follows (the values are between 0 and 1):

$$\|\mathbf{x} - \mathbf{y}\| = \frac{1}{P} \left[\sum_{j=1}^P |x_j - y_j|^2 \right]^{1/2} \quad (3)$$

where: $\mathbf{x}, \mathbf{y} \in \mathbf{R}^P$

Consider the classification problem has two classes and m variables, for each new data vector \mathbf{x}_q , the TTLSC learning algorithm performs the following steps:

- 1) Normalize the training data set and the new data (the values are between 0 and 1).
- 2) Search in the training data set in the whole space to find D_q that includes N_q training samples closest to x_q . The value of N_q can be pre-defined based on experience, or - optimized through the application of an optimization procedure. Here we assume the former approach.
- 3) If all training samples in D_q belong to the same class, the new data belongs to this class and the procedure ends. Otherwise,
- 4) Calculate the distances d_b , $i = 1, 2, \dots, N_q$, between x_q and each of data samples in D_q and calculate the vector weights $w_i = 1 - (d_i - \min(\mathbf{d}))$, here, $i = 1, 2, \dots, N_q$, $\min(\mathbf{d})$ is the minimum value in the distance vector \mathbf{d} , $\mathbf{d} = [d_1, d_2, \dots, d_{N_q}]$.
- 5) Use the Weighted Least Square method [6,7] to create a function as Eq.4 with the data pairs $[x_b, y_i]$ and w_b , $i = 1, 2, \dots, N_q$,

$$y = a_0 + a_1 x_1 + a_2 x_2 + \dots + a_m x_m \tag{4}$$

where, $y_i = 0$ if training data sample x_i belongs to class 1 and, $y_i = 1$ if x_i belongs to class 2.

- 6) Create two initial TLS functions for two classes respectively

$$f_1(\mathbf{x}, \mathbf{b}^{(0)}) = B_0^{(0)} + B_1^{(0)} x_1 + B_2^{(0)} x_2 + \dots + B_m^{(0)} x_m = 0; \quad B_j^{(0)} = a_j, \quad j = 0, 1, 2, \dots, m. \tag{5a}$$

$$f_2(\mathbf{x}, \mathbf{b}^{(0)}) = b_0^{(0)} + b_1^{(0)} x_1 + b_2^{(0)} x_2 + \dots + b_m^{(0)} x_m = 0; \\ b_0^{(0)} = a_0 - 1 \text{ and } b_j^{(0)} = a_j, \quad j = 1, 2, \dots, m. \tag{5b}$$

- 7) Apply the steepest descent method to optimize the parameters B and b for two TLS functions following Eq. 6 – 8.
- 8) Calculate the distances r_1 and r_2 , from the new data point to f_1 and f_2 respectively, the new data belongs to class1 if $r_1 < r_2$ and otherwise, it belongs to class 2.
- 9) End of the procedure.

The parameter optimization procedure is described as following:

Suppose there are N_{q1} class 1 training data D_{q1} and N_{q2} class 2 training data D_{q2} in the D_q . The f_1 and f_2 are optimized on D_{q1} and D_{q2} respectively. The optimization for f_1 is showed following (it is the same manner for f_2):

The optimization minimizes the following objective function (the re-written Eq.2) :

$$E = \sum_{i=1}^{N_{q1}} r_i^2, \quad r_i = \frac{|B_0 + B_1 x_{i1} + B_2 x_{i2} + \dots + B_m x_{im}|}{\sqrt{B_1^2 + B_2^2 + \dots + B_m^2}} = \left| B_0 + \sum_{j=1}^m B_j x_{ij} \right| / \left(\sum_{j=1}^m B_j^2 \right)^{1/2} \tag{6}$$

Then the steepest descent algorithm is used to obtain the formulas for the optimization of the parameters \mathbf{B} , so that the value of E from Eq. (6) is minimized:

$$B_0(k+1) = B_0(k) - \eta \sum_{i=1}^{N_{q1}} \left\{ \left(B_0(k) + \sum_{j=1}^m B_j(k) x_{ij} \right) / \left(\sum_{j=1}^m B_j^2(k) \right)^{1/2} \right\} \tag{7}$$

$$B_j(k+1) = B_j(k) - \eta \sum_{i=1}^{N_{q1}} \left\{ x_{ij} \left[\left(B_0(k) + \sum_{j=1}^m B_j(k) x_{ij} \right) / \left(\sum_{j=1}^m B_j^2(k) \right)^{1/2} \right] - B_j(k) r_i^2(k) \right\} \quad (8)$$

where, η is the learning rate.

In the TTLSC algorithm, the following indexes are used:

- data samples: $i = 1, 2, \dots, N_{q1} \text{ or } N_{q2};$
- variables: $j = 1, 2, \dots, m;$
- optimization iterations: $k = 1, 2, \dots$

3 Case Study Example of Applying the TTLSC for a Medical Decision Support Problem

A medical dataset is used here for experimental analysis. Data originate from the Dialysis Outcomes and Practice Patterns Study (DOPPS, www.dopps.org) [5]. The DOPPS is based upon the prospective collection of observational longitudinal data from a stratified random sample of haemodialysis patients from the United States, 8 European countries (United Kingdom, France, Germany, Italy, Spain, Belgium, Netherlands, and Sweden), Japan, Australia and New Zealand. There have been two phases of data collection since 1996, and a third phase is currently just beginning. To date, 27,880 incident and prevalent patients (approximately 33% and 66% respectively) have been enrolled in the study, which represents approximately 75% of the world’s haemodialysis patients. In this study, prevalent patients are defined as those patients who had received maintenance hemodialysis prior to the study period, while incident patients are those who had not previously received maintenance hemodialysis.

The research plan of the DOPPS is to assess the relationship between haemodialysis treatment practices and patient outcomes. Detailed practice pattern data, demographics, cause of end-stage renal disease, medical and psychosocial history, and laboratory data are collected at enrollment and at regular intervals during the study period. Patient outcomes studied include mortality, frequency of hospitalisation, vascular access, and quality of life. The DOPPS aims to measure how a given practice changes patient outcomes, and also determine whether there is any relationship amongst these outcomes, for the eventual purpose of improving treatments and survival of patients on haemodialysis.

The dataset for this case study contains 6100 samples from the DOPPS phase 1 in the United States, collected from 1996-1999. Each record includes 24 patient and treatment related variables (input): demographics (age, sex, race), psychosocial characteristics (mobility, summary physical and mental component scores (sMCS, sPCS) using the Kidney Disease Quality of Life (KD-QOL®) Instrument), co-morbid medical conditions (diabetes, angina, myocardial infarction, congestive heart failure, left ventricular hypertrophy, peripheral vascular disease, cerebrovascular disease, hypertension, body mass index), laboratory results (serum creatinine, calcium, phosphate, albumin, hemoglobin), haemodialysis treatment parameters (Kt/V, haemodialysis angioaccess type, haemodialyser flux), and vintage (years on haemodialysis at the commencement of the DOPPS). The output is survival at

2.5 years from study enrollment (yes or no). All experimental results reported here are based on 10-cross validation experiments [10].

For comparison, several well-known methods of classification are applied to the same problem, such as Support Vector Machine (SVM) and transductive SVM [16], Evolving Classification Function (ECF) [8], Multi-Layer Perceptron (MLP) [13], Radial Basis Function (RBF) [13], and Multiple Linear Regression along with the proposed TTLSC, and results are given in Table 1.

The Kappa statistic, K, formally tests for agreement between two methods, raters, or observers, when the observations are measured on a categorical scale. Both methods must rate, or classify, the same cases using the same categorical scale [1]. The degree of agreement is indicated by K, which can be roughly interpreted as follows: $K < 0.20$, agreement quality poor; $0.20 < K < 0.40$, agreement quality fair; $0.40 < K < 0.60$, agreement quality moderate; $0.60 < K < 0.80$, agreement quality good; $K > 0.80$, agreement quality very good. Confidence intervals for K were constructed using the goodness-of-fit approach of Donner & Eliasziw [2]. There is no universally agreed method for comparing K between multiple tests of agreement. In this study, K for different classification methods was compared using the permutation or Monte Carlo resampling routine of McKenzie [11,12].

Agreement refers to the quality of the information provided by the classification device and should be distinguished from the usefulness, or actual practical value, of the information. Agreement provides a pure index of accuracy by demonstrating the limits of a test's ability to discriminate between alternative states of health over the complete spectrum of operating conditions. To date, prognostic systems for the prediction of haemodialysis patient survival have published accuracy of 60-70%. The experimental results in Table 1 illustrate that the TTLSC in this paper provide incrementally better results, towards a K of > 0.60 and a level of accuracy ~80%, which are generally regarded as thresholds for clinical utility.

Table 1. Experimental Results on the DOPPS Data

Model	Kappa (95% Confidence Intervals)*		P-value	Sensitivity		
	Agreement (%)	Specificity (%)		Sensitivity (%)		
RBF	0.1675 (0.1268 - 0.2026)	<0.001	60.4	65.3	49.08	
ECF	0.1862 (0.1469 - 0.2224)	<0.001	61.5	63.4	51.76	
MLP	0.3833 (0.3472 - 0.4182)	<0.001	62.8	65.6	58.72	
Multiple Linear Regression		<0.001				
Linear Regression	0.4000 (0.3651 - 0.4357)		64.9	67.6	63.21	
SVM	0.4240 (0.3748 - 0.4449)	<0.001	65.3	68.2	62.3	
TSVM	0.4290 (0.3792 - 0.4460)	<0.001	57.2	61.2	52.9	
TTLSC	0.4503 (0.4152 - 0.4837)	Reference	70.5	73.6	68.4	

* Kappa values and confidence intervals ascertained with Stata Intercooled V 8.2 (StataCorp, College Station, TX), and P-values with KAPCOM [12]

4 Conclusions

This paper presents a transductive total least square method for classification – TTLSC. The TTLSC performs a better local generalization over new data as it develops individual models for each data vector that takes the location of new input vector in the space into account. This approach seems to be more appropriate for clinical and medical applications, where the focus is not on the model, but on the individual patient. At the same time, it is an adaptive model, in the sense that data can be added to the data set continuously and immediately, and made available for transductive TTLSC models. This type of modeling can be called “personalized”, and it is promising for medical decision support systems. The clinical plausibility of the approach and its results are satisfactory in this study. As the TTLSC creates a unique model for each data sample, it usually needs more performing time than inductive models. Further directions for the research include: (1) TTLSC system parameter optimization such as optimal number of nearest neighbors; and (2) applying the TTLSC method to other decision support systems, such as: cardio-vascular risk prognosis; biological processes modeling and classifications based on gene expression micro-array data.

Acknowledgement

The research presented in the paper is funded by the New Zealand Foundation for Research, Science and Technology under grant NERF/AUTX02-01, the New Zealand National Kidney Foundation under grant GSN-12, and the Auckland Medical Research Foundation. The Top Achiever Doctoral Scholarship from the Tertiary Education Commission (TEC) of New Zealand also funds the research.

References

1. Altman, D. G.: Practical Statistics for Medical Research, Chapman and Hall, London, Great Britain (1991)
2. Donner, A., M. Eliasziw: A goodness-of-fit approach to inference procedures for the kappa statistic: confidence interval construction, significance-testing and sample size estimation. *Statistics in Medicine* 11(1992) 1511-1519
3. Gammerman, A., Vovk, V., Vapnik, V.: Learning by transduction. In: Cooper, G. F. & Moral, S. (eds.): Proc. of the 14th Conference on Uncertainty in Artificial Intelligence. Madison, Wisconsin, Morgan Kaufmann, San Francisco, USA (1998) 148-155
4. Golub, C. L., Van Loan, C.: Matrix computations, Baltimore, MD: Jons Hopkins University Press.
5. Goodkin, D. A., Mapes, D.L., Held, P.J.: The dialysis outcomes and practice patterns study (DOPPS): how can we improve the care of hemodialysis patients? *Seminars in Dialysis*. 14 (2001) 157-159
6. Hsia, T. C.: System Identification: Least-Squares Methods. D.C. Heath and Company (1977)

7. Kasabov, N., Song, Q.: DENFIS: Dynamic, evolving neural-fuzzy inference systems and its application for time-series prediction. *IEEE Trans. on Fuzzy Systems*. 10 (2002) 144 – 154
8. Kasabov, N.: *Evolving connectionist systems: Methods and Applications in Bioinformatics. Brain study and intelligent machines.* Springer Verlag London Limited, Great Britain (2003)
9. Kukar, M.: Transductive reliability estimation for medical diagnosis. *Artif. Intell. Med.* 29 (2003) 81-106
10. Marshall, M. R., Song, Q., Ma, T.M., MacDonell, S., Kasabov, N.: Evolving Connectionist System versus Algebraic Formulae for Prediction of Renal Function from Serum Creatinine. *Kidney International*. 67 (2005) 1944-1954
11. McKenzie, D. P., Mackinnon, A. J., Peladeau, N., Onghena, P., Bruce, P. C., Clarke, D. M., Harrigan, S., McGorry, P. D.: Comparing correlated kappas by resampling: is one level of agreement significantly different from another? *Journal of Psychiatric Research*. 30 (1996) 483-492
12. McKenzie, D. P., Mackinnon, A. J., Clarke, D. M.: KAPCOM: a program for the comparison of kappa coefficients obtained from the same sample of observations. *Perceptual and Motor Skills*. 85(1997) 899-902
13. *Neural Network Toolbox User's Guide.* The Math Works Inc., 3 Apple Hill Drive, Natick, Massachusetts, Ver. 4 (2002)
14. Oja, E.: a simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 16 (1982) 267 – 273
15. Song, Q., Kasabov, N.: NFI: A Neuro-Fuzzy Inference Method for Transductive Reasoning. *IEEE Trans. on Fuzzy Systems*.13 (6) 2005, 799 – 808
16. Vapnik, V.: *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, NY (1995)
17. Xu, L., Oja E., Suen, C.Y.: Modified Hebbian Learning for Curve and Surface Fitting. *Neural Networks*. 5 (1992) 441 – 457

Forecasting Electricity Market Price Spikes Based on Bayesian Expert with Support Vector Machines

Wei Wu¹, Jianzhong Zhou^{1,*}, Li Mo¹, and Chengjun Zhu²

¹ College of Hydropower and Information Engineering,
Huazhong University of Science and Technology
430074 Hubei, Wuhan, P.R. China
weiwu208@hotmail.com, Prof.zhou.hust@263.net,
morlymorly@126.com

² China Three Gorges Project Corporation
443002 Hubei, Yichang, P.R. China
zhu_chengjun@ctgpc.com.cn

Abstract. This paper present a hybrid numeric method that integrates a Bayesian statistical method for electricity price spikes classification determination and a Bayesian expert (BE) is described for data mining with experience decision analysis approach. The combination of experience knowledge and support vector machine (SVM) modeling with a Bayesian classification, which can classify the spikes and normal electricity prices, are developed. Bayesian prior distribution and posterior distribution knowledge are used to evaluate the performance of parameters in the SVM models. Electricity prices of one regional electricity market (REM) in China are used to test the proposed method, experimental results are shown.

1 Introduction

The stochastic behavior of electricity prices, which play a central role in the power resources planning of regional electricity market (REM), is a challenge to hydropower producers operation and scheduling [1, 2].

Various techniques have been used in literature for electricity price forecasting [3]-[7]. ARIMA models and time series models are efficient ways to forecast electricity prices [3]. Artificial neural networks (ANN) have also been used to solve this problem [4], and combined with wavelet techniques, ARIMA models to improve the performance [5]. Recently, Data mining has been proposed as a novel technique in the electricity forecasting, Lu [6] has been successfully used it to forecast the electricity market price spikes. Zhao [7] introduced a data mining based approach to give a reliable forecast of the occurrence of price spikes

* Corresponding author. This paper is supported by the National Science Foundation of China (NSFC) (No.50579022, 50539140) and the research funds of University and college PhD discipline (No.20050487062).

and support vector machine (SVM) with probability classifier are chosen as programming. Support vector machine, a method of data mining, has been received an increasing attention in areas ranging from its original application in pattern recognition to the extended application of regression estimation [8]-[10]. Since the price spikes have significant vibration, those methods can't deal with the price spikes precisely. So, newly data information should be mining in the electricity price spikes forecasting with experience decisions. A classical approach of probability theory has been found to be very useful for experience decisions [11]. Bayesian expert system is a probabilistic representation for uncertain relationships and is useful for classification, forecasting, information retrieval, etc. [12].

In this paper, Bayesian expert with experience decision analysis approach is proposed. Bayesian classify approach for the electricity spikes, normal price and lower price is presented in Section 2. In Section 3, Bayesian expert with SVM methodology of price spike forecasting is described in details. Case studies is given in Section 4. The paper then concludes with Section 5.

2 Bayesian Classification Approach

The initial electricity prices data set Ψ is classified into three non-overlapping sets Ψ_1, Ψ_2, Ψ_3 via the following classified method. Suppose $x \in \Psi \subset \mathcal{R}$, $y \in \{1, 0, -1\}$, $y = f(x)$ (where x denotes a sample of electricity price and y is classified value of each sets), the sample data (x, y) has the following rules.

(1) The likelihood of occurrence of prices pikes, normal prices and lower prices are determined by the classification of each sets Ψ_1, Ψ_2, Ψ_3 . x_{max} is the maximum value of the sample, $0 \leq x_{lower} \leq a_1, x_{lower} \in \Psi_3, a_1 \leq x_{normal} \leq a_2, x_{normal} \in \Psi_2$, and $a_2 \leq x_{pike} \leq x_{max}, x_{spike} \in \Psi_1$ (where $x_{lower}, x_{normal}, x_{spike} \in \Psi$; $0 \leq a_1, a_2 \leq x_{max}$ are classified limit values of each sets).

(2) Suppose we have known the prior distribution of the sample electricity prices, $x \sim P(\cdot)$ (where $P(\cdot)$ is a distribution function).

(3) From (1) and (2), we can obtain the combined distribution function $P(x, y)$, $x \times y \in \mathcal{R} \times \mathcal{R}$. Because variable y has three features -1, 0, 1, the distribution function $P(x, y)$ will dividend into 3 function $P(x, -1), P(x, 0), P(x, 1)$.

In this paper we use 0-1 lost function to be our lost function $c(x, y, f(x))$ [8], ε_0 is a threshold.

$$c(x, y, f(x)) = \hat{c}(y - f(x)) = \begin{cases} 0 & \text{if } |y - f(x)| < \varepsilon_0 \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

The classify problem is minimization the expect risk function $R(f)$ (see Eq.(2)) to obtain classification variables a_1 and a_2 , so function $f(x)$ can be determined.

$$\begin{aligned} \min R(f) &= \int_{x \times y} c(x, y, f(x)) dP(x, y) \\ &= \int_0^{a_1} c(x, -1, f(x)) p(x, -1) dx + \int_{a_1}^{a_2} c(x, 0, f(x)) p(x, 0) dx \\ &+ \int_{a_2}^{x_{max}} c(x, 1, f(x)) p(x, 1) dx \end{aligned} \quad (2)$$

3 Bayesian Expert with Support Vector Machine

3.1 Bayesian Expert

After obtaining electricity price spikes of the Bayesian classification approach, the newly electricity price spikes sample data $D = \{(X_1, y_1), (X_2, y_2), \dots, (X_l, y_l)\}$ (where $X_i \in \mathcal{X} \subseteq \mathcal{R}^n$ is the input vector, $y_i \in \Psi_1 \subset \mathcal{R}$ is the actual electricity price spikes data and l are total numbers of learning samples). Prior distribution is updated to the posterior distribution by using Bayes's rule [11]:

$$P(\theta | D) = \frac{P(D | \theta)P(\theta)}{P(D)} \propto L(\theta | D)P(\theta) \tag{3}$$

where the likelihood function $L(\theta | D)$ gives the probability of the observed data as a function of the unknown model parameters.

Next, we should set a hierarchical model specification with its likelihood function, the forecasting of electricity spikes by using Bayesian experience approach can be outline as Fig.1. In section 2, we have the knowledge of electricity price spikes sample that has a fit distribution function $P(\cdot)$, if $P(\cdot)$ is the Gausses distribution (normally in the nature), the probability density function(pdf) for the output target is written as following:

$$p(y | X, \theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y - f(X, \theta))^2}{2\sigma^2}\right) \tag{4}$$

where σ^2 is the variance of distribution function.

3.2 Bayesian Expert with Support Vector Machine

SVM approximates the function of the following form [8, 9]:

$$f(X) = \omega \cdot \phi(X) + b \tag{5}$$

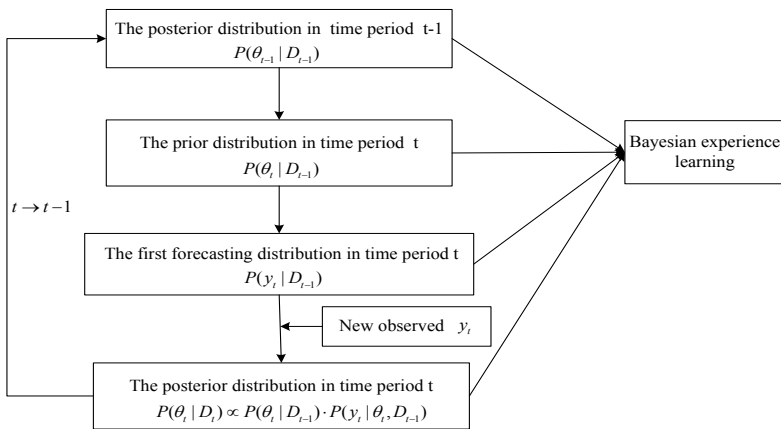


Fig. 1. The behavior of Bayesian expert

The coefficients ω and b are estimated by minimizing the regularized risk function of following:

$$\min \frac{1}{2} \|\omega\|^2 + C \frac{1}{l} \sum_{i=1}^l L_\varepsilon(y_i, f(x)) \tag{6}$$

where

$$L_\varepsilon(y_i, f(X_i)) = \begin{cases} |y_i - f(X_i)| - \varepsilon & \text{if } |y_i - f(X_i)| \leq \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$\|\omega\|^2$ is called the regularized term. C and ε are referred to parameters. The loss equals zero if forecasted value is within the ε -tube (Eq.(7)). In order to get the estimations of ω and b , Eq.(6) is transformed to the primal objective function by introducing the positive slack variables ξ_i, ξ_i^* [8, 9]

$$f(X) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(X_i, X) + b \tag{8}$$

In Eq.(8), α_i, α_i^* are Lagrange multipliers belong to slake variables ξ_i, ξ_i^* . The value of the kernel is equal to the inner product of two vectors X_i, X_j in the feature space $\phi(X_i)$ and $\phi(X_j)$; That is $K(X_i, X_j) = \phi(X_i) \cdot \phi(X_j)$. In this work, the Gaussian function $\exp(-\frac{(X_i - X_j)^2}{\sigma_0^2})$ is used in the SVMs.

The selection of there parameters, σ_0, ε, C of a SVM model are very important to the accuracy of forecasting. Suppose, we have known the fit distribution of electricity price spike data $(y_i | X, \theta) \sim N(\xi_i, \sigma_0^2)$ and $\theta = \{\sigma_0, \varepsilon, C\}$, $\xi_i \sim N(\mu, \tau^2)$. Let $\theta = (C, \varepsilon, \mu, \log \sigma_0, \log \tau)$ and define $\xi = \{\sigma_0, \varepsilon, C\}$. Then, the prior distribution is:

$$\begin{aligned} &P(\xi_1, \dots, \xi_l, \xi, \log \sigma_0, \log \tau | Y) \\ &\propto P(\xi_l, \xi, \log \sigma_0, \log \tau) \cdot \prod_{i=1}^l P(\xi_i | \xi, \tau) \prod_{i=1}^l P(y_i | \xi_i, \sigma_0) \end{aligned} \tag{9}$$

The log function of the Eq.(9) is following,

$$\begin{aligned} &\log P(\xi, \log \sigma_0, \log \tau | Y, \xi_1, \dots, \xi_l) \propto \log P(\xi_1, \dots, \xi_l, \xi, \log \sigma_0, \log \tau | Y) \\ &\propto -\log \sigma_0 - (l - 1) \log \tau - \frac{1}{2\tau^2} \sum_{i=1}^l (\xi_i - \xi)^2 - \frac{1}{2\sigma_0^2} \sum_{i=1}^l (\xi_i - y_i)^2 \end{aligned} \tag{10}$$

In that it's difficulty to maximization $\log P(\xi, \log \sigma_0, \log \tau | Y)$ and gain the parameters ξ, σ_0, τ . In this paper, we use EM statistical algorithm [11] to complete the maximization of $\log P(\xi, \log \sigma_0, \log \tau | Y)$.

In the E-step, suppose there are estimations of parameters in time period t , $\theta^{(t)} = (\xi^{(t)}, \sigma_0^{(t)}, \tau^{(t)})$ because the prior distribution of ξ_i is a normal distribution, we can get the distribution function in the the conditions of Y and $\theta^{(t)}$ in Eq.(11).

$$(\xi_i | \theta^{(t)}, Y) \sim N(\xi_i^{(t)}, V_i^{(t)}) \tag{11}$$

where

$$\begin{aligned} \hat{\xi}_i^t &= \left(\frac{\xi}{(\tau^{(t)})^2} + \frac{1}{(\sigma_0^{(t)})^2 \hat{y}_i} \right) / \left(\frac{1}{(\tau^{(t)})^2} + \frac{1}{(\sigma_0^{(t)})^2} \right) \\ V_i^{(t)} &= \left(\frac{1}{(\tau^{(t)})^2} + \frac{1}{(\sigma_0^{(t)})^2} \right)^{-1} \end{aligned} \tag{12}$$

then, for any irrelevant vector ς with ξ_i ,

$$E \left[(\xi_i - \varsigma)^2 \mid \theta^{(t)}, Y \right] = \left[E(\xi_i \mid \theta^{(t)}, Y) - \varsigma \right]^2 + var(\xi_i \mid \theta^{(t)}, Y) = (\hat{\xi}_i^{(t)} - \varsigma)^2 + V_i^{(t)} \tag{13}$$

let $\varsigma = \xi$ and $\varsigma = y_i$ respectively, the expectation function of Eq.(9) is

$$\begin{aligned} Q(\theta \mid \theta^{(t)}, Y) &= -\log \sigma_0 - (l - 1) \log \tau - \frac{1}{2\tau^2} \sum_{i=1}^l \left[(\hat{\xi}_i^{(t)} - \xi)^2 + V_i^{(t)} \right] \\ &\quad - \frac{1}{2\sigma_0^2} \sum_{i=1}^l \left[(\hat{\xi}_i^{(t)} - y_i)^2 + V_i^{(t)} \right] + c_1 \end{aligned} \tag{14}$$

here, c_1 has no relationship with parameter θ .

In the M-step, let $\frac{\partial Q}{\partial \xi} = 0, \frac{\partial Q}{\partial \sigma_0} = 0, \frac{\partial Q}{\partial \tau} = 0$, we can obtain the estimation of parameters in the period time $t + 1$,

$$\begin{cases} \xi^{(t+1)} = \frac{1}{l} \sum_{i=1}^l \hat{\xi}_i^{(t)} \\ \sigma_0^{(t+1)} = \left\{ \sum_{i=1}^l \left(y_i - \hat{\xi}_i^{(t)} \right)^2 + V_i^{(t)} \right\}^{\frac{1}{2}} \\ \tau^{(t+1)} = \left\{ \sum_{i=1}^l \left(\hat{\xi}_i^{(t)} - \xi^{(t+1)} \right)^2 + V_i^{(t)} \right\}^{\frac{1}{2}} \end{cases} \tag{15}$$

In that $\xi^{(t+1)} = \{C^{(t+1)}, \varepsilon^{(t+1)}, \mu^{(t+1)}\}$ from Eq.(15), we can get the estimation of parameters $C^{(t+1)}, \varepsilon^{(t+1)}, \sigma_0^{(t+1)}$ in the forecasting time period $t + 1$, then employe those parameters in the Eq.(8). Eq.(8) yields the forecast electricity price spike value, the electricity normal prices and lower prices can also be forecasted by using this proposed BE-SVM method.

4 Electricity Price Spikes Forecasting

This study employed electricity prices of one REM in China, electricity prices from July 15 to August 21, 2005 serve as experimental data, every day has 24(hours) point data (see Fig.2). Electricity price data are divided into three data sets: the training data set, the validation data set and the testing data set. In order to test the ability of proposed method, three models, BE-SVM, SVM and

Table 1. Training and testing data sets of the proposed model

	BE-SVM model	SVM model	ANN model
Training data	July 15-August 14	July 15-August 14	
Validation data	August 15-August 20	August 15-August 20	July15-August 20
Testing data	August 21	August 21	August 21

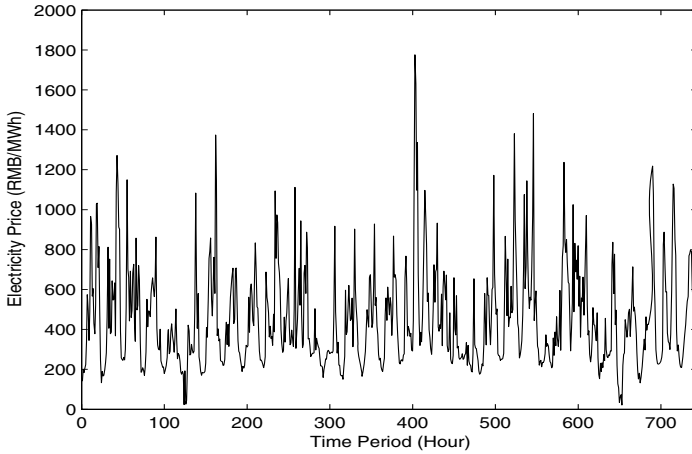


Fig. 2. The actual electricity price data from July 15 to August 15, 2005

ANN are used to compare the forecasting performance; three data sets are listed in Table 1. By using Bayesian classification approach, we can obtain electricity price spikes, normal prices and lower prices sets Ψ_1, Ψ_2, Ψ_3 and $a_1=240.91, a_2=643.43$. The newly classified electricity price spikes $PS \sim N(895.4, 209.12^2), PS \in \Psi_1$, normal price $PN \sim N(405.50, 126.32^2), PN \in \Psi_2$ and lower price $PL \sim LogN(5.17, 0.452^2), PL \in \Psi_3, N(\cdot)$ is the Gaussses distribution.

In the training stage of BE-SVM, the training data of each classified set are fed into the BE-SVM model (Eq.(8)), and the structural risk minimization principle is employed to minimize the training error. While training errors improvement occurs, are employed to calculate the validation error. Then, the adjusted parameters with minimum validation error are selected as the most appropriate parameters $\xi = \{C, \varepsilon, \mu\}$ and σ_0 . The parameters for the different price spike, normal price, and lower price BE-SVM models and SVM models are estimated to illustrate in Table 2. The mean absolute percentage error (MAPE) value of forecasted electricity price spikes in the BE-SVM model is 7.83%, less than the SVM model of 11.69%. Bayesian classification method proven to have a promising performance in electricity spike forecasting, the forecasting electricity spike occurrence time period are 10:00–12:00, 17:00–19:00, 19:00–21:00, August 21,2005, and has the same performance of the actual electricity price spike appearance, the absolute percentage errors (APE) of different forecasting method are shown in Table 3.

Table 2. Forecasting results and parameters of SVM model and BE-SVM model

	BE-SVM model			MAPE of testing(%)
	σ_0	C	ε	
Spike price	204.48	2796.87	100	7.83
Normal price	120.20	2187.93	50	5.55
Lower price	0.43	1206.13	30	4.63
	SVM model			MAPE of testing(%)
	σ_0	C	ε	
Spike price	192.37	2585.95	200	11.69
Normal price	114.20	2089.27	100	5.87
Lower price	0.40	1206.13	50	4.92

Table 3. Electricity prices classification (From 00:00 to 24:00 clock, August 21, 2005) of one REM and forecasting results of BE-SVM, SVM and ANN models in the spike price set, normal price set and lower prices set, the APE also given in the table of each forecasting model

Data time	Classified sets	Actual	forecasting prices (RMB/MWh)			APE(%)		
			BE-SVM	SVM	ANN	BE-SVM	SVM	ANN
10:00–11:00	Spikes	966.65	950.27	900.50	770.06	1.69	6.84	20.34
11:00–12:00		926.71	920.01	860.27	739.32	0.72	7.17	20.22
17:00–18:00		1028.04	884.62	857.83	792.06	13.95	16.56	22.95
18:00–19:00		1032.92	920.16	865.62	783.17	10.92	16.20	24.18
19:00–20:00		709.77	800.26	800.38	766.24	12.75	12.77	7.96
20:00–21:00		815.22	758.80	729.00	627.00	6.92	10.58	23.09
04:00–05:00	Normal	259.77	247.88	250.79	243.30	4.58	3.46	6.34
05:00–06:00		336.53	284.25	280.70	324.13	15.54	16.59	3.68
06:00–07:00		574.22	524.64	570.38	566.22	8.63	0.67	1.39
07:00–08:00		493.82	478.46	488.85	480.01	3.11	1.01	2.80
08:00–09:00		344.66	349.76	320.99	322.69	1.48	6.87	6.37
09:00–10:00		489.57	473.07	441.23	401.84	3.37	9.87	17.92
12:00–13:00		566.75	525.73	534.24	525.87	7.24	5.74	7.21
13:00–14:00		604.44	556.11	536.72	520.65	8.00	11.20	13.86
14:00–15:00		430.47	402.12	393.73	391.76	6.59	8.53	8.99
15:00–16:00		377.90	373.00	398.50	349.98	1.30	5.45	7.39
16:00–17:00		520.86	587.35	550.69	482.84	12.77	5.73	7.30
21:00–22:00		491.26	474.23	473.59	468.34	3.47	3.60	4.67
22:00–23:00		394.92	389.75	385.12	383.92	1.31	2.48	2.79
23:00–24:00	291.74	290.88	288.74	290.19	0.29	1.03	0.53	
00:00–01:00	Lower	225.64	221.66	220.55	217.10	1.76	2.26	3.78
01:00–02:00		227.97	214.48	214.37	203.77	5.92	5.97	10.62
02:00–03:00		232.58	214.04	211.58	212.57	7.97	9.03	8.60
03:00–04:00		239.51	232.63	233.66	229.52	2.87	2.44	4.17

5 Conclusion

In this paper, a data mining approach is presented for predicting the occurrence of the electricity market price spikes together with the ability of predicting normal range prices. We are among the first to present a method which can be used to successfully predict the time point of occurrence of price spikes. The case studies also show that in many existing classification algorithms, BE can give a reliable spike occurrence prediction. Moreover, the result of probability classifier can be combined with BE-SVM to increase the prediction accuracy and provide more information. In our case study we also successfully combine the spike forecast with normal price forecasting to give a complete forecast of market price. The contribution of this paper enables the hydropower producers ability to analyze the price spikes and take advantage of them.

References

1. Wu, W., Zhou, J.Z., Zhu, C.J., Yang, J.J.. A No-arbitrage Equilibrium Model for the Regional Electricity Market of China. Proceeding of 2005 IEEE International Conference on Industrial Technology,(2005) 682-687
2. Benini,M.,Marracci, M.,Pelacchi, P.. Day-ahead Market Price Volatility Analysis in Deregulated electricity markets. Proceedings of the IEEE Power Engineering Society Summer Meeting, (2002) 1354-1359
3. Contreras, J., Espinola, R., Nogales, F.J., Conejo, A.J.. ARIMA Models to Predict Next-day Electricity Prices. IEEE Transactions on Power Systems ,Vol.18, (2003)1014-1020
4. Wang, A.J., Ramsay, B.. A Neural Network Based Estimator for Electricity Real-time- Pricing with Particular Reference to weekend and Public Holidays. Neuro-computing, 23 (1998) 47-57
5. Conejo A.J., Plazas M.A., Espinola R., Molina A.B.. Day-ahead Electricity Price Forecasting Using the Wavelet Transform and ARIMA models. IEEE Transactions on Power Systems, Vol.20 (2005) 1035-1042
6. Lu, X. , Dong, Z.Y. , Li, X.. Electricity Market Price Spike Forecast with Data Mining Techniques. Electric Power Systems Research, 73(2005) 19-29
7. Zhao, J.H., Dong, Z.Y., Li, X., Wong K.P.. General Method for Electricity Market Price Spike Analysis, IEEE Power Engineering Society General Meeting, Vol.1, (2005) 1286-293
8. Deng, N.Y., Tian, Y.J.. Support Vector mearch: a New Approach in Data Mining, Beijing Science Press (2004)
9. Van Gestel, T., Suykens, J.A.K., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B., De Moor, B., Vandewalle, J.. Financial Time Series Prediction Using Least Squares Support Vector Machines within the Evidence Framework. IEEE Transactions on Neural Networks, Vol.12, (2001) 809-821
10. Cao, L.J.. Support Vector Machines Experts for Time Series Forecasting. Neuro-computing, 51(2003) 321-339
11. Mao, S.S., Wang, J.L., Pu, X.L.. Advanced Mathematical Statistics. China higher education press, Beijing and springer-Verlag, Berlin Heidelberg (1998)
12. Ni, E., Luh, P.B.. Forecasting Power Market Clearing Price and Its Discrete PDF Using a Bayesian-based Classification Method. Proceedings of the IEEE PES Winter Meeting, (2001) 1518-1523

Integrating Local One-Class Classifiers for Image Retrieval

Yiqing Tu, Gang Li, and Honghua Dai

School of Engineering and Information Technology,
Deakin University, Vic 3125, Australia
{ytu, gang.li, hdai}@deakin.edu.au

Abstract. In content-based image retrieval, learning from users' feedback can be considered as an one-class classification problem. However, the OCIB method proposed in [1] suffers from the problem that it is only a one-mode method which cannot deal with multiple interest regions. In addition, it requires a pre-specified radius which is usually unavailable in real world applications. This paper overcomes these two problems by introducing ensemble learning into the OCIB method: by Bagging, we can construct a group of one-class classifiers which emphasize various parts of the data set; this is followed by a rank aggregating with which results from different parameter settings are incorporated into a single final ranking list. The experimental results show that the proposed I-OCIB method outperforms the OCIB for image retrieval applications.

1 Introduction

With the rapid increase of digital image collections, content-based image retrieval has attracted much research interest in recent years. The retrieval engine of an image retrieval system can be regarded as a machine learning process: by learning from users' feedback, the performance of image retrieval systems can be further improved [2,3,4]. While most work regards learning from user's relevance feedback as a strict two-class classification problem, it is more natural to consider the learning problem as an one-class classification problem [5].

In one-class classification, it is assumed that only information of the target class is available. There is no information about objects of the other class(es). Typically in image retrieval system, most users of image retrieval system are only interested in their target images, and reluctant to take effort to identify irrelevant images. This leads to the fact that only a relatively small number of relevant samples are available to train the retrieval system.

In addition, it is reasonable to assume that relevant images are well defined and cluster in certain way, whereas irrelevant images can be any kind, thus to sample well the space of irrelevant class is rather difficult. Because of this, one-class classification is especially suitable for image retrieval because it constructs a tight hyper-sphere in the input space to cover most positive training data.

Currently one-class classification has been applied to typically two kind of problems: *outlier detection* and *information retrieval*. With outlier detection,

most of the data points are identified as relevant while only a small portion are considered as outliers and irrelevant. On the contrary, in information retrieval system, user looks for a small but coherent subset of data points. Therefore, the learning problem in retrieval system calls for a totally different approach.

In this paper, we will focus on one-class classification approaches that only apply to information retrieval. Recently, two novel one-class classification approaches have been proposed to tackle the information retrieval problem:

1. Chen, Zhou and Huang [5] proposed the One-Class SVM (OC-SVM) to learn the boundary of positive data points, and demonstrated that OC-SVM algorithm with nonlinear Kernel is able to model nonlinear relationship effectively. However, with OC-SVM, there exists a problem on how to choose an appropriate kernel systematically to characterize data at hand.
2. Crammer and Chechik proposed the One-Class Information Bottleneck method (OCIB) [1] based on Information Bottleneck framework. Compared with OC-SVM, the constructed boundary is not biased to the global center of the whole data set, and is more sensitive to local structure in data set.

One problem with OCIB is that it only focuses on one cluster in large data set. Real-world data are often complex and contain multiple distinct regions that should be learned separately. It would be interesting to investigate an approach to integrate individual OCIB classifiers. In addition, the idea of OCIB is to use a ball to cover positive data points, and the parameter of the ball radius R need to be pre-determined in OCIB. To characterize a specific data set, a spectrum of R values is needed. Since different R values lead to different retrieval results, how these results can be integrated together is another interesting question.

To overcome the above problems, we introduce the ensemble learning into the OCIB approach, and propose the I-OCIB algorithm for image retrieval. Two main contributions of our paper are:

1. By employing Bagging into one-class classifiers, the proposed I-OCIB algorithm can deal with data with more than one mode;
2. we alleviate the problem of a pre-defined parameter in OCIB by Borda's method [6].

Our experimental results in section 4 illustrate the effectiveness of these two strategies.

2 One-Class Information Bottleneck

The goal of One-Class Information Bottleneck(OCIB) is to find a ball with fixed radius that covers as many samples as possible. While other approaches to one-class classification use a convex cost function to capture large-scale structures, the cost function adopted by OCIB is more sensitive to local structures, resulting in a classifier that focuses on “interesting/relevant” samples.

In OCIB, a ball radius R is treated as constant and known. Let C be the event that a point \mathbf{x} is assigned to the ball, and $p(C|\mathbf{x})$ be the probability that

this event happens. Then, the learning objective of OCIB is to find a set of probability $p(C|\mathbf{x}_j)$ for each data point \mathbf{x}_j and a center \mathbf{w} of a ball. The task can be formalized as an optimization problem based on Information Bottleneck framework [7] as following:

$$\min\{\beta\mathcal{D}(C, \mathbf{w}, X) + I(C; X)\} \quad (1)$$

where $X = \{\mathbf{x}_i\}$ is the data set; $\mathcal{D}(C, \mathbf{w}, X)$ represents average distortion accounting for how the ball matches the cluster of data points; $I(C; X)$ is the mutual information between X and C ; and β is a tradeoff factor between two terms in the formula.

In other words, the goal of OCIB is to find a simple and meaningful representation C for the given data set X . It can be seen from Formula (1) that the first term indicates how strongly the model C compresses the data, whereas the second term measures the accuracy of the obtained model. Note that β is a tradeoff factor between a model's accuracy and simplicity. The sequential IB algorithm(S-IB) has been adapted to solve above optimization problem [1].

3 Integrating OCIB Classifiers

The whole procedure of I-OCIB method includes two stages when applying to image retrieval: training stage and retrieval stage. In training stage, users' feedback about relevant image to a target topic is treated as positive training data. Based on these training data, a set of classifiers are learned to cover the data as much as possible. In retrieval stage, the relevance of new images in database are evaluated with the obtained classifiers. A ranking list is constructed by taking into consider the relevance estimated by all the classifiers. The higher an image is ranked in the list, the more likely it is relevant to a target topic.

In follow sections, we will describe the proposed I-OCIB which mainly extended in two ways: bagging OCIB classifiers and aggregating ranking lists. First, the bagging of OCIB classifiers involves both training and retrieval stage. It constructs an ensemble of classifiers to capture multiple clusters in training data set. Second, in retrieval stage, multiple ranking lists obtained from various parameter settings are integrated into a single list.

3.1 Bagging One-Class Classifiers

The OCIB method aims at using one ball to cover as many data points as possible. However, real world applications often involve data of multi-mode, which usually distribute in several distinctive regions. The problem now is how to construct several one-class classifiers for separate regions and then to integrate the results together.

We adopt the idea of Bagging [8] to construct an ensemble of classifiers based on the base learner. Each base learner is an OCIB classifier. After training, the results from base learners are averaged to get final decision. With the idea of Bagging, given a data set $\{\mathbf{x}_i|i = 1..n\}$, we produce a data set of size n by

re-sampling the original data with replacement in each iteration. Since OCIB is an unstable classifier, the Bagging process is able to produce a diverse ensemble of classifiers to focus on different parts of the training data. We typically run the OCIB for 20 iterations based on various derived training sets, yielding an ensemble of 20 classifiers. Suppose a ball size R is adopted in all these iterations. Then in iteration i , one OCIB classifier is trained and represented as a ball $B(R, \mathbf{w}_i)$, where \mathbf{w}_i is the center of the ball.

Algorithm 1. Generating an ensemble of OCIB Classifiers

Input: data set $X = \{\mathbf{x}_j, j = 1, \dots, n\}$; ball size R ;

Output: $\{\mathbf{w}_i, i = 1, \dots, 20\}$: the centers of the set of balls

1: **for** $i \in \{1, \dots, 20\}$ **do**

2: Data set $D_i \leftarrow$ Re-sample original data set with replacement

3: The center of ball i : $\mathbf{w}_i \leftarrow$ Restart OCIB learner 10 times with data set D_i , choose the best classifier, and record the center of the ball

4: **end for**

The algorithm for generating an ensemble of OCIB classifiers is described in Algorithm 1. It is worth noting that we restart OCIB algorithms 10 times and choose the result that can minimize Equation (1). Since sequential algorithm is only able to find a local optimum, this restarting strategy is taken to enhance the chance of finding an optimum.

After all the centers of balls are determined, given a new image \mathbf{x} , how should it be classified? Since each classifier has its own estimation on the relevance of an image, the results from this ensemble of classifiers need to be integrated. In addition, before this can be done, it is necessary to consider how these estimations are given.

OCIB algorithm classifies a new image as positive if it is located inside of a ball. In other words, the output is $\{1, -1\}$, either positive or negative. This kind of hard assignment simply divides unknown image set into two sets. However, in image retrieval, it is more preferred to have a ranking list with which users can request the first k most relevant images at their will. Therefore, I-OCIB instead checks the position of a new image relative to each ball, and gives the probabilities that a point belongs to a ball. The probability regarding to ball i can be modeled using a Gaussian function as follows:

$$f_i(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\|\mathbf{x}-\mathbf{w}_i\|}{\sigma^2}} \quad (2)$$

Assume positive data points follow Gaussian distribution with standard deviation σ , to allow 99.7% of the data points to fall into a ball, the radius R of the ball should be set as 3σ . If we let σ_i be $\frac{R_i}{3}$, we will have:

$$f_i(\mathbf{x}) = \frac{3}{\sqrt{2\pi}R} e^{-\frac{9\|\mathbf{x}-\mathbf{w}_i\|}{R^2}} \quad (3)$$

To integrate the individual probabilities $f_i(\mathbf{x})$ regarding to each ball, one single decision function $f(\mathbf{x})$ is defined as their average:

$$f(\mathbf{x}) = \frac{\sum_i f_i(\mathbf{x})}{20} \quad (4)$$

Note here 20 is the total number of OCIB base learners.

As a result, by altering the hard assignments, $\{-1, 1\}$, of new points' labels into soft assignments, $[-1, 1]$, the output of decision function regarding to each new data point can be used to sort all new images into a ranking list, with which the top ranked images are presented to end user.

3.2 Aggregating Ranking Lists

The ball radius R in Section 3.1, as an parameter, has crucial impact on the shape of decision function. With a smaller R , smaller clusters will be obtained, and the value of the decision function rises dramatically around the cluster centers; with greater R , larger clusters will be constructed, and the decision function is smoother and has less variation.

Crammer and Chechik's paper [1] reported retrieval results with OCIB at various settings of ball radius. Instead, we aim at combining different retrieval results into a integral part. Since at each setting of ball radius, a ranking list is obtained according to the output of decision function described in Section 3.1, the problem now becomes how to combine the multiple ranking lists into a single one.

Here, we adopt Borda's method [6] to incorporate multiple ranking lists into a single final ranking list. A score corresponding to the position of a candidate in each ranking list is assigned, and then scores derived from different ranking lists are sum up to determine final ranking. Borda's method is adopted here for rank aggregation because candidates' relative positions in individual ranking list are more meaningful than their score assigned directly by a decision function.

Algorithm 2. Rank aggregating

Input: n : the number of total data points; $g_k(\mathbf{x})$: Decision function given ball size R_k ($k = 1, \dots, \alpha$);

Output: Final ranking list τ

- 1: **for** $k \in \{1, \dots, \alpha\}$ **do**
 - 2: Construct a ranking list τ_k through sorting \mathbf{x} by $g_k(\mathbf{x})$ in decreasing order
 - 3: **for** $j \in \{1, \dots, n\}$ **do**
 - 4: Assign Borda score $B_k(\mathbf{x}_j)$ according to the position of \mathbf{x}_j in list τ_k
 - 5: **end for**
 - 6: **end for**
 - 7: **for** $j \in \{1, \dots, n\}$ **do**
 - 8: Total Borda score $B(\mathbf{x}_j) \leftarrow \sum_k B_k(\mathbf{x}_j)$
 - 9: **end for**
 - 10: Construct a rank list τ by sorting \mathbf{x}_j through sorting $B(\mathbf{x}_j)$ in decreasing order
-

Formally speaking, suppose function $g_k(\mathbf{x})$ represents the decision function $f(\mathbf{x})$ with ball radius set as R_k , $k = 1, \dots, \alpha$, where α is the number of different setting of ball radius. The choice of α and the setting of each R_k will be discussed in detail in Section 4. With the decision function $g_k(\mathbf{x})$, a ranking list τ_k can be determined by Borda’s method: for each candidate data point \mathbf{x} , Borda’s method assigns a score $B_k(x) =$ “the number of candidates ranked below \mathbf{x} in τ_k ”. The total Borda score $B(\mathbf{x})$ is defined as $\sum_k B_k(\mathbf{x})$. After sorting candidates by total Borda score in decreasing order, the final ranking list τ is constructed. The pseudo code of the rank aggregating algorithm is shown in Algorithm 2.

4 Experiments on Image Retrieval

4.1 Methodology

To examine the effectiveness of our approach, we use images from a video database from TREC2003 [9]. The whole database includes 25 subsets of video key frames. Each subset is on one topic and identified by a topic ID. For each data set, the relevance between key frames and particular topics is available. The video key frames are from video shot No.134 to No.248. Since there is no relevant key frames from shot No.134 to shot No.248 in data set No.118 and No.119, these two data sets are excluded in our experiment. Information about all data sets is summarized in Table 1.

I-OCIB is compared with an algorithm called A-OCIB, which is adapted from OCIB for image retrieval purpose. Although OCIB has been used for image retrieval in [1], it can only give labels to images in the form of hard assignments, which can not be used to construct a ranking list.

To facilitate image retrieval, A-OCIB is adapted in following two ways similar to I-OCIB: 1) The training outcome of A-OCIB is still only one classifier, but this classifier’s output is in the form of probability instead of hard assignment(as described in Section 3.1); 2) Rank aggregating technique using Borda’s method is employed so that the results of A-OCIB from various ball sizes can be incorporated into a single ranking list(as described in Section 3.2).

For both I-OCIB and A-OCIB, ten 2-fold tests are carried out. That is, half the relevant images are used for training and the other half for testing. For each data set, 10 different training and testing sets are constructed. When retrieval is carried out, a ranking list is constructed based on the other half relevant images and all irrelevant images to see if the relevant images can be retrieved successfully. The retrieval performance is measured using Mean Average Precision(MAP) and precision rate at the recall of 0.05, 0.1, 0.3 and 0.5 [10]. The average result of these 2-folds tests is recorded. After that, the pair wise one-tailed t -test is performed on the results of I-OCIB and A-OCIB at the significance level 0.025. To characterize images in the data sets, Gabor textures are extracted. Feature vector of 48 dimensions are constructed.

As mentioned in Section 3.2, one issue in aggregating ranking lists is how to choose the settings of various ball radiuses. In both I-OCIB and A-OCIB algorithm, we estimate the range of ball radius using maximal and minimal

pair-wise distance. Ball radius $R_k, k = 1, \dots, \alpha$ are uniformly generated in the estimated range. To decide the value of α , i.e. the number of various setting of R , values ranging from 5 to 20 are tested. It is found that retrieval accuracy improves when α goes from 5 to 10, but it remains nearly stable when it goes from 10 to 20. As a tradeoff, we choose α be 10.

Table 1. Data sets used in image retrieval

Topic ID	# of Relevant Key Frame	# of Irrelevant Key Frame
100	87	1311
101	104	1887
102	183	725
103	33	980
104	44	1307
105	52	1191
106	31	1214
107	62	1199
108	34	2158
109	16	1341
110	13	1300
111	13	1460
112	228	1738
113	62	1286
114	26	2457
115	106	2322
116	12	1235
117	640	2462
118	0	1089
119	0	1304
120	47	1550
121	95	1095
122	122	1196
123	45	940
124	10	1386

4.2 Results and Analysis

The results from algorithm I-OCIB and A-OCIB on 23 data sets are given in Table 2. MAP and Precision at recall rate of different levels are shown in five columns. Table 2 reveals that I-OCIB gives better performance on most data sets. When the number of relevant images in the whole data set is less than 40, i.e. the number of relevant images in the testing data set is less than 20, precision at recall of 0.05(i.e. 1/20) is not available. Similarly, precision at recall of 0.1 is not available if the number of relevant images is less than 20.

The results of pair-wise one-tailed t-test performed on I-OCIB and A-OCIB are shown in Table 3, where the results “significantly better”, “significantly worse” and “not significantly different” are denoted by 1, -1 and 0 respectively.

Table 2. Precisions comparison between I-OCIB and A-OCIB

	MAP		Precision(0.05)		Precision(0.1)		Precision(0.3)		Precision(0.5)	
	I-OCIB	A-OCIB	I-OCIB	A-OCIB	I-OCIB	A-OCIB	I-OCIB	A-OCIB	I-OCIB	A-OCIB
topic100	0.1015	0.0676	0.3119	0.1903	0.2302	0.1083	0.1285	0.0605	0.0545	0.0419
topic101	0.1824	0.1589	0.3133	0.2790	0.3737	0.3223	0.2634	0.2224	0.1550	0.1525
topic102	0.4989	0.4738	0.6686	0.7210	0.6204	0.6327	0.5656	0.5250	0.5413	0.4974
topic103	0.0483	0.0421	-	-	0.1941	0.1989	0.0438	0.0293	0.0346	0.0309
topic104	0.0554	0.0432	0.2505	0.1597	0.1160	0.0702	0.0513	0.0434	0.0396	0.0391
topic105	0.0273	0.0246	0.0648	0.0603	0.0249	0.0293	0.0300	0.0269	0.0272	0.0214
topic106	0.0553	0.0257	-	-	0.2991	0.0591	0.0594	0.0265	0.0364	0.0183
topic107	0.0514	0.0337	0.1207	0.0972	0.0896	0.0359	0.0476	0.0288	0.0432	0.0320
topic108	0.0191	0.0152	-	-	0.0635	0.0526	0.0214	0.0130	0.0182	0.0128
topic109	0.0110	0.0110	-	-	-	-	0.0174	0.0114	0.0076	0.0087
topic110	0.2559	0.0924	-	-	-	-	0.4428	0.1062	0.3083	0.0057
topic111	0.0507	0.0072	-	-	-	-	0.0119	0.0077	0.0064	0.0068
topic112	0.1128	0.1380	0.1494	0.2369	0.1560	0.2192	0.1269	0.1495	0.1127	0.1237
topic113	0.0681	0.0490	0.5311	0.2902	0.2301	0.0849	0.0485	0.0413	0.0331	0.0365
topic114	0.0719	0.0267	-	-	0.5969	0.2372	0.0386	0.0148	0.0109	0.0072
topic115	0.0364	0.0358	0.0932	0.0736	0.0617	0.0413	0.0366	0.0319	0.0301	0.0282
topic116	0.0821	0.0233	-	-	-	-	0.4119	0.0721	0.0171	0.0115
topic117	0.1924	0.2594	0.3388	0.6105	0.2692	0.4921	0.1919	0.2711	0.1730	0.2048
topic120	0.3113	0.1297	0.9333	0.6725	0.7300	0.4861	0.5235	0.1518	0.2979	0.0359
topic121	0.3324	0.1872	0.8686	0.4747	0.7889	0.4737	0.5770	0.2236	0.1855	0.1252
topic122	0.4022	0.3519	1.0000	0.6748	0.9667	0.6974	0.6635	0.6030	0.2766	0.3182
topic123	0.0574	0.0505	0.3868	0.1722	0.1672	0.0798	0.0365	0.0518	0.0407	0.0463
topic124	0.0359	0.0120	-	-	-	-	0.1424	0.0280	0.0130	0.0116
Average	0.1330	0.0982	0.4308	0.3366	0.3321	0.2401	0.1948	0.1191	0.1071	0.0790

As can be seen from the last column of Table 3, the averages of MAP and precision values are greater than 0 among 13 data sets, which means the average performance of I-OCIB over these data sets is better than that of A-OCIB. The average values for remaining data set are equal to 0 except data set No.117. To further analyze this phenomenon, let us refer to Table 1 for characteristics of data set No.117. It can be seen that data set No.117 has larger proportion of relevant images than other data sets. I-OCIB’s performance on this data set is worse than that of A-OCIB probably because the relevant images in this data set form a simpler and more compact cluster so that it is probably better to use a simple model to characterize the data, as OCIB does using a single ball. Actually, in typical image retrieval systems, the available relevance information is usually only about small proportion of image data set. Therefore, as a “needle in a haystack” approach, I-OCIB has its advantage in most cases.

Finally, it can be seen from the last line of Table 3 that the average value of Precision at recall of 0.3 and 0.5 is greater than that at recall of 0.05. This observation suggests that, although the retrieval accuracy is only slightly better with I-OCIB than with A-OCIB when a user only looks for the first several most relevant images, the performance is significantly better when a user wants to find more interesting images.

Table 3. Pair wise one-tailed T-test between I-OCIB and A-OCIB

Topic ID	MAP	P(0.05)	P(0.1)	P(0.3)	p(0.5)	Average
100	1	0	1	1	1	0.80
101	0	0	0	0	0	0.00
102	0	0	0	1	1	0.40
103	0	-	0	0	0	0.00
104	0	0	0	0	0	0.00
105	1	0	0	0	1	0.40
106	1	-	1	1	1	1.00
107	0	0	1	1	1	0.60
108	0	-	0	0	0	0.00
109	0	-	-	0	0	0.00
110	0	-	-	1	0	0.33
111	1	-	-	0	0	0.33
112	0	0	0	0	0	0.00
113	0	0	0	0	0	0.00
114	1	-	1	0	1	0.75
115	0	0	0	1	1	0.40
116	1	-	-	1	0	0.67
117	-1	-1	-1	-1	-1	-1.00
120	1	0	0	1	1	0.60
121	1	1	1	1	0	0.80
122	0	1	1	0	0	0.40
123	0	0	0	0	0	0.00
124	0	-	-	0	0	0.00
Average	0.304	0.071	0.277	0.347	0.304	0.26

As a summary, the experimental result indicates that the performance of I-OCIB is better than that of A-OCIB in most cases. Because I-OCIB differs from A-OCIB in that an ensemble of classifiers, instead of a single classifier, is generated and integrated, our strategy of using bagging of one-class classifiers is justified.

5 Conclusion

In this paper, we introduce an integrated version of OCIB, named I-OCIB, for image retrieval. Two stages of integration are adopted to aggregate results from multiple classifiers and from various parameter settings respectively. The experimental results show that bagging of OCIB classifiers is able to identify multiple clusters in data sets so that the retrieval accuracy is improved.

Compared with other one-class classification method for image retrieval, such as one-class SVM, I-OCIB is free of parameter selection by integrating multiple ranking lists from different parameters into one; while the employment of one-class SVM requires careful selection of an appropriate kernel and fine tuning of parameters.

For simplicity, we only considered data points in the input space in this paper. It is expected that the performance could be further improved if kernel function is introduced into the distance measure to map points into appropriate feature space, which we plan as a future work.

References

1. Crammer, K., Chechik, G.: A needle in a haystack: Local one-class optimization. In: IEEE International Conference on Machine Learning (ICML). (2004)
2. Zhou, X.S., Huang, T.S.: Small sample learning during multimedia retrieval using biasmap. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2001)
3. Tesic, J., Manjunath, B.: Nearest neighbor search for relevance feedback. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). (2003)
4. Yan, R., Hauptmann, A., Jin, R.: Negative pseudo-relevance feedback in content-based video retrieval. In: ACM Multimedia. (2003)
5. Chen, Y., Zhou, X., Huang, T.S.: One-class svm for learning in image retrieval. In: IEEE International Conference on Image Processing (ICIP). (2001)
6. Borda, J.C.: Histoire de l'academie royale des sciences. In: Memoire sur les elections au scrutin. (1781)
7. Tishby, N., Pereira, F., Bialek, W.: The information bottleneck method. In: Allerton Conference on Communication, Control, and Computing. (1999)
8. Breiman, L.: Bagging predictors. *Machine Learning* **24** (1996) 123–140
9. : (TREC video track : <http://www-nlpir.nist.gov/projects/tv2003/tv2003.html>)
10. Muller, H., Muller, W., Squire, D.M., Marchand-Maillet, S., Pun, T.: Performance evaluation in content-based image retrieval: overview and proposals. *Pattern Recognition Letters* **22** (2001) 593–601

Incremental Discretization for Naïve-Bayes Classifier

Jingli Lu, Ying Yang, and Geoffrey I. Webb

Clayton School of Information Technology, Monash University
VIC 3800, Australia

{Jingli.Lu, Ying.Yang, Geoff.Webb}@infotech.monash.edu.au

Abstract. Naïve-Bayes classifiers (NB) support incremental learning. However, the lack of effective incremental discretization methods has been hindering NB's incremental learning in face of quantitative data. This problem is further compounded by the fact that quantitative data are everywhere, from temperature readings to share prices. In this paper, we present a novel incremental discretization method for NB, *incremental flexible frequency discretization* (IFFD). IFFD discretizes values of a quantitative attribute into a sequence of intervals of flexible sizes. It allows online insertion and splitting operation on intervals. Theoretical analysis and experimental test are conducted to compare IFFD with alternative methods. Empirical evidence suggests that IFFD is efficient and effective. NB coupled with IFFD achieves a rapport between high learning efficiency and high classification accuracy in the context of incremental learning.

1 Introduction

Naïve-Bayes classifiers (NB) are simple yet powerful [3, 4]. Its efficiency has witnessed its widespread deployment in real-world applications including medical diagnosis, fraud detection, email filtering and webpage prefetching. One key contributing factor to NB's efficiency is its capability of incremental learning from qualitative data [5, 6]. To accommodate a new training instance, NB only needs to update relevant entries in its probability table. This often has a much lower cost than non-incremental approaches that have to rebuild a new classifier from scratch in order to include new training data.

If learning involves quantitative data, NB often uses discretization to transform them into qualitative data. Briefly speaking, discretization groups sorted values of a quantitative attribute into intervals, treats each interval as a qualitative value and inputs them into NB. Ideally, discretization should also be incremental in order to be coupled with NB. When receiving a new training instance, incremental discretization is expected to be able to adjust intervals' boundaries and statistics, using only the current intervals and this new instance instead of re-accessing previous training data. Unfortunately, the majority of existing discretization methods are not oriented to incremental learning. To update discretized intervals with new instances, they need to add those new instances into previous training data, and then re-discretize on basis of the updated complete training data set. This is detrimental to NB's efficiency by

inevitably slowing down its learning process. Hence there is a real and immediate need for appropriate incremental discretization methods for NB.

Some preliminary research has been contributed to exploring incremental discretization for NB. A representative is the method PiD proposed by Gama and Pinto [6]. PiD is based on a two layer histograms and is efficient in term of time and space complexity. However it can be sub-optimal in that the histograms are not exact and the splitting operation in the first layer possibly produces inexact counters.

This paper proposes a new effective approach, incremental flexible frequency discretization (IFFD). IFFD is based on fix frequency discretization (FFD) that has been demonstrated as a very efficient and effective discretization method for NB in the context of non-incremental learning [10, 11]. IFFD produces intervals with flexible sizes, stipulated by a lower bound and an upper bound. An interval is allowed to accept new values until its size reaches the upper bound. An interval whose size exceeds the upper bound is allowed to split if the resulting smaller intervals each have a size no smaller than the lower bound. Accordingly IFFD is able to incrementally adjust discretized intervals, effectively update associated statistics and efficiently synchronize with NB's incremental learning.

The remaining of this paper is organized as follows. Section 2 introduces naïve-Bayes learning and discretization. Section 3 explains the motivation and methodology of IFFD. Section 4 describes rival incremental methods from related work. Section 5 analyzes each alternative method's complexity in terms of learning time and space. Section 6 conducts experiments to verify IFFD's efficacy and efficiency. Section 7 gives concluding remarks.

2 Discretization for Naïve-Bayes Learning

2.1 Naïve-Bayes Classifier (NB)

Assume that an instance I is a vector of attribute values $\langle x_1, x_2, \dots, x_n \rangle$, each value being an observation of an attribute X_i ($i \in [1, n]$). Each instance can have a class label $c_i \in \{c_1, c_2, \dots, c_k\}$, being a value of the class variable C . If an instance has a known class label, it is a training instance. If an instance has no known class label, it is a testing instance. The dataset of training instances is called the training dataset. The dataset of testing instances is called the testing dataset.

To classify an instance $I = \{x_1, x_2, \dots, x_n\}$, NB estimates the probability of each class label given I , $P(C = c_i | I)$ using Formula (1, 2, 3,4). Formula (2) follows (1) because $P(I)$ is invariant across different class labels and can be canceled. Formula (4) follows (3) because of NB's attributes independent assumption. It then assigns the class with the highest probability to I . NB is called naïve because it assumes that attributes are conditionally independent of each other given the class label. Although its assumption is sometimes violated, NB is able to offer surprisingly good classification accuracy in addition to its very high learning efficiency, which makes NB popular with numerous real-world classification applications [2, 8].

$$\begin{aligned}
& P(C = c_i | I) \\
&= \frac{P(C = c_i)P(I | C = c_i)}{P(I)} \tag{1} \\
&\propto P(C = c_i)P(I | C = c_i) \tag{2} \\
&= P(C = c_i)P(\langle x_1, x_2, \dots, x_n \rangle | C = c_i) \tag{3} \\
&= P(C = c_i) \prod_{j=1}^n P(X_j = x_j | C = c_i) \tag{4}
\end{aligned}$$

In naïve-Bayes classifier, the class type must be qualitative while the attribute type can be either qualitative or quantitative. When an attribute X_j is quantitative, it often has a large or even infinite number of values. As a result, the conditional probability that X_j takes a particular value x_j given the class label c_i covers very few instance if there is any at all. Hence it is not reliable to estimate $P(X_j=x_j|C=c_i)$ according to the observed instances. One common practice to solve the problem of quantitative data for NB is discretization.

2.2 Discretization

Discretization is a popular approach to transforming quantitative attributes into qualitative ones for NB. It groups sorted values of a quantitative attribute into a sequence of intervals, treats each interval as a qualitative value, and maps every quantitative value into a qualitative value according to which interval it belongs to. In the paper, the boundaries among intervals are sometimes referred to as *cut points*. The number of instances in an interval is referred to as *interval frequency*. The total number of intervals produced by discretization is referred to as *interval number*.

Incremental discretization aims at efficiently updating discretization intervals and associated statistics upon receiving each new training instance. Ideally, it does not require to access historical training instances to carry out the update. Instead it only needs the current intervals (with associated statistics) and the new instance.

3 Incremental Flexible Frequency Discretization

In this section, we propose a novel incremental discretization method, *incremental flexible frequency discretization* (IFFD). It is motivated by the pros and cons of fixed frequency discretization (FFD) in the context of naïve-Bayes learning and incremental learning [10, 11].

3.1 Fixed Frequency Discretization (FFD)

FFD has been proposed as an effective and efficient discretization method for naïve-Bayes learning through bias and variance management. It has been found that large interval size tends to increase NB's classification bias while large interval number tends to increase NB's classification variance [12]. To discretize a quantitative attribute, FFD sets a sufficient interval frequency, $m = 30$ [11,13]. It then discretizes

the ascendingly sorted values into intervals of frequency m . By introducing m , FFD aims to ensure that each interval has sufficient training instances for NB probability estimation, reducing classification variance error. On top of that, by not limiting the number of intervals formed, more intervals can be formed as the training data size increases, reducing classification bias error. Empirical evidence has demonstrated that FFD helps NB achieve lower classification error than alternative discretization methods do.

Although FFD is effective for naïve-Bayes learning, it is developed in the context of non-incremental learning. Every time when new training instances have arrived, FFD has to rebuild the discretization intervals from scratch. It is possible that even a single instance can push every boundary to (unnecessarily) move. For example, FFD discretizes the sorted values of a quantitative attribute into the following intervals. For simplicity, we assume $m = 3$:

{3.0, 4.0, 4.3}, {4.5, 5.1, 5.9}, {6.0, 6.1, 6.2}, {6.5, 6.7, 6.8}, {6.9, 7.1}

Suppose that a new instance has come with this attribute being value “5.2”. According to the current cut points, the appropriate interval to accommodate “5.2” is {4.5, 5.1, 5.9}. Inserting “5.2” into {4.5, 5.1, 5.9} will make the interval frequency increase to 4, which is greater than FFD’s specified threshold 3. Hence we need to move “5.9” out of the updated interval {4.5, 5.1, 5.2, 5.9} and insert it into the interval {6.0, 6.1, 6.2}, which produces another interval {5.9, 6.0, 6.1, 6.2} whose frequency is greater than 3. Following the same lines of reasoning, we have to move “6.2” into the next one and so on so forth until the last interval. As a result, the updated intervals are {3.0, 4.0, 4.3}, {4.5, 5.1, 5.2}, {5.9, 6.0, 6.1}, {6.2, 6.5, 6.7}, {6.8, 6.9, 7.1} and almost every cut point has been changed.

In this case, FFD has to rebuild the intervals and NB’s conditional probability table from the second interval all the way to the last one. In the best situation, the new instance is inserted into the last interval and the computation cost can be non-trivial. However in the worst situation such as when the new instance is inserted into the first interval, FFD is extremely inefficient. The reason is that FFD specifies a fixed interval frequency. This observation motivates our new incremental discretization approach as follows.

3.2 Incremental Flexible Frequency Discretization (IFFD)

IFFD sets its *interval frequency* to be a range [$minBinsize$, $maxBinsize$) instead of a single value m . The two arguments, $minBinsize$ and $maxBinsize$, are respectively the minimum and maximum frequency that IFFD allows intervals to assume. Whenever a new value arrives, IFFD first inserts it into the interval that the value falls into. IFFD then checks whether the updated interval’s frequency reaches $maxBinsize$. If not, it accepts the change and update statistics accordingly. If yes, IFFD splits the overflowed interval into two intervals under the condition that any of the resulting intervals has its frequency no less than $minBinsize$. Otherwise, even if the interval overflows because of the insertion, IFFD does not split it, in order to prevent high classification variance [10,11]. In the current implementation of IFFD, $minBinsize$ is set as 30, following FFD’s lines of reasoning so as to minimize classification bias and variance; and $maxBinsize$ is set as twice of $minBinsize$.

By assuming a more flexible interval frequency, IFFD is able to solve FFD's dilemma in incremental learning. Recall the example in Section 3.1. Assume $minBinSize = 3$ and hence $maxBinSize = 6$. When the new attribute value "5.2" comes, IFFD inserts it into the second interval {4.5, 5.1, 5.9}. That interval is hence changed into {4.5, 5.1, 5.2, 5.9} whose frequency (equal to 4) is still within [3, 6). So what we need do is only to modify NB's conditional probabilities related to the second interval. Assume another two new attribute values "5.4, 5.5" have come and are again inserted into the second interval. This time, the interval {4.5, 5.1, 5.2, 5.4, 5.5, 5.9} has a frequency as 6, reaching $maxBinSize$. Hence IFFD will split it into {4.5, 5.1, 5.2} and {5.4, 5.5, 5.9} whose frequencies are both within [3, 6). Then we only need to recalculate NB's conditional probabilities related to those two intervals. By this means, IFFD makes the update process local, affecting a minimum number of intervals and associated statistics. As a result, incremental discretization can be carried out very efficiently,

Table 1 shows the pseudo codes of the IFFD algorithm. For simplicity, we just consider one attribute value to update the discretization intervals and classifier and assume all attribute values are different. $cutPoints$ is the set of cut points of discretization intervals. $counter$ is the conditional probability table of the classifier. $minBinSize$ is minimum bin size. IFFD will update the $cutpoints$ and $counter$ according to new attribute value V . $classLabel$ is the class label of V .

Table 1. Pseudo Codes of IFFD

```

Function: IFFD(cutPoints, counter, minBinSize, V,
classLabel)
//If V is greater than the last cut point
if(V > cutPoints[size-2] ) //size is the interval
    number
    // cutPoints counts from 0
    { insert V into interval[size-1];
      counter[size-1][classLabel]++;
      chaInt = size-1; //record changed interval
    }
else
    { for(j = 0; j < size-1; j++)
      if(V <= cutPoints[j])
        { insert V into interval[j];
          intFre[j]++;
          counter[j][classLabel]++; //update contingency
          table
          chaInt = j; //record the interval which has been
          changed
          break;
        }
    }
if(intFre[chaInt] > minBinSize*2)
{ get new cut point; //split interval[chaInt] into two
  c1 and c2
  insert the new cut point into cutPoints;
  calculate counter[c1] and counter[c2]; //update
  contingency table
}

```

Please be noted that identical values are always kept in the same interval. For example, if the interval is {4.5, 5.1, 5.2, 5.2, 5.2, 5.6, 5.9}, IFFD will not split it into {4.5, 5.1, 5.2} and {5.2, 5.6, 5.9} even though its frequency has exceeds *maxBinsize* (=6). Nor will IFFD split it into {4.5, 5.1} and {5.2, 5.2, 5.2, 5.6, 5.9} or {4.5, 5.1, 5.2, 5.2, 5.2} and {5.6, 5.9}, because the smaller interval frequency is less than *minBinsize* (=3).

4 Rival Methods from Related Work

4.1 Move Boundary FFD (MFFD)

An intuitive way to relieve FFD's dilemma in incremental learning (Section 3.1) is to just move the interval boundaries instead of redoing discretization. We name this method *move boundary FFD* (MFFD). For the same example as in Section 3.1, if MFFD is applied, we just calculate the change of every interval. The second interval {4.5, 5.1, 5.9} has been inserted into an attribute value "5.2" and delete an attribute value "5.9", then we just modify the conditional probability. Attention is only paid to the inserted and deleted values. Do like this until the last interval. NB coupled with MFFD has the same classification accuracy as NB coupled with FFD, but the former is more efficient than the latter.

Table 2. The Pseudo Codes of MFFD

```

Function: MFFD(cutPoints, counter, V, classLabel)
curVal=V; curClasslabel= classLabel;
for(j = 0; j < size-1; j++) //size is the interval
    number
{ if(curVal =<= cutpoints[j])
  { // interval[j] is the jth interval of the attribute
  insert curVal into interval[j];
  //fre is the specified interval frequency
  // V[j][fre-1] is the last value in interval[j]
  remove V[j][fre-1] from interval[j];
  cutPoints[j]= V[j][fre-2]; //modify cut points
  counter[j][curClasslabel]++; //update contingency
  table
  counter[j][ V[j][fre-1].class]--;
  curVal = V[j][fre-1];
  curClasslabel = V[j][fre-1].class;
  }
}
If(fre[size-1] < split threshold)
{ insert curVal into interval[size-1];
  counter[size-1][curClasslabel]++;
}
else
{ split interval[size-1];
  calculate counter[size-1] and counter[size];
  size = size+1;
}

```

Table 2 presents the pseudo codes of MFFD. For simplicity, we just consider one attribute value to update the discretization intervals and classifier and assume all attribute values are different. *cutpoints* is the set of cut points of discretization intervals. *counter* is the conditional probability table of the classifier. MFFD will update the *cutpoints* and *counter* according to new attribute value V . *classLabel* is the class label of V .

4.2 Partition Incremental Discretization (PiD)

PiD is a two layer histograms incremental discretization method [6]. The first layer based on equal-width or equal-frequency determines the candidate cut points according to observed values. At this layer, the interval number is significantly greater than the final interval number. For example, the final interval number is 40, probably the interval number in the first layer is 200. For incremental learning, it inserts the incremental data into the appropriate intervals. To any interval whose frequency is greater than the specified threshold, it will be split. Because in this layer, it does not store the historical data, the splitting result is inaccurate. It just splits an interval into two uniformly. The second layer merges the intervals gained at the first layer. In the second layer, PiD can construct the final discretization interval by any different strategies. Namely, PiD discretizes quantitative attributes twice. At first, it uses a loose interval number to discretize; and then merges intervals if necessary. The main advantage of PiD is low time and space complexity, but during the splitting operation in the first layer, it possibly produces inexact counters.

4.3 Kernel Density Estimation (KDE)

A counterpart of discretization is probability density estimation to handle quantitative attributes for NB. It models each quantitative attribute by some continuous probability distribution. Probability density estimation methods can manipulate quantitative attributes for naïve-Bayes incremental learning. A representative method is *kernel density estimation* (KDE) [7].

KDE is a non-parametric approach that does not assume the underlying distribution to take any particular form. Instead it estimates from sample values. This circumvents unsafe assumptions and achieves better accuracy because of real world diversity. For KDE, it calculates the conditional class probability as:

$$P(X_j = x_j | C = c_i) = \frac{1}{n_i} \sum_k \int_l^h f(x_j, \mu_k, \sigma_c) dx_j \quad (5)$$

where n_i is the number of training instances with class label c_i . For every quantitative attribute of testing instance, KDE has to perform probability calculation n_i times to get $P(X_j=x_j|C=c_i)$. If the instance number is large, it has a potential computational problem.

5 Time and Space Complexity Comparison

In this section, we analyse the time and space complexity incurred by accommodating a new training instance. It includes updating the discretized intervals as well as updating required probabilities for NB.

5.1 Time Complexity

In the following, n is abbreviation of instance number; k is the attribute number; C is the number of class label, specified *Interval Frequency* is abbreviated by $IntF$, $IntN$ represents *Interval Number*, then $IntN=n/IntF$.

5.1.1 Train Time Complexity on a New Instance

Train Time Complexity of MFFD

Assume the probability of the new attribute value inserting into every interval is equal. $IntN - i + 1$ is the number of intervals which has to be changed, where i is the appropriate interval for the new instance. Inserting an instance into the interval while deleting another one from the interval has a constant cost in time complexity $O(1)$. So for every incremental attribute value, the training time complexity is presents in equation (6). This complexity repeating for k attribute is $O(k)$, so resulting in the totally complexity is $O(n)*O(k)=O(nk)$.

$$\frac{\sum_{i=1}^{IntN} (IntN - i + 1) * O(1)}{IntN} = \frac{\frac{IntN(IntN + 1)}{2} * O(1)}{IntN} = \frac{IntN + 1}{2} * O(1) = \frac{\frac{n}{IntF} + 1}{2} * O(1) = O(n). \quad (6)$$

Train Time Complexity of PiD

The time complexity of PiD depends on the discretization methods selected in each layer. In our experiments, we select equal-width and PD for the two layers separately (the reason that we select them is explained in 4.2.1). Here we just analyze time complexity in this situation.

In the first layer, when the interval frequency of a specified interval is greater than a user defined threshold (a percentage of the total instance number), the interval will be split. The more interval number is defined in the first layer, the less probability some interval will be split. In the first layer, the interval frequency is a large number, so the time for splitting operation can be ignored. The input of the second layer is the intervals and associated statistics of first layer. If the interval gained in the first layer is m , then the time complexity of PiD is $O(mk)$.

Train Time Complexity of IFFD

Assume the probability of the new attribute value inserting into some interval is equal. Max is the maximum interval frequency; Min is the minimum interval frequency.

When a new attribute value inserts into the appropriate interval, the probability that the interval does not split is $\frac{Max - Min}{Max - Min + 1}$. In this situation, the operation is just to insert the new instance. Inserting an instance into the interval has a constant cost in time complexity $O(1)$. The probability that the interval splits is $\frac{1}{Max - Min + 1}$. In this situation, the operation is to recalculate the conditional probability table of the two new intervals and change the cut points. For a single attribute, if the data structure of *cutPoints* is array, the time complexity is presented in equation (7), $\frac{IntN}{2}$ means the number of cut points have to move, when insert a new cut point into the *cutPoints*. And if tree or list structure is selected, the time complexity is demonstrated as equation (8). This complexity repeating for k attribute is $O(k)$, so resulting in the totally complexity for array structure is $O(n)*O(k)=O(nk)$ and for tree structure is $O(1)*O(k)=O(k)$. In our experiment, we select array structure to store *cutPoints*, because our select Weka as the platform, in Weka, *cutPoints* is stored in an array.

$$\frac{(Max - Min) * O(1)}{Max - Min + 1} + \frac{1}{Max - Min + 1} * (Min + 1 + \frac{IntN}{2}) = O(n). \quad (7)$$

$$\frac{(Max - Min) * O(1)}{Max - Min + 1} + \frac{Min + 1}{Max - Min + 1} = \frac{2Min + 1}{Min + 1} \approx 2 = O(1). \quad (8)$$

Train Time Complexity of KDE

At training time, KDE just store the attribute values, so its time complexity is $O(k)$.

5.1.2 Test Time Complexity on a New Instance

Test Time Complexity of MFFD, IFFD and PiD

For every class label, the classifiers which manipulate quantitative attributes by discretization methods can get the conditional probability from the conditional probability table directly, so testing time complexity on the new instance is $O(Ck)$.

Test Time Complexity of KDE

At testing time, from equation (5) we can see, for every class label c_i and every quantitative attributes, KDE must evaluate f for every observed different attribute value whose class label is in class c_i . So the testing time complexity of KDE is $O(nk)$.

5.2 Space Complexity

5.2.1 Space Complexity of MFFD, IFFD and KDE on a New Instance

MFFD, IFFD and KDE have to store the historical quantitative attributes, so their space complexity is $O(nk)$.

MFFD has to change the cut points and modify the conditional probability table, so historical quantitative attributes are necessary.

For IFFD, when the interval frequency of some interval exceeds the threshold, the interval has to be split. Historical quantitative data is necessary to splitting operation. So IFFD must store the historical quantitative attribute values for every instance. But for every new instance, the modified interval is just one: split it or insert a point into it, namely the adjustment is local. So we can store the historical data in external storage. When change is necessary, we copy it from external storage to memory. With the development of hardware, storage is not a big problem.

KDE must store every different quantitative attribute value for every class label. To classify an instance, KDE has to access every attribute value to calculate the conditional class probability. So it is necessary to store the attributes values in the memory. However memory store is more expensive than external storage. If for every class label there are many duplicate quantitative attribute values, KDE has a lower space then MFFD and IFFD; otherwise their storage space are equal.

5.2.2 Space Complexity of PiD on a New Instance

Splitting operation in PiD is to split an interval uniformly. PiD does not need to store historical quantitative attribute values. It just stores the interval information which gained at the first layer. So its space complexity is $O(m)$, where m is the number of interval in the first layer. Compared with other methods, PiD has the lowest space complexity.

The time and space complexity are summarized in Table 3.

Table 3. Algorithmic complexity. n is abbreviation of instance number; k is the attribute number; C is the number of class label; m is the number of interval number in the first layer for PiD.

Method		MFFD	IFFD	PiD	KDE
Time Complexity	Training	$O(nk)$	$O(nk)$ (Array) $O(k)$ (Tree)	$O(mk)$	$O(k)$
	Testing	$O(Ck)$	$O(Ck)$	$O(Ck)$	$O(nk)$
Space Complexity		$O(nk)$	$O(nk)$	$O(mk)$	$O(nk)$

6 Experimental Evaluation

In this section, we compare the incremental learning performance of NB when coupled with IFFD, PiD, MFFD and KDE respectively to handle quantitative attributes.

6.1 Data

The experiments use a large suite of 30 benchmark datasets from the UCI machine learning repository [1]. For the purpose of incremental learning, the chosen datasets each have more than 500 instances. Table 4 describes the statistics of each dataset.

Table 4. Experimental Datasets. For each dataset, *Size* is the number of instances, *Qa* is the number of quantitative attributes, *Ql* is the number of qualitative attributes and *C* is the number of classes.

ID	Dataset	Size	Qa	Ql	C	ID	Dataset	Size	Qa	Ql	C
1	cylinder-bands	540	20	19	2	16	Abalone	4177	8	0	3
2	balance-scale	625	4	0	3	17	spambase	4601	57	0	2
3	credit-a	690	6	9	2	18	waveform-5000	5000	40	0	3
4	breast-w	699	9	0	2	19	page-blocks	5473	10	0	5
5	diabetes	768	8	0	2	20	optdigits	5620	48	0	10
6	vehicle	846	18	0	4	21	satellite	6435	36	0	6
7	anneal	898	6	32	6	22	Musk2	6598	166	0	2
8	vowel	990	10	3	11	23	pioneer	9150	30	6	57
9	German	1000	7	13	2	24	Thyroid	9169	7	22	20
10	cmc	1473	2	7	3	25	ae	9961	12	0	9
11	yeast	1484	7	1	10	26	pendigits	10992	16	0	10
12	volcanoes	1520	3	0	4	27	Sign	12546	8	0	3
13	mfeat-zernike	2000	47	0	10	28	letter	20000	16	0	26
14	segment	2310	19	0	7	29	Adult	48842	6	8	2
15	hypothyroid	3772	7	23	4	30	Shuttle	58000	9	0	7

6.2 Design

For each instance, we randomly shuffle the instances and use the first 200 instances to initialize an NB classifier. The remaining instances come one after the other. Each instance is to be classified by the current NB first. Its true class label is then made known to the classifier which takes it as a new training instance. Accordingly, the discretized intervals are updated and so is the classifier. Then the next instance comes and the same procedure runs again, and so on so forth until the last instance is classified. We call this complete process a *trial*. We conduct five trails and average their classification error rates.

For IFFD, *minBinSize* is 30 while *maxBinsize* is 60. For PiD, the first layer is equal-width discretization and the interval number is 200 [5]. In the second layer, we choose to proportional discretization [9], which has been demonstrated efficient and work well [9].

Statistically a win/draw/lose record is calculated when we compare IFFD against each alternative method. The record represents the number of data sets in which IFFD respectively beats, tie with or loses to the rival method. A one-tailed binomial sign test will be applied to the record. If its result is less than the critical level of 0.05, the wins against losses are statistically significant, supporting the claim that IFFD has a systematic (instead of by chance) advantage over the rival method.

6.2.1 Comparing at Ten Observation Points

Along the time line, 10 observed classification error rates are recorded when 10%, 20%, 30%, ..., 100% of instances have been classified respectively. At every observation point, we calculate the win/draw/lose records on classification error rate when comparing IFFD against alternative methods. Table 5 lists the records as well as their sign test results.

Table 5. Classification error win/draw/lose records on 10 observation points

Method		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
IFFD & PiD	Win	20	19	19	20	22	21	20	21	18	19
	Draw	1	0	0	0	0	0	1	1	2	1
	Lose	9	11	11	10	8	9	9	8	10	10
	Sign test	0.031	0.1	0.1	0.049	0.008	0.021	0.031	0.012	0.092	0.068
IFFD & MFFD	Win	14	14	15	17	17	16	17	18	17	17
	Draw	1	2	0	1	0	0	0	0	0	2
	Lose	15	14	15	12	13	14	13	12	13	11
	Sign test	0.644	0.575	0.572	0.229	0.292	0.428	0.292	0.181	0.292	0.172
IFFD & KDE	Win	17	15	16	18	17	19	19	19	19	19
	Draw	0	0	0	0	0	0	0	0	0	0
	Lose	13	15	14	12	13	11	11	11	11	11
	Sign test	0.292	0.572	0.428	0.181	0.292	0.1	0.1	0.1	0.1	0.1

At every observation points, we also record the arithmetic mean of each method’s classification error rate averaged on 30 datasets, as in figure 1.

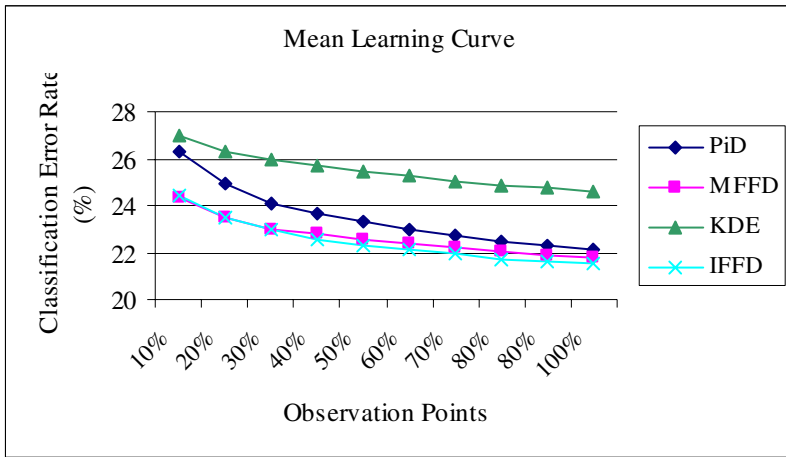


Fig. 1. Incremental Learning Curve. Comparing the classification error rate of naïve-Bayes classifiers which use the 4 methods to deal with quantitative attributes respectively at the 10 observation points, we can see, the error rate of IFFD is marginally lower than that of MFFD’s for the whole learning curve, the separation between IFFD and PiD becomes smaller and smaller with instances increasing. IFFD has substantially lower error rate than KDE.

In general, the classification error rate decreases gradually while more training instances are available. The error rate of IFFD is marginally lower than that of MFFD’s for the whole learning curve. There is a larger gap between IFFD and PiD at the beginning, which shrinks with time going on. IFFD has substantially lower error rate than KDE and its leading position remains through the whole learning period. The learning curve of PiD and KDE have small gaps at the beginning which enlarges later.

Specifically, to compare IFFD against PiD, IFFD is statistically more accurate than PiD at the 0.05 critical level when the training data size is medium (from the column 40% to the column 80%). On the other hand, IFFD is not significantly better than PiD when the training data size is extremely small or large. We suggest the reason that PiD employs proportional discretization at its second layer, which controls the interval frequency better than IFFD's interval [30,60) does.

For discretization, large interval frequency tends to produce low variance but high bias while large interval number tends to produce low bias but high variance. Proportional discretization attains equal bias and variance reduction by setting both interval frequency and interval number to be square root of the number of training instances, a strategy that has been demonstrated to react sensibly to varying training data size [9]. Figure 2 shows the ideal interval frequency's changing while training instances increase from 1 to 5000. From figure 3, we can see that when instances are fewer than 900, the ideal interval frequency should be less than 30, and when instances are more than 3600, the ideal interval frequency should be greater than 60. However, the current implementation of IFFD only allows the interval frequency to vary in the interval [30, 60). Hence for small datasets, IFFD's interval frequency can be too big; whereas for large datasets, IFFD's interval frequency can be too small. This explains why IFFD's performance is not significantly better than PiD's at the beginning and at the very end of the incremental learning curve. Our understanding of this issue also leads to an interesting future research issue, that is, how to make IFFD's flexible frequency range change according to different training data size.

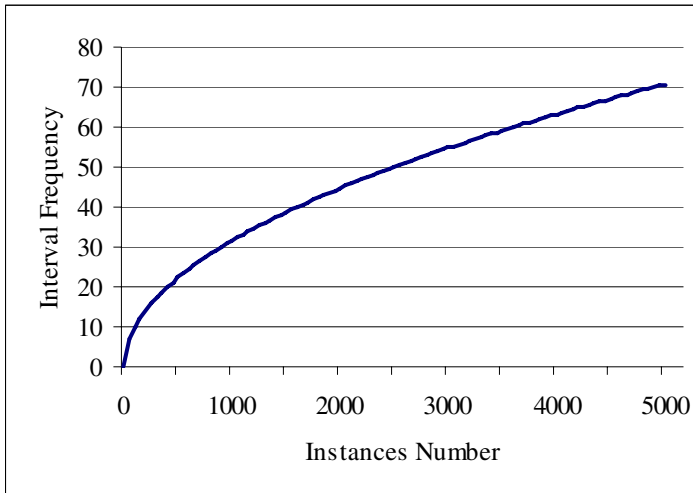


Fig. 2. Different sizes of training data require different ideal interval frequencies. Proportional discretization answers this call by setting both interval number and interval frequency to be the square root of the number of training instances. With instance number increasing, the interval frequency and number increase accordingly. When the instances number is less than 900, the ideal interval frequency should be less than 30 and when the instance number is greater than 3600, the ideal interval frequency should be greater than 60.

To compare IFFD against MFFD, according to Table 5, the difference between classification error rate of IFFD and that of MFFD's is not significant. When there are a small number of training instances, MFFD is better than IFFD. When more training instances are available, IFFD becomes better than MFFD. We suggest the reason is that the interval frequency of MFFD is 30 and is smaller than the interval frequency [30, 60) of IFFD. According to the interval frequency analysis in Fig 1, 30 is more suitable for small datasets.

To compare IFFD against KDE, according to Table 5, the difference between classification error rate of KDE and that of IFFD's is not significant. However, for some datasets, IFFD is dramatic better than KDE, as to be demonstrated in Section 6.2.2.

Table 6. Classification error win/draw/lose records on 30 datasets

Method	Win	Draw	Lose	Sign Test
IFFD & PiD	20	0	10	0.049
IFFD & MFFD	16	0	14	0.428
IFFD & KDE	19	0	11	0.1

6.2.2 Comparing on Every Dataset

For every dataset, if the classification error rate of a rival method is less than that of IFFD's at more than half of the 10 observation points, we deem that the rival method is better than IFFD for this dataset, and vice versa. The resulting win/draw/lose records across the 30 datasets are listed in Table 6. Accordingly, IFFD is significant better than PiD at the 0.05 critical level. Although not statistically significant, IFFD wins more often than not when compared with MFFD or KDE.

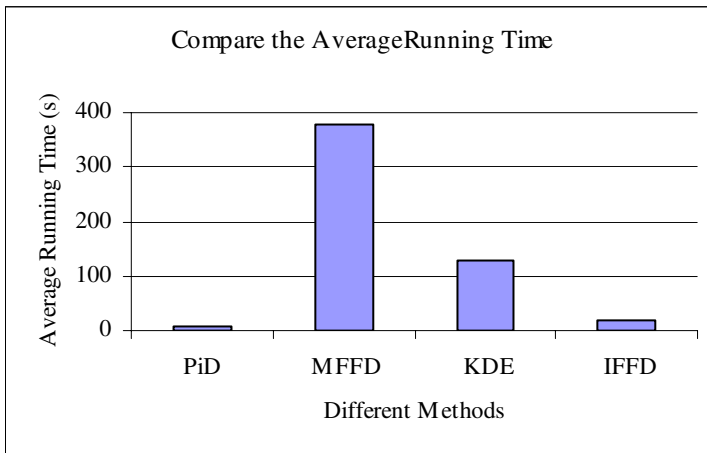


Fig. 3. NB's running time averaged on 30 datasets when coupled with PiD IFFD, KDE and MFFD respectively. PiD and IFFD are more efficient than KDE and MFFD.

6.2.3 Comparing Running Time

This section compares the running time of the four rival methods. Figure 3 demonstrates each method's running time averaged on the 30 datasets. From the fastest to slowest is PiD, IFFD, KDE and MFFD. It is consistent with our theoretical analysis in Section 5. PiD is the fastest algorithm. Although IFFD and MFFD have the same time complexity, for IFFD, it just modify one or two intervals and update the cutPoints, while for MFFD, on average it has to modify $\text{IntN} / 2$ intervals and associated statistics, where IntN is the interval number.

7 Conclusion

In this paper, we have argued that most existing discretization methods do not suit incremental learning of naïve-Bayes classifiers (NB). This is sub-optimal because NB is extensively deployed for real-world applications which often involve quantitative data. Accordingly, we have proposed a novel incremental discretization method *incremental flexible frequency discretization* (IFFD). IFFD inherits from fixed frequency discretization the strength of minimizing classification bias and variance for NB. Meanwhile, it adopts a more flexible strategy to handle to interval size so as to efficiently update discretized intervals upon receiving each new training instance. A comprehensive, theoretical and empirical study has been conducted to compare IFFD with representative alternative approaches. Observations suggest NB coupled with IFFD can achieve higher classification efficiency than those with MFFD and KDE, while achieve higher classification accuracy than those with PiD and KDE. Hence IFFD is a promising discretization approach for NB in practice where people want a rapport between learning accuracy and efficiency.

Acknowledgement

This research was supported by Australian Research Council grant DP0556279. We wish to thank João Gama and Carlos Pinto for providing the source code of PiD method.

References

1. C.L.Blake, & C.J.Merz,(1998). UCI repository of machine learning databases [<http://www.ics.uci.edu/mllearn/mlrepository.html>].
2. Peter Clark & Tim Niblett. (1989). The CN2 induction algorithm, *Machine Learning* 3(4), 261-283
3. Bojan CESTNIK. (1990). Estimating probabilities: A crucial task in machine learning. In *Proceedings of the 9th European Conference on Artificial Intelligence* (1990), pp. 147-149. (pp. 3, 23)
4. Richard O. Duda & Peter E. Hart. (1973). *Pattern classification and scene analysis*. New York: John Wiley and Sons.
5. João Gama & Gladys Castillo (2002): Adaptive Bayes. *Proceedings of the 8th Ibero-American Conference on AI: Advances in Artificial Intelligence: 765-774*

6. João Gama & Carlos Pinto .(2005) Discretization from Data Streams: Applications to Histograms and Data Mining Second International Workshop on Knowledge Discovery from data Streams
7. George H. John and Pat Langley (1995). *Estimating continuous distributions in Bayesian classifiers*. In Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, pages 338--345.
8. Pat Langley, Wayne Iba & Kevin Thompson. (1992). An analysis of Bayesian classifiers. Proceedings of the Tenth National Conference on Artificial Intelligence (pp. 223-228). San Jose, CA: AAAI Press.
9. Ying Yang & Geoffrey I. Webb (2001). Proportional kinterval discretization for naive-Bayes classifiers. 12th European Conference on Machine Learning (ECML'01) (pp. 564–575). Springer.
10. Ying Yang and Geoffrey I. Webb (2003). Discretization For Naive-Bayes Learning: Managing Discretization Bias And Variance. Technical Report 2003/131, School of Computer Science and Software Engineering, Monash University.
11. Ying Yang. Discretization for Naïve-Bayes Learning. PhD thesis, school of Computer Science and Software Engineering of Monash University.
12. Ying Yang & Geoff Webb (2003), On why discretization works for naïve-Bayes classifiers. In Proceedings of the 16th Australian Joint Conference on Artificial Intelligence (AI)
13. Neil A. Weiss. (2002). *Introductory Statistics, Sixth Edition*. Greg Tobin. (p. 98)

Distance Guided Classification with Gene Expression Programming*

Lei Duan, Changjie Tang, Tianqing Zhang, Dagang Wei, and Huan Zhang

School of Computer Science, Sichuan University,
Chengdu 610065, China
{duanlei, tangchangjie}@cs.scu.edu.cn

Abstract. Gene Expression Programming (GEP) aims at discovering essential rules hidden in observed data and expressing them mathematically. GEP has been proved to be a powerful tool for constructing efficient classifiers. Traditional GEP-classifiers ignore the distribution of samples, and hence decrease the efficiency and accuracy. The contributions of this paper include: (1) proposing two strategies of generating classification threshold dynamically, (2) designing a new approach called Distance Guided Evolution Algorithm (DGEA) to improve the efficiency of GEP, and (3) demonstrating the effectiveness of generating classification threshold dynamically and DGEA by extensive experiments. The results show that the new methods decrease the number of evolutionary generations by 83% to 90%, and increase the accuracy by 20% compared with the traditional approach.

1 Introduction

Mining classification rules from observed data set is helpful for classifying test samples [1]. Some traditional classification methods such as CART and C4.5 [2] can quickly generate rules that are relatively accurate and understandable. However, these algorithms perform local and greedy search and select only one attribute each time, and therefore the feature space is approximated by a set of hyper-cubes [3]. Another disadvantage is that the generated rules are often more complex than necessary [3].

Gene Expression Programming (GEP) [4] is a new evolutionary algorithm for data mining. It combines the advantages of both genetic algorithms (GAs) [5] and genetic programming (GP) [6], while overcoming some of their limitations. GEP performs global search and its genetic operators can modify many attributes at a time. Let $S(A_1, A_2, \dots, A_m, C)$ be the schema of a database, where A_i , $1 \leq i \leq m$, is a condition attribute, C is a class label. GEP tries to find a discriminant $f(A_j, \dots, A_k)$, $1 \leq j \leq k \leq m$, which can make instances of different classes have different values.

A generalized method for classification with GEP was proposed by C. Ferreira [7]. And the validities of this method for classifying non-linear and high dimensional

* This work was supported by the National Science Foundation of China under Grant No.60473071, the National Research Foundation for the Doctoral Program by the Chinese Ministry of Education under Grant No.20020610007 and the Software Innovation Project of Sichuan Youth under Grant No.2005AA0807.

samples were demonstrated. With a predetermined classification threshold, it discovers discriminants without any prior knowledge. However, it ignores the distribution of samples, so more runtime are consumed for GEP to discover the discriminant to fit for the predetermined threshold. As a result, the accuracy of result is lowered, and the result may be not compact.

In this study, we analyze the limitations of traditional GEP-classifiers; propose two strategies of generating classification threshold dynamically; design a new approach called Distance Guided Evolution Algorithm (DGEA) to improve the efficiency of GEP. The evaluations show that, for the same problem, the proposed strategies can decrease the number of generations by 83%, and DGEA can decrease it by nearly 90% compared with the traditional GEP-classifier. Besides, the experiments show that classification with GEP is desirable in non-real time environment.

The rest of this paper is organized as follows. Section 2 analyzes the limitations of traditional GEP-classifiers. Section 3 details the two strategies of generating classification threshold dynamically. Section 4 describes the design of DGEA. Section 5 provides experimental evaluations. Finally Section 6 concludes this paper.

2 Limitations of Traditional GEP-Classifiers

GEP is a new concept proposed by C. Ferreira based on GAs and GP in 2001 [4]. It avoids the limitations of losing in functional complexity (the case of GAs) and reproducing with modification difficultly (the case of GP). Indeed, GEP is extremely versatile and greatly surpasses the existing evolutionary techniques [7, 8, and 9].

Based on the principle of natural selection, GEP operates iteratively evolving a population of chromosomes, encoding candidate solutions, through several genetic operators. GEP offers great potentiality to solve complex modeling and optimization problems [3]. C. Ferreira [7] and Chi Zhou [3, 10] have demonstrated that the prediction accuracy obtained by GEP is higher than those obtained by C4.5 and C4.5Rules. Moreover, GEP is more efficient compared with traditional tree-based GP methods, and the classification rules discovered by GEP are more compact.

C. Ferreira proposed a generalized method for classification with GEP in [7]. In a *two-class* problem, GEP discovers a discriminant $gop(X)$, where X is the input condition attributes. t is a predefined threshold. If X belongs to a positive sample $gop(X) - t > 0$, otherwise $gop(X) - t < 0$. In this case, one discriminant discovered by GEP is sufficient to predict whether a given sample belongs to that class or not.

The authors of [7, 10] adopted *one-against-all* learning method to transform one n -class problem into n *two-class* problems when solving n -class classification problems. In this case, the classification rule is as follows:

Rule 1. IF $gop(R_i) > threshold$ THEN $R_i \in Class k$ ELSE $R_i \notin Class k$. where, $gop()$ is the discriminant discovered by GEP, R_i are attribute values of a given sample.

Experiments did by C. Ferreira show that GEP is suitable for solving classification problems [7]. However, the values of *threshold* are predetermined subjectively. Chi Zhou constructed a GEP-classifier in a slightly different way [3, 10]. The threshold was defined as 0. Then the discriminant performs classification by returning a positive or nonpositive value indicating whether or not a given instance belongs to that class.

According to Rule 1, GEP has to discover a discriminant to perform classification based on the predetermined threshold. Usually, there is no prior knowledge about the distribution of samples. Thus the threshold is determined empirically. In this case, more runtime will be consumed for GEP to explore the discriminant fit for the threshold. Furthermore, the result may be not compact.

3 Computing Classification Threshold Dynamically

The analyses above show that: (a) GEP is suitable for solving classification problems; (b) the proposed classification methods with GEP need a predefined threshold, and discover a discriminant without considering the distribution of samples. Accordingly, the efficiency of GEP is lowered and the result is not compact. To solve this problem, we propose a dynamical threshold method for classification.

Definition 1 (GEP Feature Value). Let S be a training set with each sample taking the form (a_1, a_2, \dots, a_m) , where a_i ($i = 1, 2, \dots, m$) is in the domain of attribute A_i and associated with a unique target class label, $gep(\cdot)$ be the discriminant discovered by GEP and (x_1, x_2, \dots, x_m) be a sample in S , then $gep(x_j, \dots, x_k)$, $1 \leq j \leq k \leq m$, is called GEP Feature Value. $\{gep(x_j, \dots, x_k) | (x_j, \dots, x_k) \in S\}$ is called GEP Feature Dimension.

Due to the powerful evolutionary search mechanisms, GEP is more efficient than traditional algorithms to solve numeric problems. Two ways for performing numerization of nominal attributes in GEP were proposed in [10]. In this study, we mainly focus on applying GEP to the *two-class* problems with numeric attributes.

We propose two strategies to describe the distribution of GEP Feature Values of the same class in GEP Feature Dimension, which are enlightened by two typical partition methods known as *k-means* and *k-medoids* [1].

Definition 2 (The Mean of GEP Feature Values). Let $G = \{g_i | i=1, 2, \dots, n\}$ be the set of GEP Feature Values that each element of G belongs to the same class c , then the mean of G is called the Mean of GEP Feature Values of class c , denoted as $Mean_c$, i.e., $Mean_c = (g_1 + g_2 + \dots + g_n)/n$.

Definition 3 (The Median of GEP Feature Values). Let $G' = \{g'_i | i=1, 2, \dots, n\}$ be the set of GEP Feature Values sorted by ascending or descending order and each element of G' belongs to class c , then the median of G' is called the Median of GEP Feature Values of class c , denoted as $Median_c$, i.e., $Median_c = g'_{(n/2+1)}$.

As stated in [11], given n samples the time complexity of calculating the mean or the median is $O(n)$. Based on $Mean_c$ or $Median_c$, classification threshold can be computed dynamically. Let R_p and R_N be $Mean_p$ ($Median_p$) and $Mean_N$ ($Median_N$) in a *two-class* problem, t be the classification threshold. The midpoint between R_p and R_N can be used as the classification threshold, if there is not any prior knowledge.

$$t = (R_p + R_N) / 2 \quad (1)$$

Each test instance is classified according to the relative position relation of its GEP Feature Value and the classification threshold. For example, the given instance is classified to the class whose $Mean_c$ ($Median_c$) is closer to its GEP Feature Value.

Definition 4. Let $gcp()$ be a discriminant discovered by GEP, pos and neg be the number of GEP Feature Values classified correctly for class P and class N respectively, N_P and N_N be the number of samples of class P and class N respectively, then the training score of $gcp()$ is defined as $TScore = (pos + neg)/(N_P + N_N)*100\%$.

The value of $TScore$ reflects the classification accuracy of the discriminant discovered by GEP. Discovering the discriminant with the highest training score by GEP is the purpose of our study.

Based on the preceding analyses, we propose two classification algorithms with GEP: GEP-Mean and GEP-Median. The main difference between them is that the former uses $Mean_c$ to compute the threshold, while the latter uses $Median_c$.

4 Evolution Acceleration

As GAs and GP, the fitness function is important for the efficiency of GEP. Traditional fitness function measures the fitness of $gcp()$ according to the number of training samples classified correctly. The fitness function has the form:

$$fitness = (pos + neg) * I \tag{2}$$

where I is the increase in fitness when an instance is classified correctly.

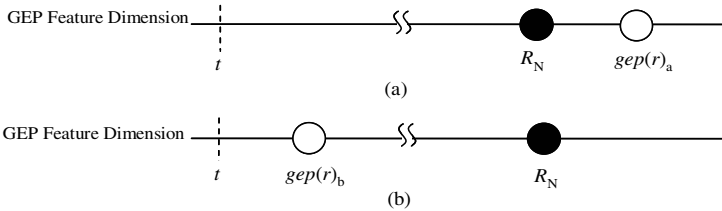


Fig. 1. The expected area of class P is on the left of t . However, the GEP Feature Value of r which belongs to class P is on the right of t . R_N is the $Mean_N$ ($Median_N$). (a) for discriminant $gcp()_a$, the GEP Feature Value of r is on right of R_N . (b) for discriminant $gcp()_b$, the GEP Feature Value of r is on the left of R_N .

C. Ferreira adopted Equation (2) to measure the classification effect of $gcp()$ in [7]. However, this method is so simple that each training instance has just one state: be classified correctly or incorrectly. Figure 1 displays the complexity in the situation of incorrect classification. In order to be observed easily, we indicate the threshold with a vertical line t , the $Mean_c$ or $Median_c$ with a black circle, the GEP Feature Value of a certain instance with a white circle.

Suppose all the GEP Feature Values of sample set computed by $gcp()_a$ and $gcp()_b$ are in corresponding expected areas except sample r , then the fitness of $gcp()_a$ and $gcp()_b$ are the same when measured by Equation (2). As shown in Figure 1, $gcp(r)_b$ is closer to the expected area than $gcp(r)_a$. It means that $gcp()_b$ is more prone to generate expected solution in its offspring than $gcp()_a$.

Observation 1. The traditional fitness function ignores the distribution of the GEP Feature Values of samples which are classified incorrectly.

We propose an algorithm called Distance Guided Evolution Algorithm (DGEA) to overcome the limitation stated in Observation 1. DGEA considers the fitness change when a sample is classified incorrectly. Algorithm 1 describes the design of DGEA.

Given a sample r classified incorrectly, let D_R be the Manhattan distance between the $Mean_P$ ($Median_P$) and $Mean_N$ ($Median_N$), D_E be the Manhattan distance between the GEP Feature Value of r and the Mean (Median) of GEP Feature Values of unexpected class. If D_E is less than D_R , the fitness increase is:

$$fitness += D_E / D_R * I \quad (3)$$

Adopting Equation (3) to calculate the fitness of the discriminant not only computes the fitness increase when a sample is classified incorrectly, but also describes how far the GEP Feature Value to the expected area. Assigning higher fitness to the discriminant which is prone to evolve to be an expected solution in its offspring can accelerate the process of discovering the optimal solution.

Algorithm 1. DGEA(TSP, TSN, Rp, Rn)

Input: (1) two training subsets whose elements belong to class P and class N: TSP and TSN;

(2) $Mean_P$ ($Median_P$) and $Mean_N$ ($Median_N$): Rp and Rn.

Output: the fitness of classification discriminant: fitness.

```

begin
  1. Rb ← abs(Rp - Rn)
  2. for each sample[i]
  3.   do toRp ← abs(sample[i] - Rp)
  4.     toRn ← abs(sample[i] - Rn)
  5.     if (toRp < toRn) and (sample[i] ∈ TSP)
  6.       then fitness ← fitness + I
  7.     else if (toRp < Rb) and (sample[i] ∈ TSP)
  8.       then fitness ← fitness + toRn/Rb *
I
  9.     if (toRn < toRp) and (sample[i] ∈ TSN)
 10.      then fitness ← fitness + I
 11.    else if (toRn < Rb) and (sample[i] ∈ TSN)
 12.      then fitness ← fitness + toRp/Rb *
I
 13. return fitness
end.
```

Note that the values of I in Equation (2) and (3) can be assigned any positive integer. In line 8 and 12, DGEA computes the fitness increase when a sample is classified incorrectly. This is the main difference between DGEA and the traditional algorithm. The time complexity of DGEA is the same as the traditional algorithm.

In Algorithm 1, the values of Rp and Rn have two selections. In order to avoid confusion, it is called DGEA-Mean when Rp (Rn) is $Mean_P$ ($Mean_N$). And it is called DGEA-Median when Rp (Rn) is $Median_P$ ($Median_N$).

5 Performance Evaluation

We implemented five algorithms, namely GEP-Mean, GEP-Median, DGEA-Mean, DGEA-Median, and the traditional GEP-classifier (GEP-TA) [7], using Java based on the platform JDK 1.4.2. Besides, we compared the results of above algorithms with some traditional classification algorithms, such as NaïveBayesUpdateable (NBU), J48 and Sequential Minimal Optimization Algorithm (SMO) available on the Weka collection of machine learning algorithms [12]. All experiments were conducted on an Athlon XP 1800+ CPU with 256M memory running Windows 2000.

First, we demonstrate the effectiveness and efficiency of our proposed algorithms from the aspect of the number of generations.

We applied algorithms to Iris dataset [13]. Table 1 shows that classification with GEP can acquire higher accuracy than other classification algorithms. Thus, it is desirable to apply GEP to classification.

Table 1. Accuracy comparison among typical classification algorithms for Iris dataset

	NBU	J48	SMO	GEP
<i>Setosa vs. NOT setosa</i>	100%	100%	100%	100%
<i>versicolor vs. NOT versicolor</i>	94%	98%	67.3333%	99.3333%
<i>virginica vs. NOT virginica</i>	92%	98%	96.6667%	99.3333%

We ran GEP-TA, GEP-Mean, GEP-Median, DGEA-Mean, and DGEA-Median until the *TScore* of the discovered discriminant equaled to the accuracy displayed in Table 1. The experiments were repeated 10 times and the results were averaged. Each algorithm correctly classified all samples to *setosa* and NOT *setosa* in 5 generations, since the characteristics of *setosa* are obvious.

As shown in Figure 2, GEP-Mean and GEP-Median decrease the number of generations by 24.1% and 13.8% when classifying *versicolor*, 83.6% and 70.1% when classifying *virginica* compared with GEP-TA. DGEA-Mean and DGEA-Median can farther decrease the number of generations. The former decrease it by 58.9% and the latter decrease it by 56.0% when classifying *versicolor*. And the former decrease it by 89.9% and the latter decrease it by 87.3% when classifying *virginica*.

Second, we demonstrate the effectiveness and efficiency of our proposed algorithms from the aspect of the accuracy of the results.

We applied the proposed algorithms to the other dataset to demonstrate that solutions with higher accuracy, compared with the traditional approach, can be discovered. The dataset was downloaded from <http://lib.stat.cmu.edu/DASL/Datafiles/FleaBeetles.html>, which contains samples of three types of flea beetle: *concinna* (*Con*), *heikertingeri* (*Hei*) and *heptapotamica* (*Hep*). We performed 3-fold cross-validation and split the dataset into two parts: one was the training dataset; the other part was reserved for testing purposes. The ratio between the training and testing set was 2:1. The experiments were repeated 10 times and the results were averaged. For each algorithm, experiments were conducted in 6 groups according to the number of generations. The results show that the accuracies obtained by proposed algorithms are

all above 90% when classifying *Con* and *Hei*. However, the accuracy of GEP-TA is below 85%. The classification accuracies of all the algorithms are greater than 97% when classifying *Hep*. Furthermore, the runtime of these five algorithms are nearly equal as the time complexities of them are the same. For space limitations, the details of results are omitted, and readers can refer to [14].

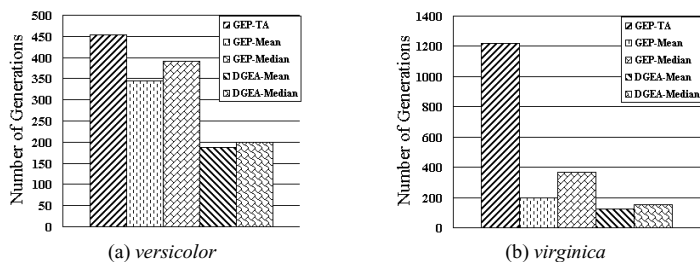


Fig. 2. The number of generations to discover the discriminants of the same *TScore*

In addition, we ran DGEA-Mean and DGEA-Median 10 times and stopped them after 5000 ms each time. NBU, J48 and SMO were also conducted in the same training set and test set. Table 2 shows that the accuracies of DGEA-Mean and DGEA-Median are higher than those of J48 in any case. NBU and SMO can obtain results with slightly higher accuracies than DGEA-Mean and DGEA-Median when classifying *Hei*. However, DGEA-Mean and DGEA-Median can obtain results with higher accuracies when classifying the other two cases.

Table 2. Accuracies of typical classification algorithms for Flea Beetle dataset

	NBU	J48	SMO	DGEA-Mean	DGEA-Median
<i>Con</i> vs. <i>NOT Con</i>	97.2222%	90.3846%	95.8333%	96.0444%	98.7179%
<i>Hei</i> vs. <i>NOT Hei</i>	98.6111%	94.6581%	98.6111%	97.3290%	97.3290%
<i>Hep</i> vs. <i>NOT Hep</i>	95.9402%	97.3290%	97.3290%	100%	100%

6 Conclusions

Applying GEP to classification can perform global search and discover the discriminant with high accuracy. However, the traditional GEP-classifier discovers a discriminant without considering the distribution of samples. To overcome this limitation, we propose two effective strategies of generating threshold dynamically, and design the Distance Guided Evolution Algorithm to accelerate the process of problem solving.

The future work includes implementing the DGEA approach into constraint-based classification and evaluating its performance with real measurements.

References

1. Jiawei Han, Micheline Kambr: Data Mining Concepts and Techniques. Higher Education Press, Beijing (2001) 185–235
2. J. Ross Quinlan: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
3. Chi Zhou, Peter C. Nelson, Weimin Xiao, and Thomas M. Tirpak: Discovery of Classification Rules by Using Gene Expression Programming. Proceedings of the International Conference on Artificial Intelligence, Las Vegas, USA (2002) 1355–1361
4. C. Ferreira: Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems. Vol. 13, 2(2001) 87–129
5. D.E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)
6. J.R. Koza: Genetic Programming. Cambridge, MA: MIT Press (1992)
7. C. Ferreira: Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence. Angra do Heroismo, Portugal (2002)
8. C. Ferreira: Discovery of the Boolean Functions to the Best Density-Classification Rules Using Gene Expression Programming. Proceedings of the 4th European Conference on Genetic Programming (EuroGP 2002), Lecture Notes in Computer Science, Vol. 2278. Springer-Verlag, Berlin Heidelberg New York (2002) 51–60
9. C. Ferreira: Mutation, Transposition, and recombination: An analysis of the evolutionary Dynamics. Proceedings of the 4th International Workshop on Frontiers in Evolutionary Algorithms, Research Triangle Park, North Carolina, USA (2002) 614–617
10. Chi Zhou, Weimin Xiao, Thomas M. Tirpak and Peter C. Nelson: Evolution Accurate and Compact Classification Rules With Gene Expression Programming. IEEE Transactions on Evolutionary Computation. Vol. 7, 6 (2003) 519–531
11. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms. Higher Education Press, Beijing (2002) 184–189
12. Ian H. Witten and Eibe Frank: Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco (2005)
13. C. L. Blake and C. J. Merz. UCI repository of machine learning databases (2000)
14. <http://www.scu.edu.cn/waim03/buf/download/DGEA.pdf>

Research on Multi-valued and Multi-labeled Decision Trees

Hong Li¹, Rui Zhao¹, Jianer Chen², and Yao Xiang¹

¹ School of Information Science and Engineering, Central South University,
Changsha 410083, China

² Department of Computer Science, Texas A&M University
Texas 77843-3112, USA

lihongcsu@mail.csu.edu.cn, zhaorui000@163.com,
chen@cs.tamu.edu, xysy_211@sina.com

Abstract. Ordinary decision tree classifiers are used to classify data with single-valued attributes and single-class labels. This paper develops a new decision tree classifier SSC for multi-valued and multi-labeled data, on the basis of the algorithm MMDT, improves on the core formula for measuring the similarity of label-sets, which is the essential index in determining the goodness of splitting attributes, and proposes a new approach of measuring similarity considering both same and consistent features of label-sets, and together with a dynamic approach of adjusting the calculation proportion of the two features according to current data set. SSC makes the similarity of label-sets measured more comprehensive and accurate. The empirical results prove that SSC indeed improves the accuracy of MMDT, and has better classification efficiency.

1 Introduction

In all classification approaches, decision tree classifier is probably the most popular and the most widely used[1]. Most classifiers are incapable of handling data with multi-valued attributes and multiple labels[2], which has been proved in Chen, Y. (2003) [3]. Therefore, it is necessary to propose decision tree-based algorithms to process multi-valued and multi-labeled data in practices. The reported algorithm MMDT(multi-valued and multi-labeled decision tree)[4] uses the scoring of similarity and appropriateness of label-sets of child nodes to determine the goodness of attributes, and the scoring approach ascribes to the measuring of similarity of label-sets which is the essential index. In MMDT, the measuring of similarity starts from one single side of the same feature or consistent feature of label-sets separately, neglecting the comprehensive contributions of same and consistent feature to the similarity of label-sets. This paper proposes a new approach of measuring similarity considering not only the same feature but also the consistent feature of label-sets, in which the calculation proportion of the two features is adjusted dynamically, and develops a new decision tree classifier SSC (Similarity of Same and Consistent) for multi-valued and multi-labeled data, which is of better classification efficiency.

2 Problem Description

2.1 Multi-valued and Multi-labeled Data

Multi-valued attribute means that the value of attribute is a set of values rather than a single field. Multi-labeled data means that a record of data may belong to several classes, that is, with several class labels as classification[3]. Table 1 shows a multi-valued and multi-labeled data of 3 records. In the table, the attribute of gender and hobby are categorical; age is numerical; hobby is multi-valued; and the classification attribute of publication type is multi-labeled, that is, each record is associated with at most three different labels C_1 , C_2 and C_3 .

2.2 Decision Tree Classifier for Multi-valued and Multi-labeled Data

Assume a training data set D , with n records, and $C=\{C_1, \dots, C_i, \dots\}$ is the set of all class labels for choice, called class set. We use a label-set which is a set of labels in C , marked S_j , to represent the classification group of a record. Besides, we mark all attributes as $A=\{A_1, \dots, A_i, \dots\}$, which can be single-valued or multi-valued, numerical or categorical. The decision tree $T(N, B)$, N is a set of internal or leaf nodes, and B is a set of branches. Each internal node corresponds to a decision on an attribute's value, each branch corresponds to a possible value of attribute when attribute is categorical or value interval when numerical, and each leaf node corresponds to a label-set as the predicted classification group. We use a user-specified parameter ub to set an upper bound on the number of branches which are fanned out from a numerical attribute.

3 The SSC Algorithm

This paper develops a new decision tree classifier SSC for multi-valued and multi-labeled data, and proposes a new approach of measuring similarity considering both same and consistent feature of label-sets, and also a dynamic method to adjust the calculation proportion, by which the similarity of label-sets can be measured more accurate. We illustrate SSC as follows: (1) conditions for stop nodes, (2) how to select the best splitting attribute, (3) how to determine prediction accuracy.

3.1 Conditions for Stop Nodes

3.1.1 Notation Definition

Def 1. Suppose a current node CN of a decision tree, with D_{CN} as its data subset, and S_j as the label-sets of D_{CN} , the definition for label support marked $Sup(C_i)$ is as follows:

$$Sup(C_i) = \frac{\text{number of records containing } C_i \text{ in } D_{CN}}{\text{number of records in } D_{CN}}$$

Def 2. Sup_{\min} is the user-specified minimum support, as the division of weighty labels and light labels. If $Sup(C_i)$ is not less than Sup_{\min} , C_i is a weighty label, or else a light label.

Def 3. Define the difference of a node as the smallest support of the set of weighty labels minus the largest support of the set of light labels, and define $Diff_{min}$ as the user-specified minimum difference of a node.

Def 4. Num_{min} is the user-specified minimum number of records in a data subset.

3.1.2 Three Conditions for Stop Nodes

Condition 1. If the difference of CN is not less than $Diff_{min}$, CN will stop growing, named a stop node, with all the weighty labels of D_{CN} as its label-set[4, 5].

Condition 2. Given all attributes have been chosen in the current path from root node down to the current node CN, CN is a stop node, with all weighty labels as label-set when any weighty label exists, or else the label with the largest support.

Condition 3. When the number of records in D_{CN} is less than Num_{min} , CN is a stop node, and the evaluating of label-set takes the same method in Condition 2.

If any of the above conditions is met, the current node CN stops growing, or else turn to the step of selecting the best splitting attribute.

Table 1. A multi-valued and multi-labeled training set

id	Gender	Age	Hobby	Publication type
1	Female	24	Traveling, watching TV	C_2, C_3
2	Female	50	Watching TV	C_1
3	Male	51	Watching TV	C_1, C_2

3.2 How to Select the Best Splitting Attribute

The information gain measure has been demonstrated not suitable for multi-valued and multi-labeled data[3]. Take the scoring of similarity and appropriateness of label-sets [3, 5] as the measure of goodness of attributes, attempting to make the label-sets of child nodes data with the greatest similarity after splitting, so that it is easy to find the most appropriate label-sets to represent the child nodes.

3.2.1 Measuring of Similarity

The measure of similarity is ascribed to the evaluation of the similar degree of two sets. A former classification algorithm MMC(multi-valued and multi-labeled classifier)[3] defined the similarity between two label-sets S_i and S_j as formula (1).

$$sim(S_i, S_j) = \frac{1}{2} \left(\frac{same(S_i, S_j)}{cardinality(S_i, S_j)} - \frac{different(S_i, S_j)}{cardinality(S_i, S_j)} + 1 \right) \tag{1}$$

where $same(S_i, S_j)$ stands for the number of labels that appear in both S_i and S_j , $cardinality(S_i, S_j)$ stands for the number of different labels that appear in S_i or S_j , and $different(S_i, S_j)$ stands for the number of labels that appear either in S_i or S_j .

The right part of formula (1) considers the rate of same elements and different elements in S_i and S_j , called the same degree of two sets. However the consistent degree

can reflect the similarity on the other hand, which means the rate of consistent act and inconsistent act that all elements for choice have in the two sets. Based on the consistent degree, MMDT proposed another similarity measuring as formula (2).

$$sim(S_i, S_j) = unitary \left(\frac{1 + consistent(S_i, S_j)}{1 + inconsistent(S_i, S_j)} \right) \tag{2}$$

In this formula, *unitary()* is a normalization function to make *sim(S_i, S_j)* a pure decimal, *consistent(S_i, S_j)* is the sum of the number of all labels in class set C that appear in both S_i and S_j, and also not appear in both S_i and S_j; corresponding *inconsistent(S_i, S_j)* is that appear only in S_i and only in S_j [3].

The measuring approach of similarity in MMC or in MMDT comes from only one side. For instance, S₁={C₁} and S₂={C₂}, from formula (1) their similarity is zero, but clearly they have resemblance that C₃ does not appear in both S₁ and S₂. So it is not exact to evaluate by formula (1). Nevertheless using formula (2) also has disadvantage. Given C={C₁,C₂,C₃}, S₁={C₁}, S₄={C₁,C₂}, S₇={C₁,C₂,C₃}, based on formula (2) *sim(S₄, S₁) = sim(S₄, S₇)*, but it is clearly that S₄ and S₁ have only one same class label, while S₄ and S₇ with two. Here we propose a new approach to measure similarity considering both same and consistent features of label-sets, which is the basic strategy in SSC, shown in formula (3). In it *occurrence(S_i, S_j)* is the sum of number of acts that each class label in C may have in a label-set. The act of class label is the act of appearing or not appearing in the label-set, and each class label only has a act in a label-set, so *occurrence(S_i, S_j)* equals to the number of class labels in C.

$$similarity(S_i, S_j) = (1 - \alpha) \frac{same(S_i, S_j)}{cardinality(S_i, S_j)} + \alpha \frac{consistent(S_i, S_j)}{occurrence(S_i, S_j)} \tag{3}$$

The former item in the right of formula (3) represents the same feature, and the latter represents the consistent feature. Set a parameter α as an adjustment for the calculating proportion of two features, α is adjusted according to the feature of current data, and each node should evaluate α according to current data when building tree.

Here give a measuring of α , using the ratio of odd points in same and consistent features. Given C={C₁,C₂,C₃}, via formula (1) and (2) we get two similarity lists of C based on same feature and consistent feature, illustrated in Table 2 and Table 3. Comparing the two lists, it is easy to find some label-sets have zero as similarity value in Table 2, but corresponding value is nonzero in Table 3, which indicates their same degree is zero, but the consistent degree is nonzero. Point (S₁, S₂) with zero as the same-based similarity value, and 0.11 as the consistent-based value, points like this are odd points in same and consistent features. The parameter α is evaluated by the ratio of odd points that appear in the combination set of any two label-sets in the current label-sets, shown in formula (4), N = (n₁, ..., n_i, ..., n_t), n_i is the number of records with S_i as label-sets, and t is the number of label-sets in C. A_{ij} is an odd points matrix, when (S_i, S_j) is an odd point, its value a_{ij} equals to 1, or else 0.

$$\alpha = N \cdot A_{ij} \cdot N^T / C_n^2 \tag{4}$$

Formula (3) has adequately considered the same and consistent features of two label-sets, and also embodies the relationship of label-sets and class set, so by formula (3) the similarity of label-sets can be measured more comprehensive and accurate.

Table 2. Same-based similarity list of class set {C₁,C₂,C₃}

Similarity	C ₁	C ₂	C ₃	C ₁ ,C ₂	C ₁ ,C ₃	C ₂ ,C ₃	C ₁ ,C ₂ ,C ₃
S ₁ C ₁	1	0	0	0.5	0.5	0	0.333
S ₂ C ₂	0	1	0	0.5	0	0.5	0.333
S ₃ C ₃	0	0	1	0	0.5	0.5	0.333
S ₄ C ₁ ,C ₂	0.5	0.5	0	1	0.333	0.333	0.667
S ₅ C ₁ ,C ₃	0.5	0	0.5	0.333	1	0.333	0.667
S ₆ C ₂ ,C ₃	0	0.5	0.5	0.333	0.333	1	0.667
S ₇ C ₁ ,C ₂ ,C ₃	0.333	0.333	0.333	0.667	0.667	0.667	1

Table 3. Consistent-based similarity list of class set {C₁,C₂,C₃}

Similarity	C ₁	C ₂	C ₃	C ₁ ,C ₂	C ₁ ,C ₃	C ₂ ,C ₃	C ₁ ,C ₂ ,C ₃
S ₁ C ₁	1	0.11	0.11	0.33	0.33	0	0.11
S ₂ C ₂	0.11	1	0.11	0.33	0	0.33	0.11
S ₃ C ₃	0.11	0.11	1	0	0.33	0.33	0.11
S ₄ C ₁ ,C ₂	0.33	0.33	0	1	0.11	0.11	0.33
S ₅ C ₁ ,C ₃	0.33	0	0.33	0.11	1	0.11	0.33
S ₆ C ₂ ,C ₃	0	0.33	0.33	0.11	0.11	1	0.33
S ₇ C ₁ ,C ₂ ,C ₃	0.11	0.11	0.11	0.33	0.33	0.33	1

3.2.2 Measuring of the Best Attribute

(1) Similarity of a set of label-sets

Based on the measuring of similarity of two label-sets, we can reduce the similarity among the set of label-sets of a node, such as S={S₁,S₂,...,S_n}, in which n is the number of records, and the similarity measuring of the node is as formula (5).

$$SIMILARITY(S) = \frac{\sum_{i < j} similarity(S_i, S_j)}{C_n^2} \tag{5}$$

(2) Profit similarity of attribute

Suppose selecting the best attribute on CN, with data set D_{CN} (n records), {a₁,a₂,...,a_m} stands for value set of A_i with m different values when A_i is categorical, and when numerical value domain of A_i is partitioned into at most ub intervals. Then according to A_i, D_{CN} is partitioned into k intervals which equals to m when categorical or ub when numerical, and each interval corresponds to a child node of A_i. Let n₁,n₂,...,n_k denote the record number of child nodes, and let n'=∑_{i=1}^k n_i, where n' ≥ n. Define the similarity gain by the splitting of A_i as the profit similarity of A_i [4], denoted as Profit_similarity(A_i) in formula (6), which is one index of the scoring measure, and select the attribute with the largest profit similarity value as splitting attribute.

$$Profit_similarity(A_i) = E_similarity(A_i) - SIMILARITY(CN)$$

$$E_similarity(A_i) = \sum_{i=1}^k \frac{n_i}{n'} SIMILARITY(i) \tag{6}$$

(3) Profit appropriateness of attribute

The appropriateness of a label-set means the appropriate degree of representing a data set using this label-set, and the appropriateness of a node is the max value among the appropriateness values of representing the data set of this node using all label-sets for choice[4]. For CN, the appropriateness of label-set S_i is evaluated by the similar degree between S_i and each label-set in D_{CN} , shown in formula (7), where sim_{ij} is the similarity of label-set S_i and S_j . Further by formula (8) we get the appropriateness of node CN. And then define the appropriateness gain by the splitting of A_i as the profit appropriateness of A_i , denoted as $Profit_appropriateness(A_i)$, shown in formula (9).

$$appropriateness(S_j) = \frac{N \cdot SS_j}{n}, N = (n_1, \dots, n_i, \dots, n_t)$$

$$SS_j = (sim_{j1}, \dots, sim_{ji}, \dots, sim_{jt}) \quad (i = 1, \dots, t) \tag{7}$$

$$APP(CN) = \overset{t}{Max}_{j=1} appropriateness(S_j) \tag{8}$$

$$Profit_appropriateness(A_i) = E_appropriateness(A_i) - APP(CN)$$

$$E_appropriateness(A_i) = \sum_{j=1}^k \frac{n_j}{n'} APP(j) \tag{9}$$

where $APP(j)$ denotes the appropriateness of the j_{th} child node of A_i , and n_j is the number of records of this child node's data set, $n' = \sum_{j=1}^k n_j$.

(4) Profit of attribute

Use the scoring of profit similarity and profit appropriateness to evaluate the profit of attributes, as the final index of determining the goodness of attributes, shown in formula (10). Choose the attribute with the largest profit as the splitting attribute.

$$PROFIT(A_i) = Profit_similarity(A_i) + Profit_appropriateness(A_i) \tag{10}$$

3.3 Determining Prediction Accuracy

The test phase of a decision tree classifier involves how to determine the predicted label-set for a new data record, and how to determine the accuracy of a prediction. Firstly traverse the tree constructed from the root to a leaf node it reaches, and then set the union of all label-sets of the leaf nodes reached as the predicted result of the record. And then take the similarity of two label-sets to compare it with the real label-set of the record to determine the accuracy of this prediction[1].

4 Performance Evaluation

We generate the synthetic data by modifying the famous synthetic data in traditional algorithms [2,7,9]. The data is synthetic customers' data, in which there are four attributes of customers' information, given in Table 4. In the data, attribute hobby and car is multi-valued, others are single-valued, age and salary is numerical, others are categorical, and all attribute values are randomly generated in certain domains. We developed a series of classification rules by using the above attributes to classify customers into five different classes $C_1 \sim C_5$, and the rules defined in Table 5. We generate all training data and testing data of our experiments by the method above.

Table 4. Description of experimental data

Attribute	Type	Number of value	Domain of value
Age	Numerical, single-valued	1	integer in[20,80]
Car	Categorical, multi-valued	From 1 to 2	integer in[1,8]
Education	Categorical, single-valued	1	integer in[1,3]
Gender	Categorical, single-valued	1	0, 1
Hobby	Categorical, multi-valued	From 1 to 3	integer in[1,8]
Job	Categorical, single-valued	1	integer in[1,10]
Marry	Categorical, single-valued	1	integer in[1,3]
Salary	Numerical, single-valued	1	[20000,150000]

Table 5. Rules of classification according to attributes

Class	Rules
C_1	[Gender=0 \wedge salary in [20000,100000] \wedge car in [1,4]] \vee [Gender=1 \wedge salary in [120000,150000] \wedge car in [5,8]] \vee Job=1,6
C_2	[Age in [20,40] \wedge car in [1,4]] \vee [Age in [40,60] \wedge car in [5,6]] \vee Job=2
C_3	[Marry=1 \wedge salary in [20000,100000] \wedge (hobby in [1,3] \vee car=4)] \vee [Marry=2 \wedge salary in [100000,150000] \wedge (hobby in [5,7] \vee car=8)] \vee Job in [3,5]
C_4	[Marry=1 \wedge salary in [20000,40000]] \vee [Marry=2 \wedge salary in [40000,80000]] \vee Job in [7,8]
C_5	[Education=1 \wedge hobby in [1,3] \wedge Marry=1] \vee [Education=3 \wedge hobby in [4,8] \wedge Marry=2,3] \vee Job in [9,10]

Use different size of data sets to take experiments on SSC, same-based MMDT_1 and consistent-based MMDT_2. We set $ub=4$, $Sup_{min}=50\%$, $Diff_{min}=15\%$, $Num_{min}=2$, and α is generated dynamically according to data set of each splitting node when building tree, and according to the whole test data when predicting. We respectively use 2000, 4000 and 6000 records of training set to build trees and corresponding records of testing set for prediction. Table 6 shows the comparison of SSC, MMDT_1 and MMDT_2 in prediction accuracy, where it is clear SSC has higher accuracies than MMDT for all size of data, which signifies SSC has better classification efficiency, and handles multi-valued and multi-labeled data very well.

Table 6. Comparison of prediction accuracy of SSC, MMDT_1 and MMDT_2

Algorithm	Training set/testing set of 2000	Training set/testing set of 4000	Training set/testing set of 6000
SSC	67.49%	66.94%	66.87%
MMDT_1	62.011%	62.58%	62.89%
MMDT_2	62.51%	63.24%	63.20%

5 Conclusion

The study of decision tree classifiers for multi-valued and multi-labeled data is still underway. This paper improves on the core formula for measuring the similarity of label-sets, combining the same feature and consistent feature of label-sets as a composite measuring strategy, proposes a dynamic approach of adjusting the calculation proportion of the two features, and develops a new decision tree classifier SSC for multi-valued and multi-labeled data. By some comparative experiments SSC is improved of better accuracy than MMDT. However the accuracy of prediction and the concision of constructed tree are still not perfect, and there will be some improvements by further study on the measuring approaches of label-sets similarity.

References

1. Han, J., & Kamber, M. *Data Mining Concept and Technology*. [M]. Peking: China Machine Press, 2001.
2. Shafer, J. C., Agrawal, R., & Mehta, M. (1996). *SPRINT: A scalable parallel classifier for data mining*. Proceedings of the 22nd International Conference on Very Large Databases (pp. 544–555). Mumbai (Bombay), India.
3. Chen, Y., Hsu, C., & Chou, S. (2003). Constructing a multi-valued and multi-labeled decision tree. *Expert Systems with Applications*, 25 (2), 199–209.
4. Chou, S. & Hsu, C. (2005). MMDT: a multi-valued and multi-labeled decision tree classifier for data mining. *Expert Systems with Applications*, 28 (2), 799–812.
5. Mantaras, R. L. D. (1991). A distance-based attribute selection measure for decision tree induction. *Machine Learning*, 6(1), 81–92.
6. Witten, I. H., & Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. China Machine Press, 2003.
7. Agrawal, R., Ghosh, S., Imielinski, T., Iyer, B., & Swami, A (1992). An interval classifier for database mining applications. Proceedings of the 18th International Conference on Very Large Databases (pp. 560–573). Vancouver, BC.
8. Ruggieri, S. (2002). Efficient C4.5. *IEEE Transactions on Knowledge and Data Engineering*, 14(2), 438–444.
9. Wang, H., & Zaniolo, C (2000). CMP: A fast decision tree classifier using multivariate predictions. Proceedings of the 16th International Conference on Data Engineering (pp. 449–460).

A Spatial Clustering Algorithm Based on SOFM

Zhong Qu and Lian Wang

College of Computer Science and Technology,
Chongqing University of Posts and Telecommunications,
400065 Chongqing, China
{quzhong, wanglian}@cqupt.edu.cn

Abstract. This paper analyses some important characteristics of self-organization map network. Based on this analysis, we propose a method that can overcome the insufficiencies of single self-organization feature map (SOFM) network. The implementation detail of our proposed self-organizing feature map network algorithm is also discussed. Our proposed algorithm has a number of advantages. It can overcome the insufficiencies identified in other similar clustering algorithms. It is able to find clusters in different shapes and is insensitive to input data sequence. It can process noisy and multi-dimensional data well in multi-resolutions. Furthermore the proposed clustering method can find the dense or sparse areas with different data distributions. It will be convenient to discover the distribution mode and interesting relationship among data. We have conducted numerous experiments in order to justify this novel ideal of spatial data clustering. It has been shown that the proposed method can be applied to spatial clustering well.

1 Introduction

Clustering is a process of grouping multidimensional input data spaces into multiple sets of similar objects, which makes the objects in the same cluster have higher similarity than those in others. Due to its unsupervised learning nature, clustering has been widely used in numerous applications, such as pattern recognition, image processing, market research etc. Clustering can be used to find out dense or sparse areas of different data distributions, which will be convenient for discovering the distribution model and interesting relationships among data. Self-organizing feature map (SOFM) groups input data according to their spatial organization model [1]. Because the neighbor neuron of SOFM can identify the neighborhood of input space, SOFM can identify the distribution and the topology structure of the training input data [2].

2 SOFM Algorithms

SOFM uses the sample X in D to study topology map: $f: D \subset R^d \rightarrow G \subset R^m$, where G is output map with a group of neurons and each neuron represents one element in m

dimensional Euclidean space; $r_i \in G$ represents the position of the i^{th} neuron of output map. $X = [x_1, x_2, \dots, x_s]^T \in D$ is the input vector. Assume each input vector is concurrently connected to each neuron of output map [3]. The vector weight of neuron i is expressed as $W_i = [w_{i1}, w_{i2}, \dots, w_{is}]^T \in R^s$. According to study rules:

$W_i(t+1) = W_i(t) + \alpha(t)\lambda(i, i^*)[X(t) - W_i(t)]$, where $t = 1, 2, 3, \dots$ is the discrete time coordinate, $\alpha(t) = 1, 2, 3, \dots$ is the study rate factor, and $\lambda(i, i^*)$ is the neighborhood function. Winner neuron i^* is defined as the neuron which weight has the smallest Euclidean distance in input space $X(t)$:

$$\|W_{i^*}(t) - X(t)\| \leq \|W_i(t) - X(t)\| \quad \forall r_i \in G.$$

The following is a standard neighborhood function presented by Kohonen: $\lambda(i, i^*) = \begin{cases} 1 & \text{for } \|r_i - r_{i^*}\| \leq N_{i^*}(t), \\ 0 & \text{otherwise} \end{cases}$, where $N_{i^*}(t)$ is some discrete time function.

3 SOFMF Algorithms

The idea of using a family of SOFMF network to clustering comes from teamwork [4], [5], [6]. And SOFMF uses a family of SOFMF network to work. Although single SOFMF network has some shortcomings, each member in the family can work together and rectify each other. On the whole, a satisfied clustering algorithm can be achieved.

3.1 The Construction of SOFMF

First construct a SOFMF with different output maps. A SOFMF with 2-dimension output is shown in figure 1.

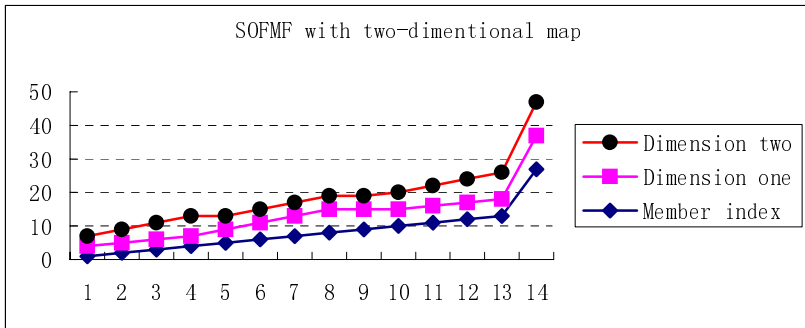


Fig. 1. SOFMF with 2-dimension map

With the increase of member index number, the output map rises slowly and monotonously. The neurons of SOFMF don't have to be managed in 2-dimension mode, it can be 1 dimension or 3 dimension or even higher. In this way, a SOFMF can be constructed with similar method.

3.2 Topological Similarity Matrix

Use input data set P to train $SOFM_k$ for several steps, and then P can be used to simulate on $SOFM_k$ and get partition c_k of P . The final result shows the objects in the same winner section have the higher similarity by measuring Euclidean distance. Because partition c_k of P has topology feature, thus topological similarity can be used to define the similarity among n number objects as following:

Assume $X(p), X(q) \in P$, then the similarity of $X(p)$ and $X(q)$ on $SOFM_k$ is:

$$TS_k(X(p), X(q)) = \begin{cases} 1 & , X(p) \in N_k(i) \text{ and } X(q) \in N_k(i) \text{ for some } i \in G_k \\ 0 & , X(p) \in N_k(i) \text{ and } X(q) \in N_k(j), i \neq j, i, j \in G_k \end{cases} \quad (1)$$

From the definition, the similarity of one object and itself is always 1. If a lower-triangle matrix is used to store the similarities between each pair among n objects, then a topological similarity matrix can be obtained. The symbol TSM_k represents the topological similarity matrix of $SOFM_k$:

$$TSM_k = \begin{bmatrix} 1 & & & & & \\ TS_k(X(2), X(1)) & 1 & & & & \\ TS_k(X(3), X(1)) & TS_k(X(3), X(2)) & 1 & & & \\ TS_k(X(n), X(1)) & TS_k(X(n), X(2)) & \dots & \dots & 1 & \end{bmatrix} \quad (2)$$

When $c_k = 1$, TSM_k is a $n \times n$ unit lower-triangle matrix, when $c_k = n$, TSM_k is a $n \times n$ unit matrix. These are two special cases for topological similarity matrix.

3.3 SOFMF Algorithms

A majority of clustering algorithms are operations on similarity and on topological similarity defined above.

Each member of SOFM family can produce a topological similarity matrix. But a single matrix nearly can't get a satisfied clustering result. Therefore, all the matrixes from SOFM family must learn the advantages mutually to counteract the weaknesses in order to correct each other. The algorithm defines a topological similarity matrix TSM for SOFM family, and uses TSM_k to update TSM by matrix addition. When $UsageFactor_k$ is less than a $AcceptUsageFactor$, the update process stops. Otherwise all the members of SOFM family must perform update process until member index stops increasing. The detail of update process has shown in following program with $AcceptUsageFactor = 0.4$.

- 1: Load data set P ;
- 2: initialize TSM with n-by-n unit lower-triangle matrix;
- 3: initialize sizes of output maps for the SOFM family with table 1 described in section 3.1;
- 4: update TSM with TSM_k obtained from $SOFM_k$;
- 5: build hierarchical clusters.

When the update stops, TSM can be used to build a hierarchical clustering. TSM is the result of cooperation of all members in SOFM family, therefore TSM stores the most interesting and valuable information of input data undoubtedly. The bigger element in TSM reflects the higher topological similarity among objects, on the contrary, the smaller elements reflects the lower topological similarity. Assume $Minlevel$ and $Maxlevel$ represent the minimal and the maximal topological similarity respectively. If TSM is used to construct an undirected complete graph (expressed as TSG) with adjacent matrix as weight, the corresponding clustering can be obtained after cutting off the edges beyond the range of $Minlevel$ and $Maxlevel$. For example, if current level is CHL, cluster at this level can be built by searching connected sub-graphs using depth-first search (DFS) technique. Obviously, the level value is in the range of $Minlevel$ and $Maxlevel$. The detail of the cluster construction process at this level has shown in following program (1), (2), (3).

(1) Update topology similarity matrix TSM with TSM_k :

- 1: Initialize current member index k with 0;
- 2: REPEAT;
- 3: $k = k + 1$;
- 4: construct $SOFM_k$, with size of the output map equal to the k^{th} member of the SOFM family, train it a fixed epochs, simulate $SOFM_k$ with input data set P to obtain partition number c_k and TSM_k from (1), (2) and (3);
- 5: $TSM = TSM + TSM_k$;
- 6: IF { $UsageFactor_k$ is smaller than $AcceptUsageFactor$ OR k reaches its maximum};
- 7: BREAK;
- 8: END.

(2) Construct clustering with TSM :

- 1: Calculate the maximal topological similarity $Maxlevel$ and the minimal topological similarity $Minlevel$ using TSM ;

- 2: initialize current hierarchical level CHL with $Minlevel$;
- 3: construct an undirected complete graph TSG using TSM as its adjacency matrix with weights;
- 4: Initialize current graph $CurTSG$ with TSG ;
- 5: REPEAT;
- 6: cut off the edges of $CurTSG$ whose weights are smaller than CHL , producing a new graph $NewTSG$;
- 7: search connected sub-graphs of $NewTSG$ using depth-first search (DFS) technique;
- 8: build sub-clusters in the current hierarchical level with connected sub-graphs;
- 9: Let the current graph $CurTSG$ be the new graph $NewTSG$;
- 10: $CHL = CHL + 1$;
- 11: IF (CHL is larger than $Maxlevel$) ;
- 12: BREAK;
- 13: END.

(3) Depth-first search algorithm:

Step 1. Choose any vertex, label it 1, and proceed to Step 2 with this vertex and label.

Step 2. Given a vertex labeled k,

- (a) if there exists a vertex adjacent to k which has not yet been labeled, assign to it the smallest unused label from the set $\{1, 2, \dots, n\}$ and repeat Step 2 for the new vertex and its labeled;
- (b) if all vertices adjacent to k have been labeled,
 - if $k=1$, stop;
 - if $k>1$, backtrack to the vertex from which you arrived at k at the time k was labeled and repeat Step 2 with this vertex and its label.

4 Simulation Experiments and Analyses

Because the clustering result from SOFMF is organized in hierarchical structure, and one data set can have multiple clustering levels, here only pick some typical levels to show. Each connected graph represents a cluster, and the number of connected graphs is the number of clusters in this level.

4.1 Simulation Experiment 1

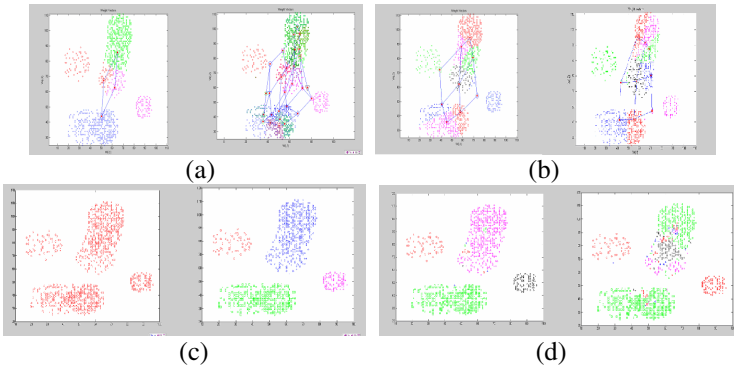


Fig. 2. Results of simulated experiment 1(a) The different results of 2×2 map and 5×5 map; (b) The results of different input orders; (c) SOFMF algorithm found a cluster at Level 1-4, found 5 clusters at Level 5-15, namely rational clusters found; (d) SOFMF algorithm found a cluster at level 18, 4 clusters produced 14 sub-clusters; at level 20, 4 clusters produced 64 sub-clusters

4.2 Simulation Experiment 2

Where Data number $n = 1762$, dimension $s = 2$, expected cluster number = 18. Cluster features are: embedded structure in clusters, various shapes clusters, such as triangle shape, four isolated points and parabola shape etc.

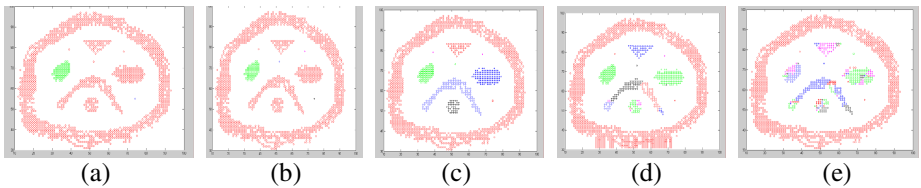


Fig. 3. Results of simulated experiment 1. (a)When Level = 13, algorithm found 3 clusters; (b) When Level = 14, algorithm found 5 clusters and 3 isolated points among them; (c)When Level = 15-21, all 10 expected clusters were found and all 4 isolated points were found; (d) When Level = 22,10 clusters produced 20 sub-clusters; (e)Level = 23, 56 clusters.

4.3 Simulation Experiment 3

Data features are: 3-dimension data with 500 points and 5 clusters, 4 rational clusters in projection on 2-dimension surface.

4.4 Simulation Experiment 4

Data features are: Iris standard database, $n = 150$, $s = 4$, cluster number expected equal 3. Cluster A: 1-50; Cluster B: 51-100; Cluster C: 101-150.

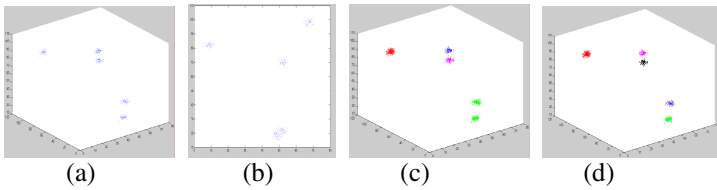


Fig. 4. Results of simulated experiment 2. (a) 3-D image; (b) 2-D image; (c) Level=1-15, 5 clusters; (d) Level=16-17, 6 clusters.

Cluster features are: cluster *A* separates from *B* and *C*, but *B* and *C* partially overlap, and the projections of Iris data on lower dimension spaces overlap. (1) When level=1-6, first cluster: 1-50; second cluster: 2-100. (2) When level equals 7-8, first cluster number: 50, 1-50; second cluster number: 48, 51-100 except 71, 73, 84 and 107; third cluster number: 52, 101-150 except 107, but add 71, 73 and 84. (3) When level is 9, first cluster number: 1-50; second cluster number: 11, sub-clusters (level 7-8): 51, 53, 55, 59, 66, 76, 77, 78, 87, 52 and 57; third cluster number: 37, sub-clusters(level 7-8): (48-11=37); Fourth cluster number: 52, as level 7-8.

4.5 Experiment Results and Analysis

SOFMF algorithm overcomes some problems existed in many clustering algorithms. For examples, cluster in any shape can be found, noisy data can be processed well, input sequence is insensitive, and multi-dimension data can be processed.

Furthermore good performance in time complexity can be achieved, by reason of the concurrency of neural network and the easy production of hardware.

In addition, clusters in different levels can be achieved according to different users, such as expert user, who can analyze results in detail and give some instructional principles for getting further positive and reasonable clustering results. It can be seen that (1) If one cluster would exist reasonably with several consecutive resolutions, it can be treated as a rational one. (2) If one cluster can be decomposed into several sub-clusters with higher resolution, it also is a rational one and its sub-clusters can be treated as clusters in a specially identified cluster. Moreover SOFM can change original input space to n -dimension neural cell space, so SOFM can increase dimension (certainly with heavy computation), and decrease dimension (usually in this way, thus 2-dimension neuron arrangement structure can be chosen).

5 Conclusion

Self-organization map network is applied to study clustering algorithm. Under the guidance of cluster theory, a new similarity measurement method is proposed: topological similarity. A new mathematical model is built under this similarity theory: topological similarity matrix to record cluster feature. The features of self-organization map network are analyzed. Based on this network, a method of using a family of self-organization map network is brought forward, which can overcome the insufficiencies of single self-organization map network. A SOFMF algorithm is

described and its implementation detail has been discussed. Finally the feasibility of this algorithm with several data sources has been proved by many simulation experiments. According to the comprehensive analyses of this algorithm, the following results are achieved: it can overcome the insufficiencies of many clustering algorithms; is able to find clusters in different shapes; is insensitive to the input data sequence, can process noisy and multi-dimensional data well, and has multi-resolutions. Furthermore clustering can find out dense or sparse areas of data distributions, which is convenient for discovering the distribution model and interesting relationships among data. Emphatically numerous experiments have proved this novel ideal of spatial data clustering.

References

1. Ordonez, C. and E. Omiecinski. Discovery association rules based on image content. in Proceedings of the 1999 IEEE Forum on Research and Technology Advances in Digital Libraries, Baltimore, MD, May 19-21, (1999) 38-49.
2. Bloch, Isabelle. Fuzzy relative position between objects in image processing: A morphological approach. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 21, No.7, (1999) 657-664.
3. Shekhar, S., Y. Huang, W. Wu, C. T. Lu, and S. Chawla. What's Spatial About Spatial Data Mining: Three Case Studies. in Data Mining for Scientific Engineering Applications, V. Kumar, R. Grossman, C. Kamath, K. Nambaru, Eds. Kluwer Academic(2001).
4. Han J Koperski K Stefanovic N.GeoMiner.A System Prototype for Spatial Data Mining[C].In.Proc ACM SIGMOD Conference on the Management of Data Tucson Arizona(1997).
5. Pokajac D Obradovic Z.Improved Spatial-Temporal Forecasting through Modeling of Spatial Residuals in Recent History. In.Proc First SIAM Int'I Conf on Data Mining SDM 2001.Chicago.USA(2001).
6. Roddick J F.Hornsby K.Spiliopoulou M. An Updated Temporal Spatial and Spatio-Temporal Data Mining and Knowledge Discovery Research Bibliography. In.Post-Workshop Proceedings of the International Workshop on Temporal, Spatial and Spatio-Temporal Data Mining TSDM2000 Springer Lecture Notes in Artificial Intelligence(2001).

Mining Spatial-temporal Clusters from Geo-databases

Min Wang, Aiping Wang, and Anbo Li

College of Geography Science,
Nanjing Normal University, Nanjing, 210097, China
sysj0918@126.com

Abstract. In order to mine spatial-temporal clusters from geo-databases, two clustering methods with close relationships are proposed, which are both based on neighborhood searching strategy, and rely on the sorted k -dist graph to automatically specify their respective algorithm arguments. We declare the most distinguishing advantage of our clustering methods is they avoid calculating the spatial-temporal distance between patterns which is a tough job. Our methods are validated with the successful extraction of seismic sequence from seismic databases, which is a typical example of spatial-temporal clusters.

1 Introduction

Clustering is a primary data mining method for structure or knowledge discovery in spatial databases [1][2]. In many circumstances, spatial data also contain temporal information. To consider temporal factor into spatial clustering can help us to discover the real underlying distribution rules in many spatial mining problems.

Seismic sequences are a typical and good example of spatial-temporal clusters. In seism, a seismic sequence is defined as a group of seismic events which have close relationships between each other and occur densely both in space and time. In this paper, we will discuss and design spatial clustering methods to discover this kind of clusters in geo-databases.

Similar work can be found in [3][4]. In [3], Golfarelli *et al* calculate the similarities of patterns in each feature dimension, multiply these similarities as the total pattern similarities, and then group those patterns less than certain similarity threshold as a cluster. In [4], Galic *et al.* normalize each feature dimension then clustering patterns with K-Means. In seism research, Wardlaw *et al* [5] propose spatial-temporal distance (D_{st}) between two seismic events, spatial-temporal converting index (C) to find spatial-temporal seismic clusters. In their methods, D_{ST} and C are defined as:

$$D_{ST} = \sqrt{d^2 + C^2 (\Delta t)^2} . \quad (1)$$

$$C = \sqrt{D_{ST}^2 - d^2} / \Delta t . \quad (2)$$

In (1), (2) d is the spatial while Δt is time distance between two seismic-events. To different seismic regions or belts, they give different C . But they can not give the physical meanings of these parameters, which is regrettable.

All these mentioned spatial-temporal clustering methods in nature are to modify the scale relationships between each feature dimension, or in other words, the contribution ratios of each feature dimension to calculate the distance between patterns in clustering. But the selection of such relationship is rather subjective and difficult to be endowed with scientific meanings.

In this paper, two clustering methods are proposed with close relationships and respective advantages and disadvantages. One is spatial-temporal grid method (ST-GRID, in abbreviation); the other is ST-DBSCAN. The main ideas of ST-GRID is: to partition spatial, temporal dimensions into a multi-dimension grid with different precisions, allocate patterns into the grid cells, and then extract and merge spatial-temporal dense regions as the final clusters. ST-DBSCAN is the extended form of DBSCAN [6] to spatial-temporal clustering problems. Both methods are based on the neighborhood searching strategy, and rely on the sorted k -dist graph [6] to find their respective algorithm arguments needed. We declare the advantages of our methods are: it only need one scan of the whole data set then are very efficient, and it avoids calculating the spatial-temporal distance between patterns which is very difficult. Our methods are validated with the successful extraction of seismic sequences from geodatabases.

2 Spatial-temporal Clustering Methods

2.1 Sorted k -Dist Graph

In ST-GRID, because of the different metrics of space and time, it's not suitable to partition the spatial, temporal dimension of the grid with same precision or same cell size. The crux is on how to specify automatically the two precisions which is completed with the sorted k -dist graph.

The principle of the sorted k -dist graph can be described as follows: in spatial clustering, we often regard those isolated patterns away from the clusters as noises. Since the distance between noises is relatively longer than that between clusters, we can often remove most noises by some distance threshold. If we take samples in the areas where the clusters and noises locate, calculate the distance from each sample point to its k -th nearest neighbor (the distance is called the k -dist value of that point), and sort the points of their k -dist values in descending order, then we can draw a graph called the sorted k -dist graph. It is obvious that the non-noises (sample points which are not noises) will have relatively smaller k -dist values, and what we want to do is to find the threshold of the 4-dist separating the non-noises and the noises in the sorted 4-dist graph. As depicted in Figure 1, the 4-dist values drop rapidly to the left of the crossing point of the reticle, but the drop to the right becomes much smoother. Such a 4-dist value becomes an appropriate threshold for separating the non-noises from the noises. In ST-GRID, we can set the grid size to less than half of this 4-dist value, which will basically satisfy the need of noise removal. This is because: if we regard the round regions around a point as a dense region which contains $k+1$ points with radius R , then we can also specify the border length of each grid cell to $2R$, in which the area of each cell will be close to the circle with radius R . If the points of the cell are more than $K+1$, then the cell is 'dense', and should be allocated into one cluster.

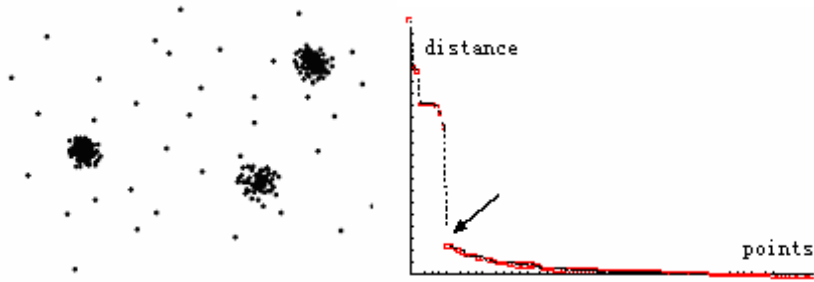


Fig. 1. A point set with its sorted 4-dist graph

2.2 ST-GRID Method

In ST-GRID, we specify the precisions of spatial, temporal dimensions with the sorted k -dist graph. We draw the sorted k -dist graph for the spatial, temporal dimensions, and then find their respective distance thresholds between noises and non-noises. The dense thresholds are same: $k+1$.

The algorithm of ST-GRID:

Input: spatial, temporal cell border lengths which are specified with the sorted k -dist graph of the grid, data set;

Output: clusters.

Construct a multi-dimension grid covering the whole spatial-temporal feature space;

Allocate every data point into these cells, and count the points of each cell;

Extract dense regions with threshold $k+1$;

Merge neighbor cells and mark them as one cluster;

Output these clusters.

In ST-GRID, the merging stage can be completed with the depth-first searching strategy. To a m -dimension grid, if we regard a pair of neighbors as two cells with only one dimension different, then every cell will have $2m$ neighbors except those boundary cells. ST-GRID will begin its depth-first, dense neighbor searching with an arbitrary dense cell until no new dense cells can be included within this searching, and mark these cells as one cluster. The next searching will begin with an un-searched dense cell, until all the dense cells are visited.

2.3 ST-DBSCAN

DBSCAN is a good clustering method for clustering clusters with non-sphere shapes. To extent DBSCAN to find spatial-temporal clusters, we separate its original argument, the neighborhood radius ε into two: the spatial neighborhood radius ε_s and temporal neighborhood radius ε_t .

Base on this, only if point p is inside the ε_s -neighborhood and ε_t -neighborhood of point q , point p can then be called ‘spatial-temporal directly density-reachable’ from point q . Similar as this, the other concepts of DBSCAN should also be extended accordingly.

Same as ST-GRID, ε_s , ε_t are calculate with the sorted k -dist graph. Draw the sorted k -dist graph for spatial and temporal dimensions; find the two distance thresholds between noises and non-noises, which are equal to ε_s , ε_t , the another argument $MinPts$ of DBSCAN equals to k . With these extensions, the searching neighborhood will be extended to spatial-temporal feature space. The core points would be those with more than $MinPts$ neighbors within their spatial-temporal neighborhood (round area with radius ε_s in space and ε_t in time).

3 Experiments

The experimental data are extracted from the database of ‘integrating seismic catalog in China and neighbor countries’ compiled by the China’s State Key Laboratory of Resources and Environmental Systems from mainly various seismic catalogs published by the Chinese national seismic bureau [7][8]. This database stores 620,000 seismic entries.

We first extract 6927 seismic events with magnitude ≥ 2.5 in North of China (37° - 41° N, 113° - 121° E) from year 1900, which include three seismic sequences: Xingtai, Bohai, Tangsan sequence. In Figure 2, the three ellipses from up to down are Tangsan, Bohai, Xingtai sequences. Each sequence not only distributes densely in space, but occupies its dense time ranges with different lengths. We will try to extract these sequences with ST-GRID and ST-DBSCAN, and compare their respective performances.

We first take samples in the testing area and calculate the sorted k -dist graph to get the needed inputs of ST-GRID and ST-DBSCAN. The rectangles in Figure 2 are the sample areas, which include a dense area and a noise area, with 643 seismic events in total. Calculating the sorted 4-dist graph of the sample areas, we get spatial distance threshold 6000m, temporal distance threshold 610d. Inputs of ST-GRID are: the spatial precision= $6000 \times 2 = 12000$ m, temporal precision= $610 \times 2 = 1220$ d. The spatial-temporal grid covering the whole testing area is separated into $57 \times 40 \times 28$ cells (longitude, latitude, time), and the dense threshold is 5. Inputs of ST-DBSCAN are: $\varepsilon_s = 6000$ m, $\varepsilon_t = 610$ d, $MinPts = 4$.

ST-GRID extracts 17 while ST-DBSCAN extracts 33 clusters. Both include the three 3 sequences we care about, with small distribution differences in both space and time. The thick border polygons in Figure 2 are the extracted spatial areas of our methods, while the thin border polygons are the real spatial areas of the three sequences. We can find they are very close to their counterparts, while our areas are a little bit smaller than the real. It may be caused by that we only extract partly seismic events from the whole data set.

Both methods give coincident results in temporal clustering with the real time distribution (See Table 2). But we find both methods override their actual time

ranges (for example, the end time of Tangsan sequence). To ST-GRID, the time range of each cell is 1220d (610*2), about 4 years. If there exist clusters and noises which occur within 4 years and fall into same cell, ST-GRID can not distinguish them. ST-DBSCAN has the similar disadvantages. Besides, some dense areas may be divided into several cells, which causes some cells covering the brim of clusters to be discarded. We call this shortcoming the roughness of ST-GRID while ST-DBSCAN is free of this. The small differences of the division of spatial-temporal dense areas between the two methods are caused by their neighborhood searching strategies, rectangle cells and round searching areas, and the ‘roughness’ of ST-GRID.

We select more sequences, which are Haicheng, Songpan, Yanyuan sequences to validate our methods ulteriorly. We also use the sorted 4-dist graph to calculate the parameters of both methods which are list in Table 1. See Table 2 for the outputs of the time ranges. All these results validate our methods once again.

Table 1. Parameters of our methods in extracting Haicheng, Songpan, Yanyuan sequences

Sequence	Data numbers	Spatial distance threshold	Time distance threshold	Cell numbers of ST-GRID	Inputs of ST-DBSCAN
Haicheng sequence	1311	4655m	447d	36×28×17	$\epsilon_s=4655m,$ $\epsilon_t=447d$
Songpan sequence	996	6490m	798d	15×14×34	$\epsilon_s=6490m,$ $\epsilon_t=798d$
Yanyuan sequence	939	12606m	239d	17×10×35	$\epsilon_s=12606m,$ $\epsilon_t=239d$

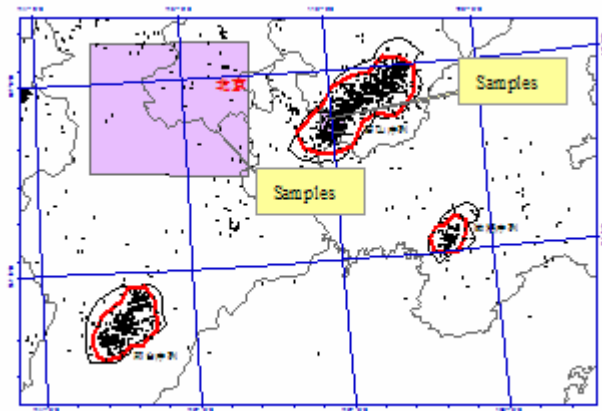


Fig. 2. Spatial boundaries of Tangsan, Bohai, Xingtai sequences

Table 2. Time ranges of the sequences

Sequence	Time range of actual clusters	Time range of ST-GRID	Time range of ST-DBSCAN	Time range of sequences
Tangsan	1976-1985	1974-1986	1975-1978	1976-1980
Bohai	1969-1972	1969-1973	1969-1971	1969-1972
Xingtai	1966-1971	1965-1973	1965-1971	1966-1985
Haicheng	1975-1983	1975-1983	1975-1983	1975-1983
Songpan	1973-1976	1973-1978	1973-1978	1973-1976
Yanyuan	1976-1981	1976-1978	1976-1981	1972-1981

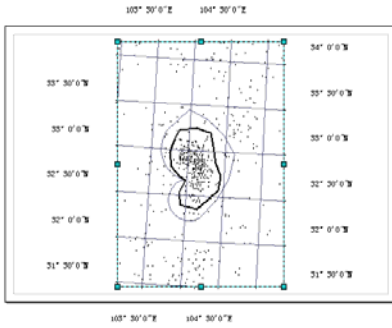


Fig. 3. Spatial boundary of Songpan sequence

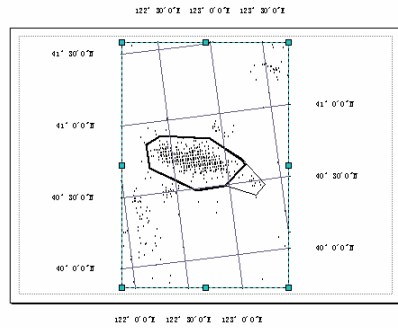


Fig. 4. Spatial boundary of Haicheng sequence

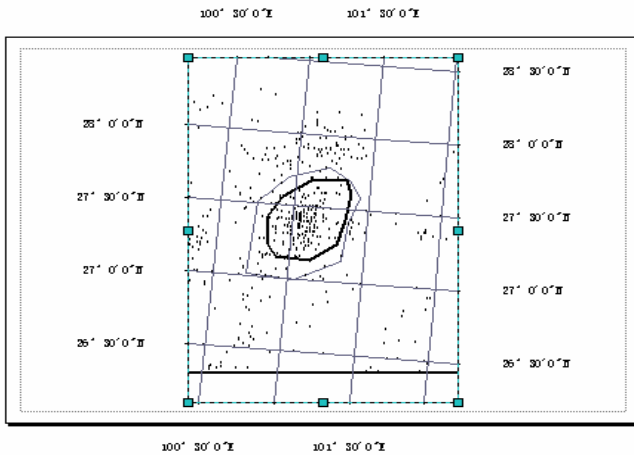


Fig. 5. Spatial boundary of Yanyuan sequence

Table 3. Comparison between the two clustering methods

Algorithm	Time efficiency	Space efficiency	Precision	Inputs
ST-GRID	High	Need store grid	Relatively high	3
ST-DBSCAN	Relatively high	Only need store points	High	3

4 Conclusions

In this paper, two clustering methods to extract spatial-temporal clusters from geo-databases are discussed and validated with the successful extraction of seismic sequences from seismic databases. From these experiments, we can find the core idea of our two methods is neighborhood searching in same. The differences are: ST-GRID searches between neighborhood cells while ST-DBSCAN searches the neighborhood of points. One groups the cells while the other groups the points. Table 3 sums up both methods advantages and disadvantages. ST-GRID only needs one scan of the whole data set, which is of linear time complexity and high performing speed. Because ST-DBSCAN involves neighborhood searching of points, its time complexity will reach $O(n^2)$ if without any spatial indices. It's not very good in time efficiency, and the improvement is to introduce R*- tree index structure [9] into the method. But ST-GRID needs additional disk space to store the grid structure, which is of lower space efficiency than ST-DBSCAN. Because of the shortcomings of 'roughness', the clustering precision of ST-GRID is a little lower than ST-DBSCAN. Users can select one method according to their needs in spatial-temporal analysis. Because the dense thresholds of both methods are global and unique, in many circumstances, they will blur many actual spatial distributing patterns. For further studies, we will pay attention to find some local-adaptive rules to specify automatically these important parameters.

Acknowledgment

This work is supported by Chinese National Natural Science Foundation (No.40401039) and startup fund for excellent scholars fetched in of Nanjing Normal University (No. 2006105XGQ0035).

References

1. Koperski K., Adhikary J., and Han J., 1996: Spatial Data Mining: Progress and Challenges Survey Paper. Proc. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal, Canada
2. Ester, M., Kriegel, H.-P., Sander, J. and Xu, X., 1998: Clustering for Mining in Large Spatial Databases. Special Issue on Data Mining, KI-Journal, 12, 18-24
3. Golfarelli M., Rizzi S., 2000. Spatial-Temporal Clustering of Tasks for Swap-Based Negotiation Protocols in Multi-Agent Systems. Proceedings 6th International Conference on Intelligent Autonomous Systems.172-179

4. Galic S., Loncaric S. and Tesla E.N., 2001. Cardiac Image segmentation using spatial-temporal clustering. Proceedings of SPIE Medical Imaging, San Diego
5. Wardlaw R.L, Frohlich C. and Davis, S.D., 1990. Evaluation of precursory seismic quiescence in sixteen subduction zones using single-link cluster analysis. PAGEOPH:134
6. Ester M., Kriegel H.-P., Sander J., Xu X.. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)
7. The seismic analysis and forecasting center, 1980. China Seismological Bureau, The Seismic Catalog in East of China, Beijing: The Earthquake Publishing House.
8. The seismic analysis and forecasting center, 1989. China Seismological Bureau, The Seismic Catalog in West of China, Beijing: The Earthquake Publishing House.
9. Beckmann N., Kriegel H.P., Schneider R. and Seeger B., 1990. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. Proc. ACM SIGMOD Int. Conf. On Management of Data, 322-331

A Fuzzy Subspace Algorithm for Clustering High Dimensional Data

Guojun Gan¹, Jianhong Wu¹, and Zijiang Yang²

¹ Department of Mathematics and Statistics, York University, Toronto, Ontario, Canada M3J 1P3

{gjgan, wujh}@mathstat.yorku.ca

² School of Information Technology, Atkinson Faculty of Liberal and Professional Studies, York University, Toronto, Ontario, Canada, M3J 1P3

zyang@mathstat.yorku.ca

Abstract. In fuzzy clustering algorithms each object has a fuzzy membership associated with each cluster indicating the degree of association of the object to the cluster. Here we present a fuzzy subspace clustering algorithm, FSC, in which each dimension has a weight associated with each cluster indicating the degree of importance of the dimension to the cluster. Using fuzzy techniques for subspace clustering, our algorithm avoids the difficulty of choosing appropriate cluster dimensions for each cluster during the iterations. Our analysis and simulations strongly show that FSC is very efficient and the clustering results produced by FSC are very high in accuracy.

1 Introduction

Data clustering[1] is an unsupervised process that divides a given data set into groups or clusters such that the points within the same cluster are more similar than points across different clusters. Data clustering is a primary tool of data mining, a process of exploration and analysis of large amount of data in order to discover useful information, thus has found applications in many areas such as text mining, pattern recognition, gene expressions, customer segmentations, image processing, to name just a few.

For data sets in high dimensional spaces, most of the conventional clustering algorithms do not work well in terms of effectiveness and efficiency, because of the inherent sparsity of high dimensional data [2]. To cluster data in high dimensional spaces, we encounter several problems. First of all, the distance between any two points becomes almost the same [2], therefore it is difficult to differentiate similar data points from dissimilar ones. Secondly, clusters are embedded in the subspaces of the high dimensional space, and different clusters may exist in different subspaces of different dimensions [3]. Because of these problems, almost all conventional clustering algorithms fail to work well for high dimensional data sets. One possible solution is to use dimension reduction techniques such as PCA(Principal Component Analysis) and Karhunen-Loève Transformation, or feature selection techniques [3].

The idea behind dimension reduction approaches and feature selection approaches is to first reduce the dimensionality of the original data set by removing less important variables or by transforming the original data set into one in a low dimensional space, and then apply conventional clustering algorithms to cluster the new data set. In either dimension reduction approaches or feature selection approaches, it is necessary to prune off some variables, which may lead to a significant loss of information. This can be illustrated by considering a 3-dimensional data set that has 3 clusters: one is embedded in (x, y) -plane, another is embedded in (y, z) -plane and the third one is embedded in (z, x) -plane. For such a data set, an application of a dimension reduction or a feature selection method is unable to recover all the cluster structures, for the 3 clusters are formed in different subspaces. In general, clustering algorithms based on dimension reduction or feature selection techniques generate clusters that may not fully reflect the original cluster structures.

This difficulty that conventional clustering algorithms encounter in dealing with high dimensional data sets inspired the invention of subspace clustering algorithms or projected clustering algorithms [3] whose goal is to find clusters embedded in subspaces of the original data space with their own associated dimensions. Some subspace clustering algorithms are designed to identify arbitrarily oriented subspace clusters (e.g. ORCLUS and Projective k -Means) whose cluster dimensions are linear combinations of the original dimensions, while others are designed to discover regular subspace clusters (e.g. PART and SUBCAD) whose cluster dimensions are elements of the set of the original dimensions.

However, almost all of the subspace clustering algorithms give equal non-zero weights to cluster dimensions and zero weights to non-cluster dimensions. Consider a cluster embedded in a 50-dimensional subspace of a 100-dimensional data set, for example, the cluster dimensions (say $1, 2, \dots, 50$) found by PROCLUS [4] are assumed to have equal contributions to the cluster, but other dimensions ($51, 52, \dots, 100$) are assumed to have zero contributions to the cluster. This practice leads to the problem of how to choose the cluster dimensions of a specific cluster.

Motivated by fuzzy clustering and LAC [5], we propose a fuzzy subspace clustering algorithm, FSC, to cluster high dimensional data sets. FSC finds regular subspace clusters with each dimension of the original data being associated with each cluster with a weight. The higher density of a cluster in a dimension, the more weight will be assigned to that dimension. In other words, all dimensions of the original data are associated with each cluster, but they have different degrees of association with that cluster.

2 Related Work

The recent subspace clustering algorithms can be roughly classified into three categories: Grid-based algorithms such as CLIQUE [3], MAFIA [6], Partitioning and/or hierarchical algorithms such as ORCLUS [7], FINDIT [8], and Neural Network-based algorithms such as PART [2].

CLIQUE [3] first partitions the whole data space into non-overlapping rectangular units, and then searches for dense units and merges them to form clusters. The subspace clustering is achieved due to the fact that if a k -dimension unit $(a_1, b_1) \times (a_2, b_2) \times \dots \times (a_k, b_k)$ is dense, then any $(k - 1)$ -dimension unit $(a_{i_1}, b_{i_1}) \times (a_{i_2}, b_{i_2}) \times \dots \times (a_{i_{k-1}}, b_{i_{k-1}})$ is also dense, where (a_i, b_i) is the interval of the unit in the i -th dimension, $1 \leq i_1 < i_2 < \dots < i_{k-1} \leq k$. ENCLUS [9] and MAFIA [6] are also Grid-based subspace clustering algorithms.

PROCLUS [4] is a variation of k -Medoid algorithm [10] for subspace clustering. PROCLUS finds out the subspace dimensions of each cluster via a process of evaluating the locality of the space near it. FINDIT [8], ORCLUS [7], FLOC [11], DOC [12], SUBCAD [13] and projective k -Means [14] are also partitioning subspace clustering algorithms.

PART [2] is a new neural network architecture to find projected clusters for data sets in high dimensional spaces. In PART, a so-called selective output signaling mechanism is provided in order to deal with the inherent sparsity in the full space of the high dimensional data points. PART is very effective to find the subspace in which a cluster is embedded, but the difficulty of tuning some parameters in the algorithm and the sensitivity to data input order restrict its application. CLTree [15] is an algorithm for clustering numerical data based on a supervised learning technique called decision tree construction. The resulting clusters found by CLTree are described in terms of hyper-rectangle regions. The CLTree algorithm is able to separate outliers from real clusters effectively, since it naturally identifies sparse and dense regions.

LAC [5] defines subspace clusters as weighted clusters such that each cluster consists of a subset of data points together with a vector of weights. To be precise, let us consider a data set D of n points in the d -dimensional Euclidean space and a set of centers $\{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\} \subset \mathbb{R}^d$, coupled with a set of corresponding weight vectors $\{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k\} \subset \mathbb{R}^d$. LAC defines the j th ($1 \leq j \leq k$) cluster as $C_j = \left\{ \mathbf{x} \in D : \left(\sum_{i=1}^d w_{ji}(x_i - z_{ji})^2 \right)^{\frac{1}{2}} < \left(\sum_{i=1}^d w_{li}(x_i - z_{li})^2 \right)^{\frac{1}{2}}, \forall l \neq j \right\}$, where x_i, z_{ji} and w_{ji} are the i th components of \mathbf{x}, \mathbf{z}_j and \mathbf{w}_j , respectively. The centers and weights are chose such that the error measure, $E = \sum_{j=1}^k \sum_{i=1}^d w_{ji} e^{-X_{ji}}$, is minimized

subject to the constraints $\sum_{i=1}^d w_{ji}^2 = 1, \forall j$, where $X_{ji} = \frac{1}{|C_j|} \sum_{\mathbf{x} \in C_j} (x_i - z_{ji})^2$.

3 Fuzzy Subspace Clustering Algorithm

The main idea behind our algorithm is to impose weights to the distance measure of the k -Means algorithm[16] in order to capture appropriate subspace information. Given a data set $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ in the d -dimensional Euclidean space and k centers $Z = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$, then the objective function of the k -Means

algorithm is formulated as $E = \sum_{j=1}^k \sum_{\mathbf{x} \in C_j} \|\mathbf{x} - \mathbf{z}_j\|^2$, where $\|\cdot\|$ is the Euclidean norm and C_j is the j th cluster.

Similar to LAC [5], we associate with each cluster a weight vector in order to capture the subspace information of that cluster. To be more precise, let W be a $k \times d$ real matrix satisfying the following conditions:

$$0 \leq w_{jh} \leq 1, \quad 1 \leq j \leq k, \quad 1 \leq h \leq d, \tag{1a}$$

$$\sum_{h=1}^d w_{jh} = 1, \quad 1 \leq j \leq k. \tag{1b}$$

Then the h th dimension is associated with the j th cluster to a degree of w_{jh} or the j th cluster has dimension weights specified by $w_{j1}, w_{j2}, \dots, w_{jd}$. We call the weight matrix W the fuzzy dimension weight matrix. Mathematically, the objective function of our algorithm is formatted as

$$E_f(W, Z) = \sum_{j=1}^k \sum_{\mathbf{x} \in C_j} \sum_{h=1}^d w_{jh}^\alpha (x_h - z_{jh})^2, \tag{2}$$

where $\alpha \in (1, \infty)$ is a weighting component or fuzzier. Given the estimates of Z and W , the j th cluster are formulated as

$$C_j = \{\mathbf{x} \in D : \sum_{h=1}^d w_{jh}^\alpha (x_h - z_{jh})^2 = \min_{1 \leq l \leq k} \sum_{h=1}^d w_{lh}^\alpha (x_h - z_{lh})^2\}, \tag{3}$$

together with the fuzzy dimension weights $\mathbf{w}_j = (w_{j1}, w_{j2}, \dots, w_{jd})$.

To find the cluster centers Z given the estimate of W such that the objective function $E_f(W, Z)$ defined in Equation (2) is minimized, we take partial derivatives of $E_f(W, Z)$ with respect to z_{jh} s, set them to zeros and solve the resulting equation system. That is,

$$\frac{\partial E_f(W, Z)}{\partial z_{jh}} = \sum_{\mathbf{x} \in C_j} -2w_{jh}^\alpha (x_h - z_{jh}) = 0, \quad 1 \leq j \leq k, \quad 1 \leq h \leq d,$$

which give

$$z_{jh} = \frac{\sum_{\mathbf{x} \in C_j} w_{jh}^\alpha x_h}{\sum_{\mathbf{x} \in C_j} w_{jh}^\alpha} = \frac{\sum_{\mathbf{x} \in C_j} x_h}{|C_j|}, \quad 1 \leq j \leq k, \quad 1 \leq h \leq d, \tag{4}$$

where $|C_j|$ denotes the number of points in C_j .

To find the fuzzy dimension weight matrix W given the estimate of Z such that the objective function $E_f(W, Z)$ is minimized, we use the method of Lagrange multipliers. To do this, we first write the Lagrangian function as

$$F(W, Z, \Lambda) = E_f(W, Z) - \sum_{j=1}^k \lambda_j \left(\sum_{h=1}^d w_{jh} - 1 \right).$$

By taking partial derivatives, we have

$$\frac{\partial F(W, Z, \Lambda)}{\partial w_{jh}} = \sum_{\mathbf{x} \in C_j} \alpha w_{jh}^{\alpha-1} (x_h - z_{jh})^2 - \lambda_j = 0, \quad 1 \leq j \leq k, 1 \leq h \leq d,$$

and

$$\frac{\partial F(W, Z, \Lambda)}{\partial \lambda_j} = \sum_{h=1}^d w_{jh} - 1 = 0, \quad 1 \leq j \leq k,$$

which, with some simple manipulations, leads to

$$w_{jh} = \frac{1}{\sum_{l=1}^d \left[\frac{\sum_{\mathbf{x} \in C_j} (x_h - z_{jh})^2}{\sum_{\mathbf{x} \in C_j} (x_l - z_{jl})^2} \right]^{\frac{1}{\alpha-1}}}, \quad 1 \leq j \leq k, 1 \leq h \leq d. \tag{5}$$

To avoid divide-by-zero error, we introduce a small bias ϵ (say $\epsilon = 0.0001$) in Equation (5). That is, we update W given the estimate of Z as follows:

$$w_{jh} = \frac{1}{\sum_{l=1}^d \left[\frac{V_{jh} + \epsilon}{V_{jl} + \epsilon} \right]^{\frac{1}{\alpha-1}}}, \quad 1 \leq j \leq k, 1 \leq h \leq d, \tag{6}$$

where $V_{jh} = \sum_{\mathbf{x} \in C_j} (x_h - z_{jh})^2$ for $1 \leq j \leq k$ and $1 \leq h \leq d$.

We see from Equation (4) and Equation (6) that FSC is very similar to the fuzzy k -Means algorithm [17] in terms of the way they update centers and fuzzy weights. FSC starts with initial centers Z , and then repeats estimating the fuzzy dimension weight matrix W given the estimate of Z and estimating the centers Z given the estimate of W until it converges.

4 Experimental Evaluations

FSC is coded in C++ programming language. Synthetic data sets are generated by a Matlab program using the method introduced by Aggarwal et al. [4]. In our experiments, we specify $\alpha = 2.1$.

Our first data set contains 300 3-dimensional points with 3 clusters embedded in different planes. We run FSC 100 times on this data with $k = 3$ and get the

Table 1. FSC: The input clusters (left) and the output fuzzy dimension weights together with the cluster dimensions (right) for the first data set

Input	Dimensions	Points	Found	w_{i1}	w_{i2}	w_{i3}	Dimensions	Points
A	1,2	100	1	0.040034	0.444520	0.515446	2, 3	100
B	2,3	100	2	0.573635	0.391231	0.035134	2, 1	100
C	3,1	100	3	0.427178	0.036284	0.536538	1, 3	100

same result. The best $E_f(W, Z)$ and average $E_f(W, Z)$ of the 100 runs are identical to 10.691746. Table 1 summarizes the clustering results. We see from Table 1 that FSC is capable of clustering each object correctly and at the same time identifying the true subspaces for each cluster. Note that the cluster dimensions of a cluster are arranged in ascending order according to their weights and the cutting point is obtained by clustering the fuzzy dimension weights of the cluster into 2 groups by k -Means.

Table 2. FSC: Dimensions of input clusters (left) and output clusters (right) for the second data set

Input	Dimensions	Points	Found	Dimensions	Points
A	6,7,8,10,11	387	1	20,13,10,1	129
B	5,7,8,10,11,12,13,16	87	2	9,3,1,6,10,13,11,18	80
C	3,5,6,10,12,13	317	3	3,5,12,6,13,10	317
D	1,3,6,9,10,11,13,18	80	4	11,6,10,7,8	387
E	1,10,13,20	129	5	7,16,11	87

Our second data set contains 1,000 20-dimensional points with 5 clusters embedded in different subspaces of different dimensions (See Table 2). We also run FSC 100 times on this data set with $k = 5$. The best $E_f(W, Z)$ and the average $E_f(W, Z)$ are 1102.126302 and 1396.434035, respectively. In particular, the number of correct clusterings is 49 out of 100. The best output is given in Table 2 from which we see that in the best case all subspace clusters are recovered by FSC except for cluster B where k -Means gives only 3 cluster dimensions.

Table 3. FSC: Dimensions of input clusters for the third data set

Input	Dimensions	Points
A	8,17,27,46,48,52,56,57,68,71,76,80,89,93	1462
B	5,8,17,26,27,37,40,46,48,53,56,71,84,86,89,95,97	4406
C	7,9,17,26,41,46,65,73,84,86,97	1415
D	4,17,25,26,45,65,75,83,84,97	556
E	2,6,17,18,26,29,32,39,45,49,75,83,84,97	1661
Outlier		500

Our third data set has 10,000 100-dimensional points with 5 clusters embedded in different subspaces of different dimensions and contains 500 outliers. We run FSC on this data set 5 times with $k = 5$ and 5 times with $k = 6$. The results are given in Table 4 from which we see that all objects are clustered correctly in both cases and all outliers are differentiated from real clusters in the case of $k = 6$. In the case of $k = 5$, the best $E_f(W, Z)$ and the average $E_f(W, Z)$ are 3328.104712 and 4460.156128, respectively, and the number of correct clusterings is 2 out of 5, while in the case of $k = 6$, the best $E_f(W, Z)$ and the average $E_f(W, Z)$ are 6102.280185 and 7703.287459, respectively, and the number of correct clusterings is 3 out of 5. We also see from Table 4 that cluster 1 has the

Table 4. FSC: The misclassification matrices when $k = 5$ (top left) and $k = 6$ (top right), output clusters when $k = 5$ (middle) and $k = 6$ (bottom) for the third data set

	1	2	3	4	5		1	2	3	4	5	6
A	0	0	0	1462	0	A	0	1462	0	0	0	0
B	4406	0	0	0	0	B	0	0	0	4406	0	0
C	0	1415	0	0	0	C	0	0	0	0	1415	0
D	0	0	0	0	556	D	0	0	0	0	0	556
E	0	0	1661	0	0	E	0	0	1661	0	0	0
Outlier	0	0	0	0	500	Ourlier	500	0	0	0	0	0

Found	Dimensions	Points
1	48,56,53,17,5,46,95,86,26,84,40,97	4460
2	73,46,41,86,84,26,97,17,7,65	1415
3	26,84,17,32,39,49,6,18,83,29,75,45,2,97	1661
4	76,89,71,27,56,52,68,8,46,17,57,93,80,48	1462
5	83,25,45,4,65,97	1056

Found	Dimensions	Points
1	30,22,35,7,16,11,73,100,2,33,39,10,53,62,34,12,45,9,76,54,85,61,47,82,65,20,14,43,94,77,99,41,70,96,74,23,68,59,19,50,71,92,57,26,32, 3,15,51,98,37,80,79,84,49	500
2	76,89,71,27,56,52,68,8,46,17,57,93,80,48	1462
3	26,84,17,32,39,49,6,18,83,29,75,45,2,97	1661
4	48,56,53,17,5,46,95,86,26,84,40,97	4406
5	73,46,41,86,84,26,97,17,7,65	1415
6	65,26,17,83,4,75,84,25	556

number of cluster dimensions significantly greater than other clusters do. This indicates that cluster 1 may be an outlier cluster.

The experiments presented above show that FSC is very powerful in recovering clusters embedded in subspaces of high dimensional spaces. FSC is simple and natural in terms of the presentation of the algorithm, and it is much easier to use than other subspace clustering algorithms such as PART and PROCLUS.

5 Conclusions and Remarks

We presented the fuzzy subspace clustering algorithm FSC for clustering high dimensional data sets. The novel contribution is the adoption of some fuzzy techniques for subspace clustering in a way that each dimension has a fuzzy dimension weight associated with each cluster. The experimental results have shown that FSC is very effective in recovering the subspace cluster structures embedded in high dimensional data. It is certainly of great interest to us if we can adopt fuzzy techniques for identifying arbitrarily oriented subspace clusters in high dimensional data.

References

- [1] Jain, A., Murty, M., Flynn, P.: Data clustering: A review. *ACM Computing Surveys* **31** (1999) 264–323
- [2] Cao, Y., Wu, J.: Projective ART for clustering data sets in high dimensional spaces. *Neural Networks* **15** (2002) 105–120
- [3] Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: *SIGMOD Record ACM Special Interest Group on Management of Data.* (1998) 94–105
- [4] Aggarwal, C., Wolf, J., Yu, P., Procopiuc, C., Park, J.: Fast algorithms for projected clustering. In: *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, ACM Press (1999) 61–72
- [5] Domeniconi, C., Papadopoulos, D., Gunopulos, D., Ma, S.: Subspace clustering of high dimensional data. In: *Proceedings of the SIAM International Conference on Data Mining*, Lake Buena Vista, Florida (2004)
- [6] Goil, S., Nagesh, H., Choudhary, A.: MAFIA: Efficient and scalable subspace clustering for very large data sets. Technical Report CPDC-TR-9906-010, Center for Parallel and Distributed Computing, Department of Electrical & Computer Engineering, Northwestern University (1999)
- [7] Aggarwal, C., Yu, P.: Finding generalized projected clusters in high dimensional spaces. In Chen, W., Naughton, J.F., Bernstein, P.A., eds.: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, May 16–18, 2000, Dallas, Texas, USA. Volume 29., ACM (2000) 70–81
- [8] Woo, K., Lee, J.: FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting. PhD thesis, Korea Advanced Institute of Science and Technology, Department of Electrical Engineering and Computer Science (2002)
- [9] Cheng, C., Fu, A., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM Press (1999) 84–93
- [10] Kaufman, L., Rousseeuw, P.: *Finding Groups in Data—An Introduction to Cluster Analysis.* Wiley series in probability and mathematical statistics. John Wiley & Sons, Inc., New York (1990)
- [11] Yang, J., Wang, W., Wang, H., Yu, P.: δ -clusters: capturing subspace correlation in a large data set. *Data Engineering, 2002. Proceedings. 18th International Conference on* (2002) 517–528
- [12] Procopiuc, C., Jones, M., Agarwal, P., Murali, T.: A monte carlo algorithm for fast projective clustering. In: *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, ACM Press (2002) 418–427
- [13] Gan, G., Wu, J.: Subspace clustering for high dimensional categorical data. *ACM SIGKDD Explorations Newsletter* **6** (2004) 87–94
- [14] Agarwal, P., Mustafa, N.: k -means projective clustering. In: *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems(PODS)*, Paris, France, ACM (2004) 155–165
- [15] Liu, B., Xia, Y., Yu, P.: Clustering through decision tree construction. In: *Proceedings of the ninth international conference on Information and knowledge management*, McLean, Virginia, USA, ACM Press (2000) 20–29
- [16] Hartigan, J.: *Clustering Algorithms.* John Wiley & Sons, Toronto (1975)
- [17] Huang, Z., Ng, M.: A fuzzy k -modes algorithm for clustering categorical data. *IEEE Transactions on Fuzzy Systems* **7** (1999) 446–452

Robust Music Information Retrieval on Mobile Network Based on Multi-Feature Clustering

Won-Jung Yoon, Sanghun Oh, and Kyu-Sik Park

Dankook University

Division of Information and Computer Science

San 8, Hannam-Dong, Yongsan-Ku, Seoul Korea, 140-714
{helloril, taru74, kspark}@dankook.ac.kr

Abstract. In this paper, a music information retrieval system in real mobile environment is proposed. In order to alleviate distortions due to the mobile noise, a noise reduction algorithm is applied and then a feature extraction using Multi-Feature Clustering is implemented to improve the system performance. The proposed system shows quite successful performance with real world cellular phone data.

1 Introduction

This paper is motivated from the observation that for the music industry to sell musical commodities and services via the internet and there is a high demands to develop the advanced tools to support new ways to retrieve and browse with the music audio content through the ubiquitous mobile services.

Content-based music information retrieval (MIR) is typically performed by analyzing a query signal to obtain a number of representative music features, and then applying a similarity measure to the derived features to locate database files that are most similar to the query signal. A number of content-based music retrieval methods are available in the literature as in [1-3]. However these studies are mainly concern on the PC based music retrieval system with no noise condition. These methods are tend to fail when the query music signal contains background noises and network errors as in mobile environment.

Burges et al [4] proposed an automatic dimensionality reduction algorithm called Distortion Discriminant Analysis (DDA) for the mobile audio fingerprint system. Kurozumi et al [5] combined local time-frequency-region normalization and robust subspace spanning, to search for the music signal acquired by the cellular phone. Phillips [6] introduces a new approach to audio fingerprinting that extracts a 32-bit energy differences along the frequency and time axes to identify the query music. Wang et al. [7] extracts the landmark and audio fingerprint features where landmark represents the time-point of the spectral features.

This paper focuses on the following issues on the mobile music information retrieval system in mobile network. In order to release noises due to a mobile network

and environment, a signal subspace noise reduction algorithm is applied. Further effort to extract a noise robust feature is performed by Multi-feature clustering. This Multi-feature clustering technique can also resolve the problem with system performance due to the different input query patterns.

This paper is organized as follows. Section 2 describes proposed mobile-based music information retrieval system. Section 3 describes methods of robust feature extraction for mobile-based music information retrieval (MIR) system. MFC method is introduced. Section 4 shows experimental results of the proposed system. Finally, a conclusion is given in section 5.

2 Proposed Mobile-Based MIR System

The proposed system is illustrated in Fig. 1. The system consist 4 stages – music signal acquisition, mobile noise reduction, robust feature extraction, and music information retrieval and SMS service to the user request. Firstly, a queried music signal is picked up by the single microphone of the cellular phone and then transmitted to the MIR server. Then the signal is acquired by the INTEL dialogic D4PCI-U board in 8 kHz sampling rate, 16 bit, MONO. Secondly, a signal subspace noise reduction algorithm is applied to the query signal. This stage is required to enhance music signal by reducing mobile noises. Thirdly, pre-defined set of features are extracted from the enhanced query signal. At this moment, MFC and SFS feature optimization is applied to extract robust features against mobile noise. Finally, the queried music is retrieved and the retrieval result will be transmitted to the user request via SMS server.

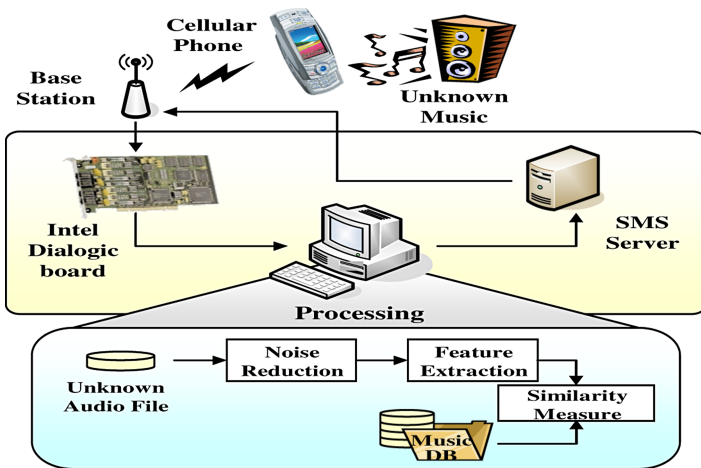


Fig. 1. Proposed mobile-based MIR system

3 Music Feature Extraction and Multi-Feature Clustering

3.1 Music Feature Extraction

A well known signal subspace noise reduction algorithm [8] is applied to the query signal acquired by the cellular phone to reduce the mobile noises. Enhancement is performed by removing the noise subspace and estimating the clean signal from the remaining signal subspace. So the music feature extraction is performed on the clean signal space. At the sampling rate of 22 kHz, the music signals are divided into 23ms frames with 50% overlapped hamming window at the two adjacent frames. Features computed from each frame are spectral centroid, spectral roll off, spectral flux, zero crossing rates, thirteen mel-frequency cepstral coefficients (MFCC) and ten linear predictive coefficients (LPC). The means and standard deviations of these six original features and their delta values are computed over each frame for each music file to form a total of 102-dimensional feature vector.

In order to reduce the computational burden and so speed up the search process, an efficient feature selection method is desired. As described in paper [9], a sequential forward selection (SFS) method is used to meet these needs. In this paper, we adopt the same SFS method for feature selection to reduce dimensionality of the features and to enhance the classification accuracy. Firstly, the best single feature is selected and then one feature is added at a time which in combination with the previously selected features to maximize the classification accuracy.

3.2 Multi-Feature Clustering

The classification results corresponding to different query patterns within the same music file may be much different. It may cause serious uncertainty of the system performance. In order to overcome these problems, a new robust feature extraction method called multi-feature clustering (MFC) with previous feature selection procedure is implemented. MFC extracts pre-defined features over the full-length music signal in a step of 20 sec large window and then cluster these features in four disjoint subsets (centroids) using LBG-VQ clustering technique.

4 Experimental Results

The proposed algorithm has been implemented and used to retrieve music data from a database of 240 music files. 60 music samples were collected for each of the four genres in Classical, Hiphop, Jazz, and Rock, resulting in 240 music files in database. The excerpts of the dataset were taken from radio, compact disks, and internet MP3 music files. The 240 music files are partitioned randomly into a training set of 168 (70%) sounds and a test set of 72 (30%) sounds.

Two sets of experiment have been conducted in this paper.

- Experiment 1: Retrieval performance for the proposed MIR system
- Experiment 2: MFC performance with different query patterns

Table 1 shows average retrieval accuracy of the system with noise reduction algorithm and MFC - SFS feature optimization method with respect to music query captured by cellular phone. As seen on the table 1, the proposed method achieves more than 20% higher accuracy than the one without noise reduction and MFC-SFS algorithm even with less number of feature set.

Table 1. MIR statistics comparison

	Noisy Query	Query processing with NR and MFC-SFS
Retrieval accuracy	45%	65%
Feature dimension	102	20

As pointed out earlier, the music retrieval results corresponding to different query patterns (or portions) may be much different. It may cause serious uncertainty of the system performance. In order to overcome this problem, MFC-SFS is used as explained in section 2. To verify the performance of the proposed method, seven excerpts with fixed duration of 5 sec were extracted from every other position in same query music- at music beginning and 10%, 20%, 30%, 40%, 50%, and 80% position after the beginning of music signal. Fig. 3 shows the retrieval results with seven excerpts at the prescribed query position.

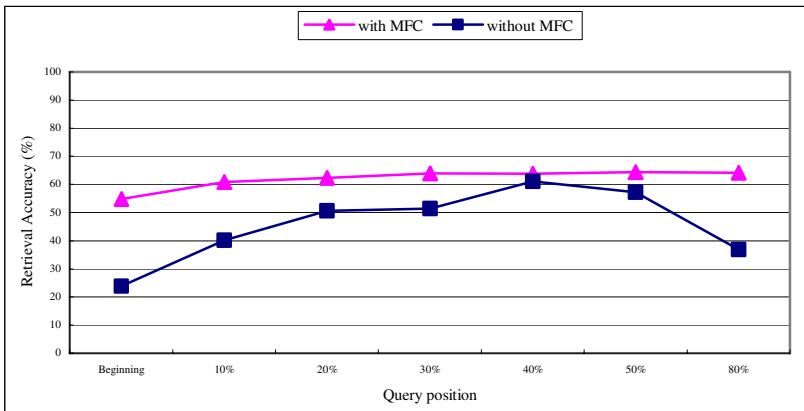


Fig. 2. Retrieval results at different query portions with MFC-SFS

As we expected, the retrieval results without MFC-SFS greatly depends on the query positions and it's performance is getting worse as query portion towards to two extreme cases of beginning and ending position of the music signal. This is no wonder because, in general, the musical characteristics are not rich enough at those extreme

intervals of music signal. On the other hand, we can find quite stable retrieval performance with MFC-SFS method and it yields relatively higher accuracy rate in the range of 55% ~ 67%. Even at two extreme cases of beginning and ending position, the system with MFC-SFSS can achieve high classification accuracy which is more than 20% improvement over the system without MFC-SFS.

5 Conclusion

In this paper, we propose music information retrieval system on mobile network. A query music signal is captured by a cellular phone in real world. A signal subspace noise reduction algorithm is applied to alleviate mobile noises. Then a robust feature extraction method called Multi-Feature Clustering combined with SFS feature selection is implemented to improve and stabilize the system performance. The proposed system has been tested with using cellular phones in the real mobile environment and it shows about 65 % of average retrieving success rate. Experimental comparisons for music retrieval with several query excerpts from every other position are presented and it demonstrates the superiority of MFC-SFS method in terms of the retrieval stability and accuracy. Future work will involve the development of new features and further analysis of retrieval system for practical implementation.

References

1. G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 5, pp. 293-302, July 2002.
2. E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio," *IEEE Multimedia*, vol.3, no. 2, 1996.
3. J. Foote, "Content-based retrieval of music and audio," in *Proc. SPIE Multimedia Storage Archiving Systems II*, vol. 3229, C.C.J. Kuo *et al.*, Eds., 1997, pp. 138-147.
4. 4 Burges, C.J.C, Platt, J.C., and Jana, S., "Extracting noise robust features from audio data", in Proceedings of ICASSP, pp. 1021-1024, 2002.
5. T.Kurozumi, K.Kashino, H.Murase, "A robust audio searching method for cellular-phone-based music information retrieval", IAPR 16thICPR, Vol.3, pp.991-994, 2002.
6. J. Haitsma and T. Kalker, "A Highly Robust Audio. Fingerprinting System", 3rd Int. Symposium on Music. Information Retrieval (ISMIR), Oct. 2002.
7. Shazam website <http://www.sahzamentertainment.com>
8. Y. Ephraim. "A signal subspace approach for speech enhancement" *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 251-266. July, 1995
9. M. Liu and C. Wan, "A study on content-based classification retrieval of audio database," *Proc. of the International Database Engineering & Applications Symposium*, pp. 339 - 345. 2001.

Joint Cluster Based Co-clustering for Clustering Ensembles

Tianming Hu¹, Liping Liu¹, Chao Qu¹, and Sam Yuan Sung²

¹ Department of Computer Science, DongGuan University of Technology
DongGuan, 523808, China
tmhu05@gmail.com

² Department of Computer Science, South Texas University
McAllen, Texas 78501, USA

Abstract. This paper introduces a new method for solving clustering ensembles, that is, combining multiple clusterings over a common dataset into a final better one. The ensemble is reduced to a graph that simultaneously models as vertices the original clusters in the ensemble and the joint clusters derived from them. Only edges linking vertices from different types are considered. The resulting graph can be partitioned efficiently to produce the final clustering. Finally, the proposed method is evaluated against two graph formulations commonly used.

1 Introduction

Data clustering is a difficult problem. It is ill-posed if no prior information is provided about the well-defined underlying data distributions. Over the same dataset, different clustering algorithms produce different partitions, which may capture various distinct aspects of the data. A natural question arises if we can combine the strengths of many individual clusterings to produce a final better one. This is the focus of research on clustering ensembles [1,2,3]. In particular, clustering ensembles are expected to be more robust, which means they must have better average performance across the domains and datasets, and have lower sensitivity to noise and outliers. They must be able to find a combined solution unattainable by any single clustering algorithm. Other quality measures include parallelization and scalability, etc.

Clustering can be regarded as an unsupervised classification problem. For supervised classification, there is an extensive body of work on combining multiple classifiers [4,5], such as bagging and boosting. In the case of clustering, however, the true clustering is unknown and the data are unlabeled. There is no explicit correspondence between the labels delivered by different clusterings. To make matters worse, different partitions may contain different numbers of clusters, which often makes the label mapping intractable. The combination of multiple partitions can also be viewed as finding a median partition with respect to the given partitions, which is proven to be NP-complete [6].

Clustering ensembles generally divides into two steps. The first step is to derive component partitions in the ensemble. Diversity of the individual clusterings can

be achieved by a number of approaches, including using different conventional algorithms [7], their relaxed versions [8], built-in randomness [1], or by data sampling [9].

This paper focuses on the second step, combining the components into the final one, which is often referred to as the consensus function. Such a process is also called consensus clustering, since the combined one is expected to be most compatible to the ensemble as a whole. An underlying assumption is that if the ensemble is ‘good’ enough, closeness to the ensemble means closeness to the true clustering. In our method, the ensemble is reduced to a graph that simultaneously models as vertices the original clusters in the ensemble and the joint clusters derived from them. Only edges linking vertices from different types are considered. The resulting graph can be partitioned efficiently to produce the final clustering.

1.1 Problem Formulation

Given a clustering ensemble, we will assume that the cluster labels are nominal values, i.e. every component is a hard partition. In general, however, the clusterings can be soft, with real values indicating the degree of pattern membership in each cluster in a partition. Note that our method introduced below can be easily generalized to the soft case with slight modification.

First we formulate the consensus function f . Suppose we are given a set of N objects $X = \{x_i\}_{i=1}^N$ and a set of M clusterings $\Phi = \{C^m\}_{m=1}^M$. Each clustering C^m groups X into K^m disjoint clusters, represented as $C^m = \{C_1^m, \dots, C_{K^m}^m\}$. That is, $C_i^m \cap C_j^m = \emptyset, i \neq j$, and $\cup_{i=1}^{K^m} C_i^m = X$. Generally speaking, the values of K^m for different clusterings may be either the same or different. The job of the consensus function is to find a new partition $C^* = f(\Phi)$ of X that summarizes the information from the gathered partitions Φ . In other words, C^* is expected to be as compatible as possible to the ensemble as a whole. Our main goal is to construct a consensus partition without the assistance of the original patterns in X , but only from their cluster labels.

The rest of the paper is organized as follows. Related work is reviewed in Section 2 and two graph formulations are introduced in Section 3. Our method is presented in Section 4. Empirical results are reported in Section 5 and concluding remarks are given in Section 6.

2 Related Work

Preliminary theoretical work on consensus clustering was given in [10], where it was proved that using certain generation models for the components in the ensemble, the consensus solution will converge to a true underlying clustering as the number of partitions increases. Practical approaches can be classified according to the two main constituent steps, the way the components are generated and the way they are combined.

As for the first step, approaches can be classified based on how many attributes and instances are used in components. Although most approaches utilize all

attributes in order to produce good components, in some cases, due to reasons like privacy or sheer size, only a subset of attributes are available for each component. The usefulness of having multiple views of data for better clustering was addressed in [11]. Besides, empirical studies in [8] showed that on certain datasets, the combined results from K-means partitions in the full space were beaten by those from partitions in projected 1-D subspaces.

Based on whether all instances are used in components, approaches also come in three categories: all instances, bagging and boosting. Although most approaches utilize all instances, some employ sampling techniques to introduce more diversity between components for improved robustness and accuracy of the final results [9]. Weighted sampling was studied in [12], where weights depends on the consistency of its previous assignments in the ensemble.

As for the second step, the simplest one is voting [9], provided that all components can be relabeled according to a reference partition with the target number of clusters. Otherwise, an underlying principle is often assumed. That is, the similarity between objects is proportional to the fraction of components that assign them together. Approaches differ in the way how this similarity is represented (i.e. how to summarize the ensemble) and in the way the principle is implemented. One can compute the co-association values for every pair of objects and feed them into any reasonable similarity based partitioned algorithms, such as hierarchical clustering [1] and graph partitioning [7]. In fact, a clustering ensemble directly provides a new set of features (i.e. cluster labels) describing the instances. The problem can be transformed to clustering this set of categorical vectors, e.g. using the EM/K-means algorithm to maximize likelihood [13]/generalized mutual information [8,14]. Another way is to represent each cluster by a hyperedge in a hypergraph where the nodes correspond to objects. The final clustering can be obtained using hypergraph partitioning algorithms like HMETIS [7,15].

3 Graph Partitioning Based Algorithms

Because we summarize the clustering ensemble in a graph and partition it to yield the final clustering, we introduce briefly graph partitioning first.

A weighted graph $G = (V, E)$ consists of a vertex set V and an edge set $E \subseteq V \times V$. All edge weights can be stored in a nonnegative symmetric $|V| \times |V|$ matrix W , with entry $W(i, j)$ describing the weight of the edge linking vertices i and j . Given a weighted graph G and a prespecified number K , the job is to partition the graph into K parts, namely, K disjoint clusters of vertices. The edges linking vertices in different parts are cut. The general goal is to minimize the sum of the weights of those cut edges. To avoid trivial partitions, the constraint is imposed that each part should contain roughly the same number of vertices. In practice, different optimization criteria have been defined, such as the normalized cut criterion [16] and the ratio cut criterion [17].

In this paper, METIS [18], a multilevel graph partitioning algorithm, is employed for its robustness and scalability. From a different angle, it partitions a

graph in three basic steps. First it recursively coarsens the graph to reduce its size by collapsing vertices and edges. During coarsening, METIS employs algorithms that make it easier to find a high-quality partition at the coarsest graph. Then it partitions the smaller graph. Finally it recursively refines the partitions, focusing primarily on the portion of the graph that is close to the partition boundary. Compared to other algorithms, METIS is highly efficient with quasi-linear computational complexity. The partitions produced by METIS are consistently better than those produced by spectral partitioning algorithms in various domains including finite element methods, linear programming and VLSI [19].

Cluster-based Similarity Partitioning Algorithm (CSPA) and Meta-CLustering Algorithm (MCLA) are two graph partitioning based algorithms for combining multiple partitions [7]. The former use vertices to represent the original objects, while the latter use them to represent clusters in the ensemble. Because we compare our method against these two algorithms, they are introduced in some detail below.

CSPA first computes an $N \times N$ co-association(similarity) matrix W , with entry (i, j) denoting the fraction of component clusterings in which the two objects i and j are in the same cluster. That is, given an ensemble $\{C^m\}_{m=1}^M$, $W(i, j) = \frac{1}{M} \sum_{m=1}^M I(C^m(i) = C^m(j))$, where $I(\cdot)$ is the indicator function and $C^m(k)$ denotes the cluster in C^m to which object k is assigned. This similarity matrix is directly fed into METIS (vertex = object, edge weight = similarity) to produce a final clustering. CSPA is the simplest and most heuristic, but its computational complexity is $O(N^2 \sum_{m=1}^M K_m)$, since it needs to compute an $N \times N$ similarity matrix,

MCLA first groups the original clusters in the ensemble using METIS, where every cluster is represented by a vertex. The similarity between two clusters C_i^m and C_j^n is computed using the Jaccard measure: $|C_i^m \cap C_j^n| / |C_i^m \cup C_j^n|$. Each resulting cluster, called meta-cluster, has an association value for each object describing its level of association between them. It is defined as the number of original clusters in the meta-cluster to which the object is assigned. Finally the final clustering is obtained by assigning each object to the meta-cluster with the largest association value. Its complexity is $O(N(\sum_{m=1}^M K_m)^2)$, since it needs to compute a $\sum_{m=1}^M K_m \times \sum_{m=1}^M K_m$ similarity matrix. In practice, MCLA tends to be best in low noise/diversity settings, because MCLA assumes that there are meaningful cluster correspondences, which is more likely to be true when there is little noise and less diversity.

4 Joint Cluster Based Co-clustering

In this section, we present the Joint cluster based Co-clustering (JC) algorithm. As shown below, it consists of three steps: forming joint clusters, generating the graph and partitioning the graph.

– Input

Φ : An ensemble of M component clusterings.

K : Desired number of clusters.

- Output

C^* : A combined clustering that summarizes Φ .

- Steps

1. Form joint clusters

Every joint cluster is the intersection of M clusters from M components. Every object belongs to exactly one joint cluster.

2. Generate the graph

Every node is either for a joint cluster or an original cluster in the ensemble. Every edge links a joint cluster to an original cluster where it belongs.

3. Partition the graph

Employ graph partitioning algorithms, e.g., METIS, to partition the graph into K parts. Label original objects with the cluster label assigned to its joint cluster.

Next we give details of each step.

4.1 Joint Clusters

Obviously, every clustering C can be uniquely mapped to a discrete random variable with $Pr(C = k)$ interpreted as the fraction of data in cluster k . Analogous to the joint distribution, given a clustering ensemble $\Phi = \{C^m\}_{m=1}^M$, we can construct a joint clustering that comprises $\prod_{m=1}^M K_m$ joint-clusters. Every joint cluster can be represented by a vector $(C_{k_1}^1, \dots, C_{k_M}^M)$, i.e. the intersection of clusters $C_{k_1}^1, \dots, C_{k_M}^M$. These joint clusters are mutually exclusive and exhaustively collective, which means the joint clustering can map to a corresponding joint distribution, with the original data set as the sample space, and the corresponding component distributions as the marginal distributions. Each such joint cluster is a maximal group of objects that are completely contained by a cluster in every component, which means they are indistinguishable in the eyes of the ensemble.

The above joint cluster using all M components can be referred to as joint cluster of order M (or full order). The original cluster in the ensemble can be called joint cluster of order 1. Similarly, joint clusters of other orders can be defined.

There is a corresponding relation to the object representation used in [8,13], where objects are represented using cluster labels in the ensemble and subsequent clustering is done by K-means/EM on those categorical vectors. That is, all objects in the joint cluster receive the same representation and they are no longer distinguishable.

4.2 Graph Generation

To avoid quadratic size N^2 , we choose to cluster at the resolution of joint clusters. At this time, a natural method is to represent each joint cluster with a vertex and use METIS to partition the full connected graph. Finally each object can be uniquely assigned to the cluster where its joint cluster belongs. In this case the

similarity matrix size is quadratic in the joint clustering size $\prod_{m=1}^M K_m$. To avoid this quadratic size, we introduce the original cluster to the graph and only edges linking clusters at different levels are allowed. METIS is employed to partition the graph, that is, it simultaneously cluster original clusters and joint clusters. The final clustering is obtained by assigning each object to the cluster where its joint cluster belongs.

Formally, our method constructs a graph $G = (V, E)$. $V = V^1 \cup V^M$, where V^1 contains $\sum_{m=1}^M K_m$ vertices each representing a cluster in the ensemble (joint cluster of order 1), and V^M contains $\prod_{m=1}^M K_m$ vertices each representing a joint cluster (of order M). Edge set E only contains edges linking vertices (clusters) of different orders. So the graph is a bipartite actually. Bipartite graphs have been used to model various relationships between different types of entities, such as (words, documents) [20] and (authors, publications) [21]. They have also been employed to describe relevance between the same type of entities, e.g., sentences from two news articles for correlated summarization [22].

For convenience, if there is no edge linking two vertices, we assume the edge weight is 0. The weight matrix W is defined as follows. If the vertices i and j are clusters of the same order, $W(i, j) = 0$. Otherwise, if joint cluster i is contained in the original cluster j , $W(i, j) = W(j, i) = |i|(|i|$ denotes the size of joint cluster i), 0 otherwise. Hence W is in the form $\begin{pmatrix} 0, S \\ S^T, 0 \end{pmatrix}$. Positive values

only appear in the $\prod_{m=1}^M K_m \times \sum_{m=1}^M K_m$ sub-matrix S , where each row is for a joint cluster and each column is for an original cluster. Because of its special structure, the real size of the graph partitioning problem is the size of S , which is significantly smaller than the size N^2 of CSPA, assuming that $\prod_{m=1}^M K_m \ll N$ and $\sum_{m=1}^M K_m \ll N$. Note that if components in the ensemble are similar, the number of non-empty joint clusters will be far less than $\prod_{m=1}^M K_m$, which reduces the graph size further.

All joint cluster sizes can be determined in $O(NM)$ by one scan of the dataset. Therefore our method's complexity is $O(NM + \prod_{m=1}^M K_m \times \sum_{m=1}^M K_m)$.

An illustrative example is given in Fig. 1. Eight objects indexed with 1, ..., 8 are grouped in three clusterings, which yield five non-empty joint clusters.

5 Experimental Evaluation

5.1 Evaluation Criteria

Because the true class labels in our experiments are known, we can measure the quality of the clustering solutions using external criteria that measure the discrepancy between the structure defined by a clustering and what is defined by the class labels. Many metrics have been proposed in the literature to compare two partitions of a data set, such as Rand index, Jaccard coefficient, Fowlkes and Mallows index, and Hubert's Γ , all of which were discussed in [23].

We choose three measures: normalized mutual information (NMI)[7], conditional entropy(CE) and error rate(ERR). As we introduced before, the cluster

$$\begin{aligned}
 C^1 : C_1^1 = \{1,2,3\}, C_2^1 = \{4,5\}, C_3^1 = \{6,7,8\} & & jc_1 : (C_1^1, C_1^2, C_1^3) = \{1,2,3\} \\
 C^2 : C_1^2 = \{1,2,3,4\}, C_2^2 = \{5,6,7,8\} & & jc_2 : (C_2^1, C_2^2, C_2^3) = \{4\} \\
 C^3 : C_1^3 = \{1,2,3\}, C_2^3 = \{4,5,6\}, C_3^3 = \{7,8\} & & jc_3 : (C_2^1, C_2^2, C_2^3) = \{5\} \\
 & & jc_4 : (C_3^1, C_2^2, C_2^3) = \{6\} \\
 & & jc_5 : (C_3^1, C_2^2, C_3^3) = \{7,8\}
 \end{aligned}$$

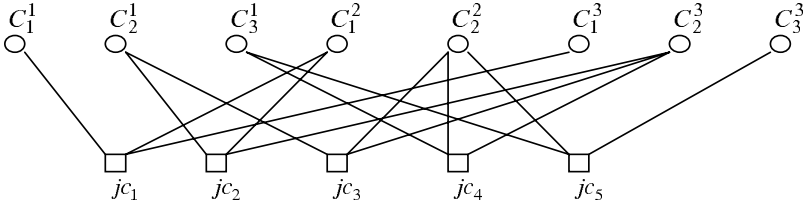


Fig. 1. The ensemble contains three clusterings over eight objects. The graph models as vertices the eight original clusters and the five joint clusters.

label can be regarded as a random variable with the probability interpreted as the fraction of data in that cluster. Let T and C denote the random variables corresponding to the true class and the cluster label, respectively. The two entropy-based measures are computed as follows:

$$\begin{aligned}
 NMI &= \frac{H(T) + H(C) - H(T, C)}{\sqrt{H(T)H(C)}} \\
 CE &= H(T|C) \\
 &= H(T, C) - H(C)
 \end{aligned}$$

where $H(X)$ denotes the entropy of X . NMI measures the shared information between T and C . It reaches its maximal value of 1 when the clustering is the same as the true classification. It is minimized to 0 when they are independent. CE tells the information remained in T after knowing C . It reaches its minimal value of 0 when T and C 's distributions are the same. It is not normalized and is upper bounded by $H(T)$. Error rate $ERR(T|C)$ just computes the fraction of misclassified data when all data in every cluster of C is classified as the majority class in that cluster. It can be regarded as a simplified version of $H(T|C)$. Note that CE and ERR are biased towards clusterings with a large number of clusters, since both are minimized to 0 when C only contains singleton clusters. However, it is not a problem in our experiments later, for the target number of clusters used for the consensus functions are all set equal to the true number of classes.

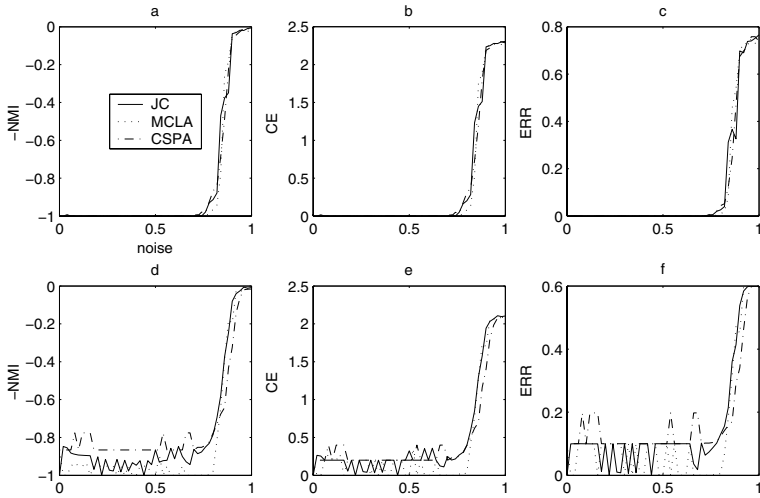


Fig. 2. The first row shows the comparison when the true clustering is balanced. The second row is for the imbalanced case.

5.2 Random Relabeling

Following [7], we devise a set of experiments where components in the ensemble are derived from a hypothetical true clustering with cluster labels $1, \dots, K = 5$ via random relabeling. In detail, at each noise level $\epsilon \in [0, 1]$, a fraction ϵ of data are randomly chosen. Their true cluster labels are replaced with random values from the uniform distribution from $1, \dots, h(K)$. Such process is repeated 100 times to produce of an ensemble of size 100.

Two true clusterings are tried. One is balanced with 100 data in each of five clusters and we set $h(K) = K$. The other one is imbalanced. Its five clusters are of size 50, 100, 200, 50, 100, respectively, and the degree of balance is $(\text{avg cluster size}) / (\text{max cluster size}) = 0.5$. We set $h(K) = 2K$ this time, i.e. the number of clusters in each component is also a random number from $K, \dots, 2K$. Such setting introduces more diversity between components. In practice, we seldom know the true number of clusters and we need to try a range of values, in the hope that the true number is included.

The results are plotted in Fig. 2 with 51 noise levels at 0.02 intervals. For compatibility, $-NMI$ is recorded so that all three measures are to be minimized. From the first row where the true clustering is balanced, one can see that till $\epsilon = 0.7$, all three methods keeps yielding the true clustering. Mixed results appear after $\epsilon > 0.7$. Note that balanced true clusterings favor CSPA most, for METIS tries to output balanced clusterings and each vertex corresponds an object in CSPA. This explains why CSPA performs worst at $\epsilon < 0.7$ in the second row of Fig. 2, where the true clustering is imbalanced. MCLA works best and still yields the true clustering at noise level up to 0.7, as in the balanced case. The ensemble in this case is of high noise and strong correspondence. That is,

Table 1. Dataset summary

dataset	iris	glass	heart	image	satimage
#instance	150	214	303	2100	4435
#class	3	6	6	7	6
#dim	4	9	13	19	36
balance	1	0.47	0.31	1	0.69

because of large ensemble size, the Jaccard measure for the two clusters with the same cluster label may be far smaller than 1, but it is still comparatively higher than the measure for the two clusters with different cluster labels. The same argument holds for the meta-cluster’s association values for the objects. JC’s performance is between CSPA and MCLA. When $\epsilon > 0.7$, CSPA takes the lead.

5.3 Random Subspace

For the real data when the ensemble is unavailable, first we need to generate different clusterings for the combination. Of course we can use numerous existing sophisticated clustering algorithms, but that would make consensus clustering less justified. If very good clusterings already have been obtained, why bother to combine them? Consensus clustering is most useful if we could generate the partitions using weak but less expensive clustering algorithms and still achieve comparable or better performance. The key motivation is that the synergy of many such components will compensate for their weaknesses.

We consider clustering of the data projected to a random subspace of lower dimension. In the simplest case, the data is projected on 1-D subspace. The K-means algorithm clusters the projected data and gives a partition for the combination. Each random 1-D subspace is by itself not very informative, but clustering in 1-D subspace is computationally cheap and can be effectively performed by the K-means algorithm. The main subroutine of the K-means algorithm, distance computation, becomes d times faster in 1-D space, compared to the original d -D space.

Summarized in Table 1, five labeled datasets from the UCI repository [24] are chosen: iris, glass, Cleveland heart, image and satimage. The ensemble size is set to 100. Each component is generated by (1) randomly choosing a dimension, and (2) applying K-means in that 1-D space with a random target number of clusters from $K, \dots, 2K$, where K is the true number of classes. In practice, a random number of clusters in each partitions ensures a greater diversity of components. Besides, it is necessary to set it higher than the true number of classes when some classes are multi-modal.

The average results of 20 runs are reported in Table 2, together with the average results of components in the ensemble. In general, CSPA achieves the best results in terms of all three measures. JC comes next and MCLA is the worst. The only exception is on iris, where JC is slightly worse than MCLA. The lead of CSPA over JC is very slight on the last three datasets. On image, both CSPA and JC achieve significantly better results than MCLA, whose variance

Table 2. Comparison on real datasets

		-NMI	CE	ERR
iris	JC	-0.7586 ± 0.0373	0.3837 ± 0.0589	0.0940 ± 0.0212
	MCLA	-0.7685 ± 0.0356	0.3687 ± 0.0564	0.0890 ± 0.0203
	CSPA	-0.8325 ± 0.0489	0.2655 ± 0.0775	0.0550 ± 0.0276
	ensemble	-0.5172	0.6705	0.2236
glass	JC	-0.2626 ± 0.0366	1.5902 ± 0.0809	0.4353 ± 0.0314
	MCLA	-0.2594 ± 0.0491	1.6423 ± 0.0905	0.4666 ± 0.0423
	CSPA	-0.3194 ± 0.0330	1.4190 ± 0.0784	0.4042 ± 0.0287
	ensemble	-0.2156	1.6889	0.5052
heart	JC	-0.1952 ± 0.0266	1.4839 ± 0.0477	0.4012 ± 0.0154
	MCLA	-0.1143 ± 0.0647	1.6785 ± 0.0963	0.4488 ± 0.0118
	CSPA	-0.1980 ± 0.0183	1.4359 ± 0.0378	0.4069 ± 0.0132
	ensemble	-0.0867	1.6961	0.4490
image	JC	-0.5555 ± 0.0306	1.2488 ± 0.0860	0.3788 ± 0.0223
	MCLA	-0.4313 ± 0.1332	1.6955 ± 0.4561	0.5141 ± 0.1365
	CSPA	-0.5778 ± 0.0308	1.1851 ± 0.0863	0.3740 ± 0.0223
	ensemble	-0.2866	1.9587	0.5828
satimage	JC	-0.3905 ± 0.0126	1.4860 ± 0.0319	0.4298 ± 0.0149
	MCLA	-0.3864 ± 0.0156	1.5024 ± 0.0380	0.4451 ± 0.0164
	CSPA	-0.3885 ± 0.0258	1.4910 ± 0.0653	0.4109 ± 0.0293
	ensemble	-0.3113	1.6347	0.4677

is also the largest. Compared to the component clusterings, the considerable improvement is achieved by the combined clustering on iris and image. Note that the average quality of components on image is not the best. Actually it is the worst in terms of CE and ERR.

6 Conclusion

A new method using graph partitioning was presented for solving the problem of clustering ensembles. Both the original clusters in the ensemble and the joint clusters derived from them are modeled as vertices. Only edges linking vertices from different types are allowed. Thus the resulting graph can be partitioned efficiently to produce the final clustering.

The proposed method was evaluated against CSPA and MCLA, two commonly used methods based on graph partitioning. Empirical studies showed that its performance is between CSPA and MCLA. Its complexity is also between them. Although CSPA generally gave best results, quadratic complexity effectively prohibits CSPA from being used on large data sets. Despite the linear complexity, MCLA cannot work well when there is no cluster correspondence, due to great diversity or poor quality of components. Therefore, in the case of large data size and great diversity between components, our method is a competitive alternative for the problem of clustering ensembles.

References

1. Fred, A., Jain, A.K.: Combining multiple clustering using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(6) (2005) 835–850
2. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(12) (2005) 1866–1881
3. Zhou, Z.H., Tang, W.: Clusterer ensemble. *Knowledge-Based Systems* **19**(1) (2006) 77–83
4. Dietterich, T.G.: Ensemble methods in machine learning. In: *Proceedings of the 2nd International Workshop on Multiple Classifier Systems*. (2001) 1–15
5. Ghosh, J.: Multiclassifier systems: Back to the future. In: *Proceedings of the 3rd International Workshop on Multiple Classifier Systems*. (2002) 1–15
6. Barthelemy, J.P., Leclerc, B.: The median procedure for partition. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **19** (1995) 3–33
7. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* **3** (2002) 583–617
8. Topchy, A., Jain, A.K., Punch, W.: Combining multiple weak clusterings. In: *Proceedings of the IEEE International Conference on Data Mining*. (2003) 331–338
9. Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. *Bioinformatics* **19**(9) (2003) 1090–1099
10. Topchy, A., Law, M.H., Jain, A.K., Fred, A.: Analysis of consensus partition in cluster ensemble. In: *Proceedings of The 4th IEEE International Conference on Data Mining*. (2004) 225–232
11. Kargupta, H., Huang, W., Johnson, E.: Distributed clustering using collective principal component analysis. *Knowledge and Information Systems Journal* **3** (2001) 422–448
12. Topchy, A., Minaei, B., Jain, A., Punch, W.: Adaptive clustering ensembles. In: *Proceedings of the International Conference on Pattern Recognition*. (2004) 272–275
13. Topchy, A., Jain, A.K., Punch, W.: A mixture model of clustering ensembles. In: *Proceedings of the 4th SIAM International Conference on Data Mining*. (2004)
14. Mirkin, B.: Reinterpreting the category utility function. *Machine Learning* **45**(2) (2001) 219–228
15. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: Applications in VLSI domain. In: *Proceedings of the 34th Conference on Design Automation*. (1997) 526–529
16. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000) 888–905
17. Hagen, L., Kahng, A.: New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on CAD* **11** (1992) 1074–1085
18. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* **20**(1) (1998) 359–392
19. Barnard, S.T., Simon, H.D.: A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. In: *Proceedings of the 6th SIAM Conference on Parallel Processing for Scientific Computing*. (1993) 711–718
20. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. (2001) 269–274

21. Sun, J., Qu, H., Chakrabarti, D., Faloutsos, C.: Relevance search and anomaly detection in bipartite graphs. *ACM SIGKDD Explorations* **7**(2) (2005) 48–55
22. Zhang, Y., Chu, C.H., Ji, X., Zha, H.: Correlating summarization of multisource news with k-way graph bi-clustering. *ACM SIGKDD Explorations* **6**(2) (2004) 34–42
23. Jain, A., Dubes, R.: *Algorithms for Clustering Data*. Prentice Hall (1988)
24. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI repository of machine learning databases. <http://www.ics.uci.edu/~mllearn/MLRepository.html> (1998)

Mining Gait Pattern for Clinical Locomotion Diagnosis Based on Clustering Techniques

Guandong Xu¹, Yanchun Zhang¹, and Rezaul Begg²

¹ School of Computer Science and Mathematics
Victoria University, PO Box 14428, VIC 8001, Australia
{xu, yzhang}@csm.vu.edu.au

² Centre for Ageing, Rehabilitation, Exercise & Sports
Victoria University, PO Box 14428, VIC 8001, Australia
Rezaul.Begg@vu.edu.au

Abstract. Scientific gait (walking) analysis provides valuable information about an individual's locomotion function, in turn, to assist clinical diagnosis and prevention, such as assessing treatment for patients with impaired postural control and detecting risk of falls in elderly population. While several artificial intelligence (AI) paradigms are addressed for gait analysis, they usually utilize supervised techniques where subject groups are defined *a priori*. In this paper, we explore to investigate gait pattern mining with clustering-based approaches, in which *k*-means and hierarchical clustering algorithms are employed to derive gait pattern. After feature selection and data preparation, we conduct clustering on the constructed gait data model to build up pattern-based clusters. The centroids of clusters are then treated as the subject profiles to model the various kinds of gait pattern, e.g. normal or pathological. Experiments are undertaken to visualize the derived subject clusters, evaluate the quality of clustering paradigm in terms of silhouette and mean square error and compare the results with the discovery derived from hierarchy tree analysis. In addition, analysis conducted on test data demonstrates the usability of the proposed paradigm in clinical applications.

1 Introduction

Gait pattern analyses have been addressed to reveal kinematic, kinetic and electromyographic (EMG) gait characteristic for modeling human walking [1-3]. The discovery of gait pattern will help to identify any change of gait that reflects the gait degeneration due to pathological reasons. One of its applications is to monitor the ageing influence on gait pattern, which causes constant threats to elderly population and help prevent them from potential risk of falls. There are various types of gait variables that are used to describe and analyze gait. However, basic gait variables (e.g. walking speed, stride length, cadence, leg length etc), are frequently employed in modeling human walking [4]. As a result of gait pattern analysis, there is a demand to identify the related subsets of these gait parameters as feature vector (i.e. gait data model) and employ applicable statistical analysis tools on the derived data model to reveal the underlying pattern of gait hidden in the gait data.

Related work: To characterize gait pattern and differentiate the normal from the pathological, some pattern recognition and machine learning techniques have been used to address this problem. Some academics utilize supervised approaches [5, 6], in which groups of subjects are defined *a priori*, to model the quantitative correlation among them by using discriminant analysis paradigm, while others exploit descriptive or subjective techniques to build up collections of subjects [7, 8]. Alternatively, unsupervised (or weak supervised) techniques are also considered as effective paradigms for gait pattern recognition. [9] proposed a fuzzy clustering technique employed on temporal-distance parameters to group normal and pathological gait subjects into various clusters accordingly. Neural network (*NN*) and Support Vector Machine (*SVM*), two types of well-studied machine learning approaches recently are used for identification of the at-risk gait in the elderly population. The former adopts neural network model to classify various gait types, while the latter is to recognize gait patterns by finding an optimal separating hyperplane to separate two groups' data. For example, [10, 11] applied *NN* on the selected feature subset from lower-limb joint-angle measures to differentiate various gait pattern, whereas [12] exploited Minimum Foot Clearance (MFC) histogram-plot and Poincaré-plot images to train *SVM*, for automated recognition of gait pattern changes due to ageing. Results from their work have shown they are effective gait analysis tools for solving classification problems by learning gait data with satisfactory performance [13-15]. Furthermore, the differentiated subject groups will provide biomechanical insight and treatment assessment criteria for the population with pathological gait characteristics. However, most of the above research is mainly focused on the classification of the subject gait data into predefined subject groups rather than the discovery of underlying relationships among gait data that is used to derive gait patterns as well as identification of subject groups (i.e. gait patterns) from the gait parameters. In most case, it is crucial to address the issues of how to find a reasonable grouping scheme and then partition the subjects into the corresponding groups since it is hard to define *a priori* subject groups in real scenarios.

The purpose of the present study is to investigate the discovery of the underlying gait pattern from the viewpoint of data mining domain. Unlike previous work, we exploit clustering technique, one kind of unsupervised learning approach, to objectively differentiate the subjects with normal or pathological gait characteristics and build up pattern-oriented gait categories. Particularly, we explore to employ two kinds of clustering techniques, i.e. *k*-means and hierarchical clustering techniques, to discover the underlying correlation among human walking behaviors and identify the gait pattern exhibited from them. Then, the centroids of obtained clusters are treated as the subject profiles to represent the various types of gait patterns, which will provide an instrumental means to determine the likelihood of a specific subject's belonging to a cluster. Experiments are conducted on the real gait dataset to validate the proposed research, visualize the clustering results and assess the treatment effectiveness for those subjects with impaired postural controls. In addition, clustering evaluation and comparison are studied as well, to investigate the selection of initial cluster number in *k*-means algorithm and the changes caused by employing different clustering algorithms.

The rest of the paper is organized as follows: Section 2 briefly introduces the gait data model, data preparation and the principles of the involved clustering algorithms.

Experimental results, such as clustering visualization, evaluation and comparison are presented in Section 3. To further assess the discovered gait patterns, analysis on test data is carried out to demonstrate the shift of individual's affiliated gait clusters pre- and postoperatively, and propose the potentially promising clinical applications in Section 4. We summarize and outline the future direction in Section 5.

2 Gait Data Model and Data Preparation

In a biomechanical analysis, there are a variety of basic time-distance parameters that are frequently used for modeling human walking, such as walking speed, stance/swing times. This may be due to the fact that temporal-distance parameters are probably more fundamental for the purpose of gait analysis [16]. In this work, we simply exploit the specific two-dimensional temporal-distance parameters, i.e. stride length and step frequency/cadence to construct gait data model. Both normal and pathological (cerebral palsy) data relating to children's gait for developing the models were taken from [9]. In this model, the gait data is expressed as a two-dimensional feature vector matrix, in which each row represents a subject vector in terms of stride length and cadence parameters, whereas every column corresponds to one of the two-selected gait variable. In the following experiments, we will conduct data mining on the constructed gait data to reveal the individual-specific gait pattern. In addition to kinematic parameters, other two biomechanical features, i.e. leg length and age, are taken into consideration for normalizing and scaling preprocessing to eliminate the diversity in individuals.

2.1 Normalization and Scaling

To remove the relative difference within the gathering of data in subject's age and leg length and leave any pathological trends, a polynomial-based normalization technique [9] is employed on stride length and cadence parameters respectively:

$$NSL = SL - (a_0 + a_1LL + a_2(LL)^2 + \dots + a_k(LL)^k) + \overline{SL}_N \quad (1)$$

$$NCAD = CAD - (b_0 + b_1AGE + b_2(AGE)^2 + \dots + b_k(AGE)^k) + \overline{CAD}_N \quad (2)$$

where $NSL, SL, LL, NCAD, CAD, AGE$ are subject's (intact or pathological) normalized stride length, original stride length, leg length, normalized cadence, original cadence and actual age respectively, $\overline{SL}_N, \overline{CAD}_N$ stand for the average stride length of intact subjects and average cadence of intact subjects.

Since *Euclidean* distance is employed to measure the similarity between two subjects' gait characteristics, a scaling process on gait data is needed to have unity variance and decrease the influence of one feature dominating the distance over another feature with its significant value.

$$SNSL = C_{SL}NSL \quad (3)$$

$$SNCAD = C_{CAD}NCAD \quad (4)$$

where $SNSL$ and $SNCAD$ are subject's stride length and cadence after normalizing and scaling, C_{SL} and C_{CAD} are coefficients for stride length and cadence scaling.

2.2 Similarity Measurement

After data normalization and scaling transformation, the discrepancy not only in individual's physical condition (i.e. leg length and age), but also in the observation variance of stride length and cadence caused by one feature dominating another in value, will be removed. Therefore, one basic similarity metric, i.e. *Euclidean* distance that is well-adopted to measure the distance of two feature vectors in *Information Retrieval* [17], is utilized to measure the similarity of two subject since every gait data could be considered as a feature vector in this case.

$$sim(s_i, s_j) = d_2(s_i, s_j) = \sqrt{\sum_{t=1}^2 (s_{it} - s_{jt})^2} \quad (5)$$

where s_i is the i -th subject of gait data.

Moreover, since the centroid of subject cluster could be virtually viewed as a subject in the form of feature vector, the distance between the generated centroid and individual subject could be further expressed as the affiliation distance of this subject with the subject group.

$$AD(s_i, C_k) = d_2(s_i, cid_k) \quad (6)$$

In clustering stage, this kind of distance is calculated repeatedly until the mean distance is converging to a local optimal value.

2.3 Clustering Algorithms

Two types of clustering algorithms i.e. k -means and hierarchical clustering are conducted to group the gait data in terms of temple-distance parameter. In k -means clustering analysis, we investigated the implementation of grouping the ambulation of neurologically intact individuals and those with cerebral palsy (CP) into K subject categories, whose number would be preset in advance, the visualization of separation layout of grouped subject clusters and the evaluation of clustering quality in terms of mean silhouette and mean square error. The k -means clustering algorithm works as follows [18]:

- Step 1: arbitrarily choose K subjects as initial cluster mean centers;
- Step 2: then assign each subject to the cluster with the nearest centers, and update each mean center of cluster;
- Step 3: repeat step 2 until all centers don't change and no reassignment is needed;
- Step 4: finally output subject clusters and their corresponding centers.

In contrast to k -means clustering, hierarchical clustering is also conducted to reveal the possible grouping strategy for gait data of normal and pathological subjects from the view point of hierarchy tree analysis. Meanwhile, construction of hierarchy tree and its corresponding visualization layout of clusters in terms of centroids are plotted

as well for comparison of these two kinds of dominating algorithms. The procedure of hierarchy clustering is [18]:

- Step 1: calculate the mutual distance of paired subjects (distance matrix) as the clustering criteria;
- Step 2: decompose subject dataset into a set of levels of nested aggregation based on distance matrix (i.e. tree of clusters);
- Step 3: cut the hierarchical tree at the desired level by selecting a predefined threshold, and then explicitly merge all connected subjects below the cut level to create various clusters;
- Step 4: output the dendrogram and clusters.

3 Experiments and Results

3.1 Experimental Data and Design

The stride length and cadence data for the 68 normal children and the 88 children with CP are constructed as temple-distance gait data from previous work [9]. In order to accomplish normalization process described above, parameters of age and leg length are also taken into account.

Table 1. Normalization Coefficients and scaling factors for gait data

name	a_0	a_1	\overline{SL}	b_0	b_1	b_2	\overline{CAD}	C_{SL}	C_{CAD}
value	0.28	1.31	1.02	174.07	-7.04	0.22	136.84	5.88	0.034

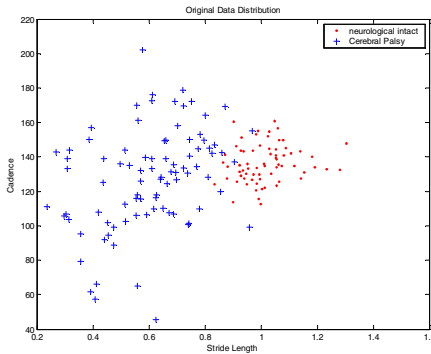


Fig. 1. Normalized gait data for children in normal and pathological group

Here, we utilized a first-order and second-order polynomial models for normalizing stride length and cadence respectively. The engaged coefficients are tabulated in Table 1 [9]. In addition, the scaling factors that are used to unify the amplitude in variance of stride length and cadence parameters are listed in table 1 as well. Figure 1 illustrates the normalized two-dimensional plot of gait data, i.e. stride length vs. cadence, for 68

neurological intact children and 88 children with CP. In this figure, red solid dot stands for the subjects in neurological intact group, whereas black cross symbol represents subjects with pathological symptoms. Consequently, our aim is to separate these two main types of subject into various groups, within which the elements should share the similar gait characteristics pattern. Especially, after clustering stage, the subjects in neurological intact group should be ideally categorized into the same cluster.

3.2 Experimental Results

3.2.1 K-Means Clustering

The clustering results with respect to K-means are visualized in Fig.2 (a)-(b) for $k = 5, 6$ respectively, where the grouped subjects are symbolized with a variety of point types and colors. In addition, the corresponding centroids of clusters are marked in the figures in black solid dots as well. From these plots, it is visually demonstrated which subjects are grouped together into the same cluster according to their mutual Euclidean distance, how close the subjects within same cluster keep and how far the subjects separate from others in different cluster. For example, the neurologically intact subjects are almost partitioned into the first cluster in blue square in case of $k = 5$, while for $k = 6$, such subjects are separated into two individual clusters, in which they are represented by blue square and cyan cross symbols accordingly.

Meanwhile, the centroids of clusters for $k = 5$ and $k = 6$ are tabulated in Tables 2-3 respectively, which are indicated by the point sequence number ranging from 1 to 6 in the cluster layout (i.e. Fig. 2 (a)-(b)). Furthermore, the created centroids are treated as the pattern-based gait profiles to represent the overall gait characteristics of corresponding gait groups [19].

3.2.2 Evaluation of Clustering

In order to evaluate the quality of clustering, we introduce two basic coefficients, namely silhouette coefficient and mean square error, in this paper.

Silhouette coefficient

Firstly, we compare clustering quality with respect to a variety of parameter settings of cluster number (i.e. K value). In order to be independent from the number of clusters produced, we use the silhouette coefficient for the purpose of evaluation [19]

The silhouette coefficient is an indicator to measure the quality of clustering, which is normally a value between 0 and 1, and rather independent from the number of clustering, k . Theoretically, the larger the value of SC is, the higher the quality of the cluster will be.

Table 2. Centroids of clusters with k -means when $k = 5$

Centroid #	Stride Length	Cadence
$P1$	0.7190	160.8955
$P2$	0.4557	80.6862
$P3$	0.3630	129.6800
$P4$	1.0102	136.0906
$P5$	0.6533	122.2424

Table 3. Centroids of clusters with k -means when $k = 6$

Centroid #	Stride Length	Cadence
$P1$	0.7024	163.4505
$P2$	0.6428	121.9912
$P3$	0.4557	80.6862
$P4$	0.3630	129.6800
$P5$	1.0811	141.4067
$P6$	0.9552	131.5741

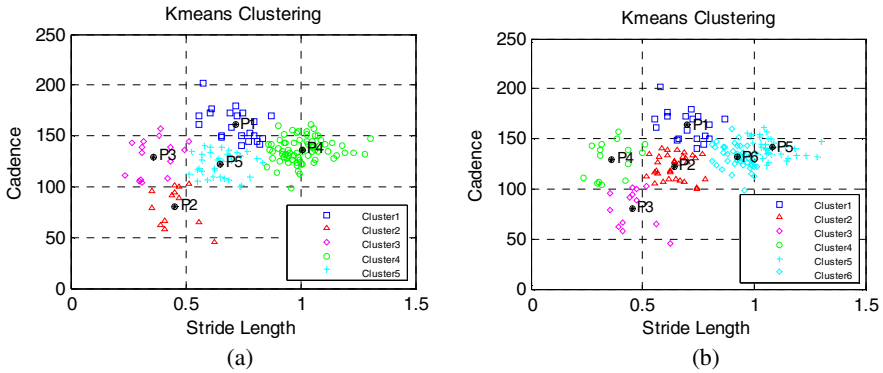


Fig. 2. Cluster visualization of k -means with $k = 5$ (a) and $k = 6$ (b)

Mean Square Error

In addition to silhouette coefficient, we also conducted further evaluation study on overall mean errors for clustering rather than on a single cluster quality, for the purpose of comparison [19].

It is easily concluded that the *MSE* stands for the overall mean distance for each subject within same cluster from its corresponding centroid, which reveals the quality of clustering as well.

Table 3 summarizes the calculated results in terms of *SC* and *MSE* for k -means clustering in case of $k = 5$ and $k = 6$. Interestingly, the table shows that the highest value for *SC* is for $k = 5$, $k = 4$ and $k = 6$ rank second and third, whereas the smallest mean square error occurs in $k = 6$ instead of $k = 5$. This is mainly due to the separation of neurological intact group into two individual sub-clusters, which will result in the decrease of distance from every subject in the sub-cluster to its centroid of sub-cluster.

Table 4. Mean silhouette and Mean Square Error for k -means with $k = 4, 5, 6$

K	4	5	6
<i>SC</i>	0.5981	0.6408	0.5510
<i>MSE</i>	134.95	97.42	72.26

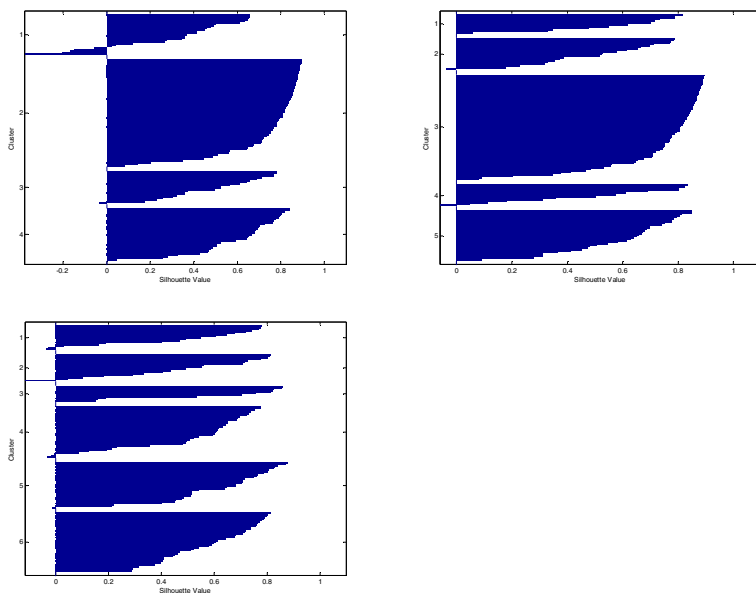


Fig. 3. Silhouette value plots of k -means clustering with $k = 4, 5, 6$

Figure 3 shows the silhouette plots of k -means clustering for various cluster number settings. From this plot, it is shown that there is one particular cluster that seems to be rather well-separated, which actually consists of neurological intact subjects, while others are not distinct enough for the three different k settings. In addition, the plots indicate that the clusters generated with $k=5$ exhibit a little bit higher quality than other two k settings, which is also validated by silhouette coefficient (SC) shown in Table 3. Especially, the negative values of SC reflect that the corresponding subjects are partitioned wrongly into inappropriate subject groups, according to its definition. Consequently, the more occurrence rate of negative SC reveals the poor quality of clustering correspondingly. From this viewpoint, it is demonstrated that the selection of cluster number with $k=5$ is much more appropriate than those of cluster number settings with $k=4, 6$. Conclusively, we will stick to selection of $k=5$ to conduct hierarchical clustering and test data validation in the following analysis.

3.3 Hierarchical Clustering

3.3.1 Dendrogram and Cluster Visualization

In comparison with k -means clustering, we also investigate the partition of gait data via hierarchy tree approach. Hierarchy clustering is to create a hierarchical tree of clusters based on the mutual distance between each pair of observations.

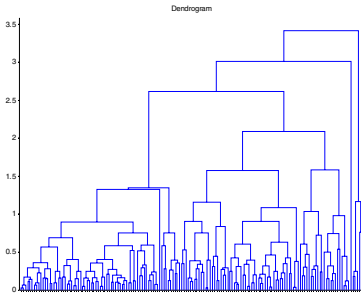


Fig. 4. Hierarchical cluster tree

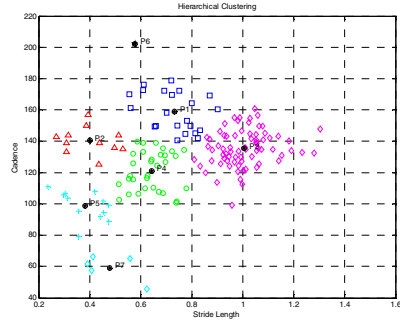


Fig. 5. Cluster visualization of hierarchy tree

Fig.4, 5 illustrate the plotted hierarchical tree of clusters in the form of Dendrogram and visualized cluster layout of hierarchy tree respectively. In Dendrogram, the x coordinate stands for the processed subject sequence number, whereas y coordinate conveys the distance information between two adjacent nodes. Interestingly, the first 68 x coordinates in dendrogram is exactly same as the subject order in the original gait dataset, which will result in the production of first big subject cluster. However, the figure indicates further that there exist several oddish subjects, which are far enough from other objects so that they could not be grouped into either established cluster in the lower level.

Table 5. Centroids of clusters with hierarchy clustering

Centroid #	Stride Length	Cadence
$P1$	0.7338	158.99
$P2$	0.4002	140.48
$P3$	1.0065	135.49
$P4$	0.6424	120.97
$P5$	0.3821	98.82
$P6$	0.5781	202.38
$P7$	0.4789	59.14

4 Test Data and Pattern Assessment

4.1 Test Dataset

To validate and assess the clustered gait pattern, two sets of subjects are introduced in this work. The test dataset comprises of 6 subjects shown in Table 5 [20], which represents the gait information with respect to two patient with CP at three different age times.

4.2 Affiliated Probability

With the discovered gait pattern, our aim is to investigate, for each patient, the affiliated probability (AP) distribution over the set of gait clusters, then identify which gait

Table 6. Testing gait data of two subjects tc , td

Subject	Stride length	Cadence	Leg length	Age
tc_1	0.59	134.0	0.66	8
tc_2	0.89	110.0	0.67	9
tc_3	1.04	119.0	0.71	11
td_1	0.20	49.5	0.45	3
td_2	0.51	74.0	0.47	4
td_3	0.76	131.0	0.52	5

category the patient is most likely belonging to, and finally assess the effectiveness of treatment during various period based on the change of AP .

The affiliated probability of subject over C_k is defined as

$$AP(ts_i, C_j) = \frac{1/AD(ts_i, C_j)}{\sum_{m=1}^K (1/AD(ts_i, C_m))} \tag{7}$$

where $AD(ts_i, C_j)$ is the affiliation distance between a testing subject and a given cluster.

4.3 Pattern Assessment

The affiliated probability distributions for patients tc and td are displayed in bar graph shown in Fig.6 (a) (b) respectively. Particularly, cluster 1 in these figures stands for the collecting of subject who are neurologically intact. For each pathological child, the three AP distributions corresponding to three various ages are illustrated in the form of bars, and the most “dominating” affiliated category could be explicitly identified by the presence of the highest bar. For example, for patient tc , the most “dominating” affiliated group shifts from group No.5 originally, then to group No.3, and

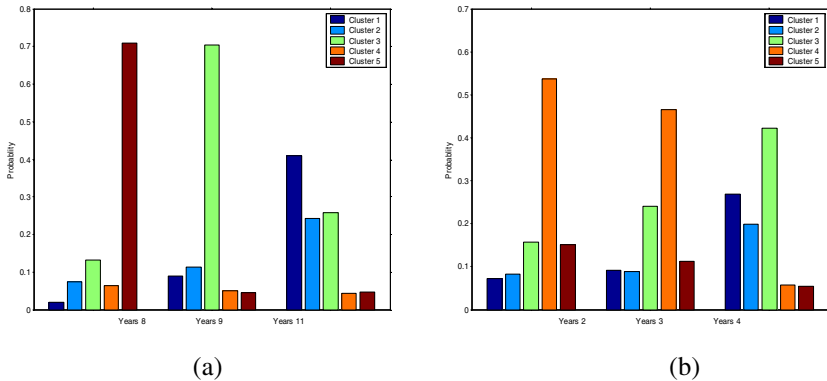


Fig. 6. Affiliated probability shift for subject tc (a) and td (b) at three age stages

finally to group No.1 (i.e. normal group) gradually. Similarly, that kind of shift for patient td is not as significant as tc , that group No.4 dominates in first two stages, then group No.3 overrun.

On the other hand, by comparison of the changes of AP distribution, it maybe suggested that the effectiveness of treatment and trend of recovery for patient tc is much better than that for patient td . In this case, the calculated AP could provide a suggestive measure and tool for doctor to assess the effectiveness and modify the strategy of treatment.

5 Conclusion and Future Work

Scientific gait (walking) analysis provides valuable information about an individual's locomotion function, in turn, assists to undertake appropriate measures for clinical diagnosis and prevention, such as assessing treatment for patients with impaired postural control and detecting risk of falls in elderly population.

In this paper, we addressed mining gait pattern for clinical locomotion diagnosis by using clustering techniques. Upon the constructed temporal-distance gait data model, clustering algorithm is employed to reveal the underlying correlation among subject observations and create pattern-based subject clusters. With the created clusters, gait pattern profiles, derived from the centroids of clusters, are then built up to reflect the overall gait characteristics. Moreover, analysis on clustering quality and comparison is conducted to visualize and evaluate the experimental results with k -means and hierarchical clustering algorithm respectively. In particular, the patient-specific gait data are tested to distinguish the likelihood of their belongings to various clusters and assess the changes of them before and after treatment. The results show the clustering-based technique is capable of efficiently identifying gait patterns and providing assistance for clinical applications.

In future work, we plan to investigate the selection of various feature subsets of gait parameters, discovery of possibly existing association rule among those features and collection of more gait dataset to testify the scalability of the proposed approach.

Acknowledgement

This research has been partly supported through Natural Science Foundation of Zhejiang Province, China (No Y105654).

References

1. Vaughan, C.L., B.L. Davis, and J.C. O'Connor, *Dynamics of Human Gait*, in *Human Kinetics*. 1992: Champaign, IL.
2. Judge, J.O., R.B. Davis, and S. O' unpuu, *Step length reductions in advanced age: The role of ankle and hip kinetics*. *J. Gerontol.: Med. Sci.*, 1996. **51**: p. 303-312.
3. Nigg, B.M., V. Fisher, and J.L. Ronsky, *Gait characteristics as a function of age and gender*. *Gait Posture*, 1994. **2**: p. 213-220.

4. Ostrosky, K.M., et al., *A comparison of gait characteristics in young and old subjects*. Phys. Ther., 1994. **74**: p. 637-646.
5. Tibarewala, D.N. and S. Ganguli, *Pattern recognition in tachographic gait records of normal and lower extremity handicapped human subjects*. J. Biomed. Eng., 1982. **4**: p. 233-240.
6. Damiano, D.L. and M.F. Abel, *Relationship of gait analysis to gross motor function in cerebral palsy*. Develop. Med. Child Neurol., 1996. **38**: p. 389-396.
7. Winters, T.F., J.G. Gage, and R. Hicks, *Gait patterns in spastic hemiplegia in children and young adults*. J. Joint Bone Surg, 1987. **69A**: p. 437-441.
8. Perry, J., et al., *Classification of walking handicap in the stroke population*. Stroke, 1995. **26**: p. 982-989.
9. O'Malley, M.J., et al., *Fuzzy Clustering of Children with Cerebral Palsy Based on Temporal-Distance Gait Parameters*. IEEE Tran. ON Rehab. Eng. , 1997. **5**(4).
10. Barton, J.G. and A. Lees, *An application of neural networks for distinguishing gait patterns on the basis of hip-knee joint angle diagrams*. Gait Posture, 1997. **5**: p. 28-33.
11. Holzreiter, S.H. and M.E. Kohle, *Assessment of gait pattern using neural networks*. J. Biomech., 1993. **26**: p. 645-651.
12. Begg, R.K., M. Palaniswami, and B. Owen, *Support Vector Machines for Automated Gait Classification*. IEEE Tran. on Biomed. Eng., 2005. **52**(5).
13. Lee, L. and W.E.L. Grimson. *Gait analysis for recognition and classification*. in *Proc. 5th Int. Conf. Automatic Face Gesture Recognition (FGR'02)*. 2002.
14. Chapelle, O., P. Haffner, and V.N. Vapnik, *Support vector machines for histogram-based classification*. IEEE Trans. Neural Netw., 1999. **10**(5): p. 1055-1064.
15. Chan, K., et al., *Comparison of machine learning and traditional classifiers in glaucoma diagnosis*. IEEE Trans. Biomed. Eng., 2002. **49**(9): p. 963-974.
16. Inman, V.T., H.J. Ralston, and F. Todd, *Human Walking*. 1981, Baltimore, MD: Williams and Wilkins.
17. Baeza-Yates, R. and B. Ribeiro-Neto, *Modern information retrieval*. 1999, Sydney: Addison Wesley.
18. Han, J. and M. Kambe, *Data Mining: Concepts and Techniques*. Data Management Systems. 2000: Morgan Kaufmann Publishers.
19. Hotho, A., A. Mädche, and S. Staab. *Ontology-based Text Clustering*. in *Workshop "Text Learning: Beyond Supervision"*, *IJCAI 2001*. 2001.
20. Vaughan, C.L., B. Berman, and W.J. Peacock, *Gait analysis and rhizotomy. A three year follow-up evaluation with gait analysis*. J. Neurosurg., 1991. **74**: p. 178-184.

Combining Multiple Clusterings Via k-Modes Algorithm

Huilan Luo^{1,2}, Fansheng Kong¹, and Yixiao Li¹

¹ Artificial Intelligence Institute, Zhejiang University, Hangzhou 310027, China
d051luohuilan@zju.edu.cn

² Institute of Information Engineering, Jiangxi University of Science and Technology,
Gangzhou 341000, China

Abstract. Clustering ensembles have emerged as a powerful method for improving both the robustness and the stability of unsupervised classification solutions. However, finding a consensus clustering from multiple partitions is a difficult problem that can be approached from graph-based, combinatorial or statistical perspectives. A consensus scheme via the k-modes algorithm is proposed in this paper. A combined partition is found as a solution to the corresponding categorical data clustering problem using the k-modes algorithm. This study compares the performance of the k-modes consensus algorithm with other fusion approaches for clustering ensembles. Experimental results demonstrate the effectiveness of the proposed method.

1 Introduction

Data clustering is a difficult inverse problem, and as such is ill-posed when prior information about the underlying data distributions is not well defined. Numerous clustering algorithms are capable of producing different partitions of the same data that capture various distinct aspects of the data. The exploratory nature of clustering tasks demands efficient methods that would benefit from combining the strengths of many individual clustering algorithms. This is the focus of research on clustering ensembles, seeking a combination of multiple partitions that provides improved overall clustering of the given data.

While the problem of clustering combination bears some traits of a classical clustering it also brings its own new challenges. One challenging issue of the problem of combining multiple clusterings is the choice of the generation method of the component partitions for the ensemble. Diversity of the individual clusterings of a given data set can be achieved by a number of approaches. Applying various clustering algorithms [4], [5], using one algorithm with different built-in initialization and parameters [6], [7], [11], projecting data onto different subspaces [4], [8], choosing different subsets of features [4], and selecting different subsets of data points [3], [9], [10], [14], [15] are instances of these generative mechanism.

The major difficulty of clustering combination is in finding a consensus partition from the ensemble of partitions. Fred [7] proposed to summarize various clustering results in a co-association matrix. Co-association values represent the strength of association between objects by analyzing how often each pair of objects appears in

the same cluster. Then the co-association matrix serves as a similarity matrix for the data items. The final clustering is formed from the co-association matrix by linking the objects whose co-association value exceeds a certain threshold. Further work by Fred and Jain [11] also used co-association values, but instead of a fixed threshold, they applied a hierarchical (single-link) clustering to the co-association matrix. One drawback of the co-association consensus function is its quadratic computational complexity in the number of objects $O(N^2)$. And experiments [2] show co-association methods are usually unreliable with number of clusterings $H < 50$.

Strehl and Ghosh [4] have considered three different consensus functions for ensemble clustering. The Cluster-based Similarity Partitioning Algorithm (CSPA) [4] induces a graph from a co-association matrix and clusters it using the METIS algorithm [12]. Hypergraph partitioning algorithm (HGPA) [4] represents each cluster by a hyperedge in a graph where the nodes correspond to a given set of objects. Good hypergraph partitions are found using minimal cut algorithms such as HMETIS [13] coupled with the proper objective functions, which also control partition size. Hyperedge collapsing operations are considered in another hypergraph-based Meta-Clustering algorithm (MCLA) in [4]. The meta-clustering algorithm uses these operations to determine soft cluster membership values for each object. Complexity of CSPA, HGPA and MCLA are $O(kN^2H)$, $O(kNH)$, and $O(k^2NH^2)$, respectively.

A different consensus function was developed in [6] based on information-theoretic principles, namely using generalized mutual information (MI). It was shown that the underlying objective function is equivalent to the total intra-cluster variance of the partition in the specially transformed space of labels. Therefore, the k-means algorithm in such a space can quickly find corresponding consensus solutions. Computational complexity of this algorithm is low, $O(kNH)$, but it may require a few restarts in order to avoid convergence to low quality local minima.

In [9], [14], [15], a combination of partitions by re-labeling and voting is implemented. Their works pursued direct re-labeling approaches to the correspondence problem. A re-labeling can be done optimally between two clusterings using the Hungarian algorithm [19]. After an overall consistent re-labeling, voting can be applied to determine cluster membership for each object. However, this voting method needs a very large number of clusterings to obtain a reliable result.

Alexander Topchy et al. [2] offer a probabilistic model of consensus using a finite mixture of multinomial distributions in the space of cluster labels. A combined partition is found as a solution to the corresponding maximum likelihood problem using the EM algorithm. EM consensus function needs to estimate at least kHM parameters. Therefore, accuracy degradation will inevitably occur with increasing number of partitions when sample size is fixed [2].

This work focuses on the primary problem of clustering ensembles, namely the consensus function, which creates the combined clustering. We propose a new fusion method for the clustering ensemble that is based on the k-modes algorithm. By transforming the problem of combining partitions to a categorical clustering problem, we use the k-modes [1] algorithm to solve it.

2 The k-Modes Algorithm

The k-modes algorithm [1] extends the k-means paradigm to categorical domains. It is built upon four basic operations: (1) selection of the initial k modes for k clusters, (2) calculation of the dissimilarity between an object and the mode of a cluster, (3) allocation of an object to the cluster whose mode is nearest to the object, (4) recalculation of the mode of a cluster from the objects allocated to it so that the intra cluster dissimilarity is minimized. Except for the first operation, the other three operations are repeatedly performed in the algorithm until the algorithm converges. Three major modifications to the k-means algorithm have been made in the k-modes algorithm, i.e., using different dissimilarity measures, replacing k means with k modes, and using a frequency based method to update modes. These modifications are discussed below.

Let x, y be two categorical objects described by m categorical attributes. The dissimilarity measure between x and y can be defined by the total mismatches of the corresponding attribute categories of the two objects. The smaller the number of mismatches is, the more similar the two objects. Formally,

$$d(x, y) = \sum_{i=1}^m (1 - \delta(x_i, y_i)) \quad (1)$$

Theorem 1. Let $Q = [q_1, \dots, q_i, \dots, q_m]$ be the mode of the cluster C such that

$$f_r(A_j = q_j | C) \geq f_r(A_j = v_{k,j} | C) \quad j=1, \dots, m.$$

where $f_r(A_j = v_{k,j} | C) = \frac{n_{v_{k,j}}}{|C|}$ and $n_{v_{k,j}}$ be the number of objects whose attribute

A_j have category $v_{k,j}$ in the cluster C .

To decrease the effect of the data input order, we modified the k-modes algorithm [1] a little and implemented according the following steps:

1. Select k initial modes, one for each cluster.
2. Allocate an object to the cluster whose mode is the nearest to it according to their distance defined as the equation (1).
3. Update the mode of the cluster according to the theorem 1 after all allocations.
4. Retest the dissimilarity of objects against the current modes. If an object is found such that its nearest mode belongs to another cluster rather than its current one, reallocate the object to that cluster.
5. Repeat 3-4 until no object has changed clusters.

3 Problem of Consensus Clustering

Let X be a set of N data points (objects) in d -dimensional space. Suppose we are given a set of m partitions for the same data set X , $\Pi = \{\pi_1, \pi_2, \dots, \pi_m\}$. Each component partition π_i in Π is a set of k_i disjoint, exhaustive and nonempty clusters. The problem of consensus clustering is to find a new partition $P = \{c_1, \dots, c_k\}$ of data set X given the partitions in Π , such that the objects in a cluster of P are more similar to each other than two objects in different clusters of P . This statement of the problem is virtually the same as for a conventional clustering except that it uses information contained in already existing partitions $\{\pi_1, \pi_2, \dots, \pi_m\}$. It is convenient to characterize consensus clustering as clustering in a space of new features induced by the set Π . Indeed, each component partition π_i represents a feature with categorical values. The values assumed by the i -th new feature are simply the cluster labels from partition π_i . Therefore, membership of each object in different partitions is treated as a new feature vector, a m -tuple, given m different partitions in Π : $x_i \rightarrow y_i = [\pi_1(x_i), \pi_2(x_i), \dots, \pi_m(x_i)]$. Here, $\pi_j(x_i)$ denotes a label assigned to x_i by the j -th partition. The consensus clustering is found as a partition P of a set of vectors $Y = \{y_i\}$ that can directly translate to the partition of the underlying data points $\{x_i\}$. The problem of clustering combination is to find a new partition of data X that summarizes the information from the gathered partitions Π . Our main goal is to construct a consensus partition without the assistance of the original patterns in X , but only from their labels delivered by the contributing clustering algorithms. Clustering ensemble problem becomes equivalent to the clustering of the new data set Y with m new categorical features if we ignore the original data attributes. Then the k-modes algorithm [1] is applied to clustering the categorical data set Y .

We choose the k-modes algorithm because it has low time complexity. The computational cost of this algorithm is $O((t+1)kN)$, where N is the number of objects, k the number of clusters and t is the number of iterations of the reallocation process. Usually, $k \ll N$ and t does not exceed 100 according to our experiments on a large real world data set. Therefore, this algorithm is efficient in clustering large data sets.

The proposed ensemble clustering based on the k-modes algorithm is summarized below:

1. for $i=1$ to m // m - number of clusterings
 - use k-means to cluster the dataset X to get a clustering π_i ; add the partition to the ensemble Π ;
 - end
2. Form a new set of vectors $Y = \{y_i\}$ according to the ensemble Π such that

$$y_i = [\pi_1(x_i), \pi_2(x_i), \dots, \pi_m(x_i)]$$
;
3. Apply the k-modes algorithm on the set Y and get the final partition P ;

4. Directly translate P to the partition of the underlying data points $\{x_i\}$. Assign the original point x_i to cluster C if and only if the corresponding y_i of the matrix Y was assigned to cluster C .

Note that any clustering algorithm can be used to generate ensemble instead of the k-means algorithm shown in the above pseudo code. We have chosen the k-means algorithm as the partition generation mechanism, mostly for its low computational complexity.

4 Experimental Results and Discussion

The experiments were conducted with artificial and real world datasets, where true natural clusters are known, to validate both accuracy and robustness of consensus via the k-modes algorithm. We evaluated the performance of clustering algorithms by matching the detected and the known partitions of the datasets.

4.1 Comparison with Five Different Consensus Functions

In this section we present the results of a comparison between our ensemble approach and five other consensus functions: CSPA, HGPA, MCLA, EM, MI. The Iris data from UCI benchmark repository is used in the comparison. The results for the Iris datasets are presented in Table 1. The table reports the mean error rate (%) of clustering combination from 20 runs. First observation is that none of the consensus functions is the absolute winner. Good performance was achieved by different combination algorithms across the values of parameters k and H . The MCLA algorithm slightly outperforms other algorithms for ensembles of smaller size, while CSPA becomes stronger when the number of clusterings $H > 10$. All co-association methods are usually unreliable with number of clusterings $H < 50$. Our ensemble clustering approach has the stable and desirable performance when $H > 15$. The combination approach via the k-modes algorithm also should benefit from the datasets of large size due to its low computational complexity. Another valuable property of the k-modes consensus algorithm is its fast convergence rate.

Table 1. Mean error rate (%) for the Iris dataset

H	k	k-modes	EM	MI	CSPA	HGPA	MCLA
5	3	16.333	11.0	14.7	11.2	41.4	10.9
10	3	15.2	10.8	10.8	11.3	38.2	10.9
15	3	15.667	10.9	11.9	9.8	42.8	11.1
20	3	10.667	10.9	14.5	9.8	39.1	10.9
30	3	10.667	10.9	12.8	7.9	43.4	11.3
40	3	10.667	11.0	12.4	7.7	41.9	11.1
50	3	10.667	10.9	13.8	7.9	42.7	11.2

4.2 Analysis of Diversity and Quality for Cluster Ensembles

For supervised ensemble approaches, diversity of the base-level classifiers has proven to be a key element in increasing classification performance [17]. In the relatively new area of unsupervised ensembles, the impact of diversity and quality of the individual clustering solutions on the final ensemble performance has not been fully understood. To gain further insight into these issues, we introduced an artificial data set [18] shown in the figure 1 and examined the impact of the ensemble quality and diversity on performance of combination clustering via the k-modes algorithm in this data set.

To analyze how the quality of the individual clustering solutions influence our ensemble clustering performance. A random noise with probability p is applied to the true class label $C(x_j)$ of each object $x_j, j = 1, 2, \dots, N$. The value $C(x_j)$ is replaced by a new random label L from $\{1, \dots, k\}$ with equal probability, for all values $L \neq C(x_j)$. Hence, we can obtain a clustering with the quality $q=1-p$. Using this method we can produce an ensemble of 20 individual partitions with quality q . Figure 2 shows the quality-performance diagram for the circle data set in figure 1. The ensemble performance is measured by the error rate, the lower the error rate the higher the performance.

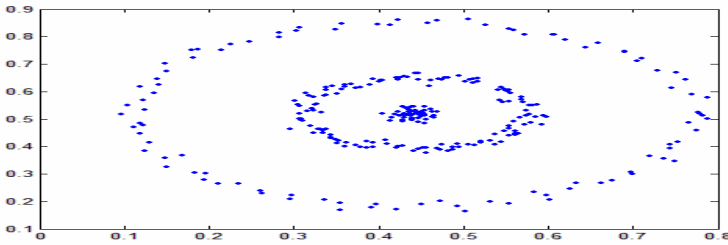


Fig. 1. The circle data set

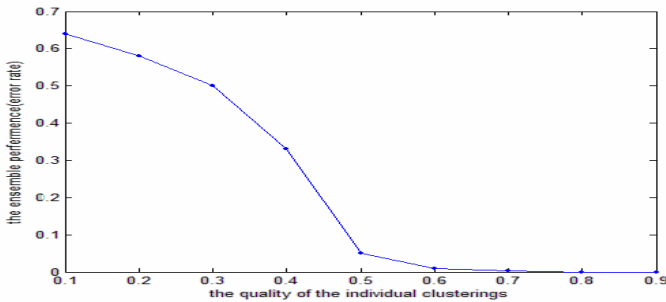


Fig. 2. The error rate of ensemble clustering via the k-modes algorithm versus the quality of the individual clusterings

From figure 2, we see evidence that the quality of individual clustering solutions limits the performance of a fixed size ensemble. When the quality of the component clusterings increase to 0.5, our clustering combination method via the k-modes algorithm can obtain a reliable result with a rather low error rate. As the quality of the individual clusterings increases to 0.6 or better, the ensemble performance increases to a perfect result with nearly zero error rate.

To analysis the diversity for cluster ensembles, we obtain 20 ensembles with different diversity but with equivalent average quality. Each ensemble is formed by 20 runs of k-means on the circle data set in figure 1. To measure diversity of an ensemble, we average the normalized mutual information (NMI) values between each pair of clustering solutions in the ensemble. Note that when the NMI values between two solutions is low the diversity is high. Figure 3 shows the diversity-performance diagram for the circle data set. In comparing the diversity to the performance of the entire ensemble we see evidence that for an ensemble of size 20, high diversity leads to greater improvements in the ensemble quality.

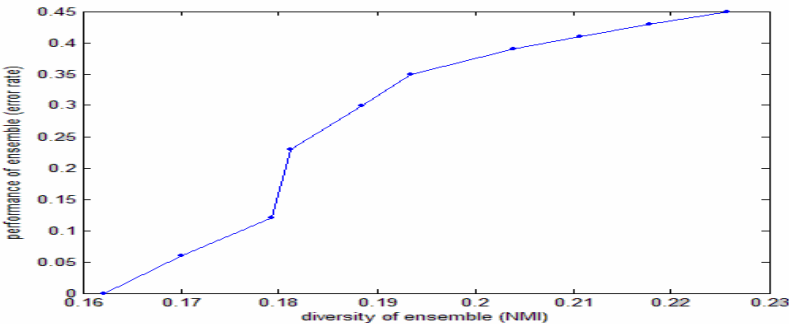


Fig. 3. The ensemble performance versus the diversity of the ensemble

These results suggest that our clustering combination method is strongly influenced by both the quality and the diversity of the individual clustering solutions. If the individual clustering solutions have little diversity, then not much leverage can be obtained by combining them. The quality of the individual solutions also limits the performance of a fixed-size ensemble and low quality solutions may cause the ensemble performance to oscillate as the ensemble size changes [8].

5 Conclusions

We have proposed a solution to the problem of clustering combination. A consensus clustering is derived from a solution of the categorical data clustering problem. The categorical data clustering problem is effectively solved using the k-modes algorithm. Experimental results indicate good performance of the approach for several datasets and favorable comparison with other consensus functions. Among the advantages of the approach is its low computational complexity. We also identified the importance of the quality and diversity of individual clustering solutions and illustrated their influence on the ensemble performance with empirical results.

References

1. Huang, Z.: A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (1997)
2. Topchy, A., Jain, A.K., Punch, W.: A Mixture Model for Clustering Ensembles. Proc. SIAM Conf. on Data Mining (2004) 379-390
3. Minaei-Bidgoli, B., Topchy, A.P., Punch, W.F.: A Comparison of Resampling Methods for Clustering Ensembles. IC-AI (2004) 939-945
4. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3 (2002) 583-617
5. Law, M., Topchy, A., Jain, A.K.: Multiobjective Data Clustering. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2 (2004) 424-430
6. Topchy, A., Jain, A., Punch, W.: Combining multiple weak clusterings. Proc. Third IEEE International Conference on Data Mining (ICDM'03) (2003)
7. Fred, A.L.N.: Finding Consistent Clusters in Data Partitions. Multiple Classifier Systems, Second International Workshop, MCS 2001 Cambridge, UK (2001) 309-318
8. Fern, X.Z., Brodley, C.E.: Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. Proc. of the 20th International Conference on Machine Learning (ICML 2003), Washington DC, USA (2003)
9. Fischer, B., Buhmann, J.M.: Path-Based Clustering for Grouping of Smooth Curves and Texture Segmentation. IEEE Trans. on PAMI 25 (2003) 513-518
10. Minaei-Bidgoli, B., Topchy, A.P., Punch, W.F.: Ensembles of Partitions via Data Resampling. International Conference on Information Technology: Coding and Computing (ITCC'04) (2004) 188-192
11. Fred, A.L.N., Jain, A.K.: Data Clustering using Evidence Accumulation. Proc. of the 16th Intl. Conference on Pattern Recognition ICPR 2002, Quebec City (2002) 276-280
12. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal of Scientific Computing 20 (1998) 359-392
13. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: Application in VLSI domain. Proc. 34th ACM/IEEE Design Automation Conference (1997) 526-529
14. Fischer, B., Buhmann, J.M.: Bagging for Path-Based Clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 25 (2003) 1411-1415
15. Dudoit, Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics 19 (2003) 1090-1099
16. Topchy, A., Jain, A.K., Punch, W.: Clustering Ensembles: Models of Consensus and Weak Partitions. IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1866-1881
17. Dietterich, T.G.: An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. Machine learning 2 (2000) 139-157
18. Zelnik-Manor, L., Perona, P.: Self-Tuning Spectral Clustering. Eighteenth Annual Conference on Neural Information Processing Systems, (NIPS) (2004)
19. Kuhn, H.W.: The hungarian method for the assignment problem. Naval Research Logistics Quarterly 2 (1955) 83-97

HOV³: An Approach to Visual Cluster Analysis

Ke-Bing Zhang¹, Mehmet A. Orgun¹, and Kang Zhang²

¹Department of Computing, Macquarie University, Sydney, NSW 2109, Australia
{kebing, mehmet}@ics.mq.edu.au

²Department of Computer Science, University of Texas at Dallas
Richardson, TX 75083-0688, USA
kzhang@utdallas.edu

Abstract. Clustering is a major technique in data mining. However the numerical feedback of clustering algorithms is difficult for user to have an intuitive overview of the dataset that they deal with. Visualization has been proven to be very helpful for high-dimensional data analysis. Therefore it is desirable to introduce visualization techniques with user's domain knowledge into clustering process. Whereas most existing visualization techniques used in clustering are exploration oriented. Inevitably, they are mainly stochastic and subjective in nature. In this paper, we introduce an approach called HOV³ (*Hypothesis Oriented Verification and Validation by Visualization*), which projects high-dimensional data on the 2D space and reflects data distribution based on user hypotheses. In addition, HOV³ enables user to adjust hypotheses iteratively in order to obtain an optimized view. As a result, HOV³ provides user an efficient and effective visualization method to explore cluster information.

1 Introduction

Clustering is an important technique that has been successfully used in data mining. The goal of clustering is to distinguish objects into groups (clusters) based on given criteria. In data mining, the datasets used in clustering are normally huge and in high dimensions. Nowadays, clustering process is mainly performed by computers with automated clustering algorithms. However, those algorithms favor clustering spherical or regular shaped datasets, but are not very effective to deal with arbitrarily shaped clusters. This is because they are based on the assumption that datasets have a regular cluster distribution.

Several efforts have been made to deal with datasets with arbitrarily shaped data distributions [2], [11], [9], [21], [23], [25]. However, those approaches still have some drawbacks in handling irregular shaped clusters. For example, CURE [11], FAÇADE [21] and BIRCH [25] perform well in low dimensional datasets, however as the number of dimension increases, they encounter high computational complexity. Other approaches such as density-based clustering techniques DBSCAN [9] and OPTICS [2], and wavelet based clustering WaveCluster [23] attempt to cope with this problem, but their non-linear complexity often makes them unsuitable in the analysis of very large datasets. In high dimensional spaces, traditional clustering algorithms tend to break down in terms of efficiency as well as accuracy because data do not cluster well

anymore. The recent clustering algorithms applied in data mining are surveyed by Jain *et al* [15] and Berkhin [4].

Visual Data Mining is mainly a combination of information visualization and data mining. In the data mining process, visualization can provide data miners with intuitive feedback on data analysis and support decision-making activities. In addition, visual presentations can be very powerful in revealing trends, highlighting outliers, showing clusters, and exposing gaps in data [24].

Many visualization techniques have been employed to study the structure of datasets in the applications of cluster analysis [18]. However, in practice, those visualization techniques take the problem of cluster visualization simply as a layout problem. Several visualization techniques have been developed for cluster discovery [2], [6], [16], but they are more exploration oriented, i.e., stochastic and subjective in the cluster discovery process.

In this paper, we propose a novel approach, named HOV³, *Hypothesis Oriented Verification and Validation by Visualization*, which projects the data distribution based on given hypotheses by visualization in 2D space. Our approach adopts the user hypotheses (quantitative domain knowledge) as measures in the cluster discovery process to reveal the gaps of data distribution to the measures. It is more object/goal oriented and measurable.

The rest of this paper is organized as follows. Section 2 briefly reviews related work on cluster analysis and visualization in data mining. Section 3 provides a more detailed account of our approach HOV³ and its mathematical description. Section 4 demonstrates the application of our approach on several well-known datasets in data mining area to show its effectiveness. Finally, section 5 evaluates our approach and provides a succinct summary.

2 Related Work

Cluster analysis is to find patterns (clusters) and relations among the patterns in large multi-dimensional datasets. In high-dimensional spaces, traditional clustering algorithms tend to break down in terms of efficiency as well as accuracy because data do not cluster well anymore. Thus, using visualization techniques to explore and understand high dimensional datasets is becoming an efficient way to combine human intelligence with the immense brute force computation power available nowadays [19].

Many studies have been performed on high dimensional data visualization [18]. While, most of those visualization approaches have difficulty in dealing with high dimensional and very large datasets, for example, icon-based methods [7], [17], [20] can display high dimensional properties of data. However, as the amount of data increases substantially, the user may find it hard to understand most properties of data intuitively, since the user cannot focus on the details of each icon. Plot-based data visualization approaches such as Scatterplot-Matrices [8] and similar techniques [1], [5] visualize data in rows and columns of cells containing simple graphical depictions. This kind of a technique gives bi-attributes visual information, but does not give the best overview of the whole dataset. As a result, they are not able to present clusters in the dataset very well.

Parallel Coordinates [14] utilizes equidistant parallel axes to visualize each attribute of a given dataset and projects multiple dimensions on a two-dimensional surface. Star Plots [10] arranges coordinate axes on a circle space with equal angles between neighbouring axes from the centre of a circle and links data points on each axis by lines to form a star. In principle, those techniques can provide visual presentations of any number of attributes. However, neither parallel coordinates nor star plots is adequate to give the user a clear overall insight of data distribution when the dataset is huge, primarily due to the unavoidably high overlapping. And another drawback of these two techniques is that while they can supply a more intuitive visual relationship between the neighbouring axes, for the non-neighbouring axes, the visual presentation may confuse the users' perception. HD-Eye [12] is an interactive visual clustering system based on density-plots of any two interesting dimensions. The 1D visualization based OPTICS [2] works well in finding the basic arbitrarily shaped clusters. But they lack the ability in helping the user understand inter-cluster relationships.

The approaches that are most relevant to our research are Star Coordinates [16] and its extensions, such as VISTA [6]. Star Coordinates arranges coordinate axes on a two-dimensional surface, where each axis shares the same origin point. This approach utilizes a point to represent a vector element. We give a more detailed discussion on Star Coordinates in contrast with our model in the next section. The recent surveys [3], [13] provide a comprehensive summary on high dimensional visualization approaches in data mining.

3 Our Approach

Data mining approaches are roughly categorized into *discovery* driven and *verification* driven [22]. Discovery driven approaches attempt to discover information by using appropriate tools or algorithms automatically, while verification driven approaches aim at validating a hypothesis derived from user domain knowledge. Discovery driven method can be regarded as discovering information by exploration, and the verification driven approach can be thought of as discovering information by verification.

Star Coordinates [16] is a good choice as an exploration discovery tool for cluster analysis in a high dimensional setting. Star Coordinates technique and its salient features are briefly presented below.

3.1 Star Coordinates

Star Coordinates arranges values of n -attributes of a database to n -dimensional coordinates on a 2D plane. The minimum data value on each dimension is mapped to the origin, and the maximum value, is mapped to the other end of the coordinate axis. Then unit vectors on each coordinate axis are calculated accordingly to allow scaling of data values to the length of the coordinate axes. Finally the values on n -dimensional coordinates are mapped to the orthogonal coordinates X and Y , which share the origin point with n -dimensional coordinates. Star Coordinates uses x - y values to represent a set of points on the two-dimensional surface, as shown in Fig.1.

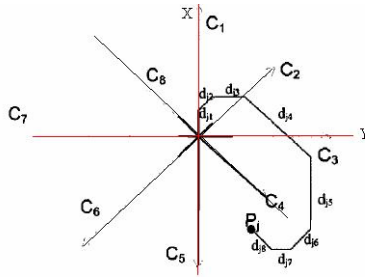


Fig. 1. Positioning a point by an 8-attribute vector in Star Coordinates [16]

Formula (1) states the mathematical description of Star Coordinates.

$$p_j(x, y) = \left(\sum_{i=1}^n \bar{u}_{xi} (d_{ji} - \min_i), \sum_{i=1}^n \bar{u}_{yi} (d_{ji} - \min_i) \right) \tag{1}$$

$p_j(x, y)$ is the normalized location of $D_j=(d_{j1}, d_{j2}, \dots, d_{jn})$, where d_{ji} is the coordinates point of j th record of a dataset on C_i , the i th coordinate in Star Coordinates space.

And $\bar{u}_{xi} \cdot (d_{ji} - \min_i)$ and $\bar{u}_{yi} \cdot (d_{ji} - \min_i)$ are unit vectors of d_{ji} mapping to X direction and Y direction respectively, where $\bar{u}_i = \bar{C}_i / (\max_i - \min_i)$, in which $\min_i = \min(d_{ji}, 0 \leq j < m)$, $\max_i = \max(d_{ji}, 0 \leq j < m)$, where m is the number of records in the dataset.

By mapping high-dimensional data into two-dimensional space, Star Coordinates inevitably produces data overlapping and ambiguities in visual form. For mitigating these drawbacks, Star Coordinates established visual adjustment mechanisms, such as scaling the weight of attributes of a particular axis; rotating angles between axes; marking data points in a certain area by coloring; selecting data value ranges on one or more axes and marking the corresponding data points in the visualization [16]. However, Star Coordinates is a typical method of exploration discovery.

Using numerically supported (quantitative) cluster analysis is time consuming and inefficient, while using visual (qualitative) clustering approaches, such as Star Coordinates is subjective, stochastic, and less of preciseness. To solve the problem of precision of visual cluster analysis, we introduce a new approach in the next section.

3.2 Our Approach HOV³

Having a precise overview of data distribution in the early stages of data mining is important, because having correct insights of data overview is helpful for data miners to make decisions on adopting appropriate algorithms for the forthcoming analysis stages.

3.2.1 Basic Idea

Exploration discovery (qualitative analysis) is regarded as a pre-processing of verification discovery (quantitative analysis), which is mainly used for building user hy-

potheses based on cluster detection, or other techniques. But it is not an aimless and/or arbitrary process.

Exploration discovery is an iterative process under the guidance of user domain knowledge. Each of iterations of exploration feeds back users new insight and enriches their domain knowledge on the dataset that they are dealing with. However, the way in which the qualitative analysis is done by visualization mostly depends on each individual user’s experience. Thus subjectivity, randomness and lack of preciseness may be introduced in exploration discovery. As a result, quantitative analysis based on the result of imprecise qualitative analysis may be inefficient and ineffective.

To fill the gap between the imprecise cluster detection by visualization and the un-intuitive result by clustering algorithms, we propose a new approach, called HOV³, which is a quantified knowledge based analysis and provides a bridging process between qualitative analysis and quantitative analysis. HOV³ synthesizes the feedbacks from exploration discovery and user domain knowledge to produce quantified measures, and then projects test dataset against the measures. Geometrically, HOV³ reveals data distribution against the measures in visual form. We give the mathematical description of HOV³ below.

3.2.2 Mathematic Model of HOV³

To project a high-dimensional space into a two-dimensional surface, we adopt the Polar Coordinates representation. Thus any vector can be easily transformed to the orthogonal coordinates *X* and *Y*.

In analytic geometry, the difference of two vectors *A* and *B* can be presented by their inner/dot product, *A.B*.

Let $A=(a_1, a_2, \dots, a_n)$ and $B=(b_1, b_2, \dots, b_n)$, then their inner product can be written as:

$$\langle A, B \rangle = a_1 \cdot b_1 + a_2 \cdot b_2 + \dots + a_n \cdot b_n = \sum_{k=1}^n a_k b_k \tag{2}$$

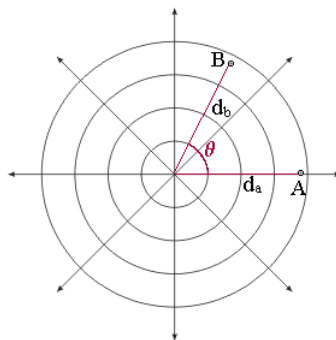


Fig. 2. Vector *B* projected against vector *A* in Polar Coordinates

Then we have the equation:

$$\cos(\theta) = \frac{\langle A, B \rangle}{|A| |B|}$$

where θ is the angle between A and B , and $|A|$ and $|B|$ are the lengths of A and B correspondingly, as shown below:

$$|A| = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2} \text{ and } |B| = \sqrt{b_1^2 + b_2^2 + \dots + b_n^2}$$

Let A be a unit vector; the geometry of $\langle A, B \rangle$ in Polar Coordinates presents the gap from point B (d_b, θ) to point A , as shown in Fig. 2, where A and B are in 8 dimensional space.

• **Mapping to Measures**

In the same way, a matrix D_j , a set of vectors (dataset) also can be mapped to a measure vector M . As a result, it projects the matrix D_j distribution based on the vector M .

Let $D_j = (d_{j1}, d_{j2}, \dots, d_{jn})$ and $M = (m_1, m_2, \dots, m_n)$, then the inner product of each vector d_{ji} , ($i = 1, \dots, n$) of D_j with M has the same equation as (2) and written as:

$$\langle d_{ji}, M \rangle = m_1 d_{j1} + m_2 d_{j2} + \dots + m_n d_{jn} = \sum_{k=1}^n m_k d_{jk} \tag{3}$$

So from an n -dimensional dataset to one measure (dimension) mapping $F: R^n \rightarrow R^2$ can be defined as:

$$F(D_j, M) = (\langle D_j, M \rangle) = \begin{pmatrix} d_{11}, d_{12}, \dots, d_{1n} \\ d_{21}, d_{22}, \dots, d_{2n} \\ \dots \dots \dots \dots \\ d_{m1}, d_{m2}, \dots, d_{mn} \end{pmatrix} \cdot (m_1, m_2, \dots, m_n) = \begin{pmatrix} m_1 d_{11}, m_2 d_{12}, \dots, m_n d_{1n} \\ m_1 d_{21}, m_2 d_{22}, \dots, m_n d_{2n} \\ \dots \dots \dots \dots \\ m_1 d_{m1}, m_2 d_{m2}, \dots, m_n d_{mn} \end{pmatrix} \tag{4}$$

Where D_j is a dataset with n attributes, and M is a quantified measure.

• **In Complex Number System**

Since our experiments are run by MATLAB (MATLAB®, the MathWorks, Inc), in order to better understand our approach, we use complex number system.

Let $z = x + i.y$, where i is the imaginary unit. According to the Euler formula:

$$e^{ix} = \cos x + i \sin x \quad \text{Let } z_0 = e^{2\pi i/n}; \text{ we see that } z_0^1, z_0^2, z_0^3, \dots, z_0^{n-1}, z_0^n$$

(with $z_0^n = 1$) divide the unit circle on the complex plane into $n-1$ equal sectors. Then mapping in Star Coordinates (1) can now be simply written as:

$$p_j(z_0) = \sum_{k=1}^n [(d_{jk} - \min_k d_{jk}) / (\max_k d_{kj} - \min_k d_{jk})] z_0^k \tag{5}$$

where, $\min_k d_{jk}$ and $\max_k d_{kj}$ represents minimal and maximal values of the k th attribute/coordinate respectively.

This is the case of equal-divided circle surface. Then the more general form can be defined as:

$$p_j(z_k) = \sum_{k=1}^n [(d_{jk} - \min_k d_{jk}) / (\max_k d_{kj} - \min_k d_{jk})] z_k \tag{6}$$

where $z_k = e^{i*\theta_k}$; θ is the angle of neighbouring axes; and $\sum_{k=1}^n \theta_k = 2\pi$.

While, the part of $(d_{jk} - \min_k d_{jk}) / (\max_k d_{kj} - \min_k d_{jk})$ in (5) (6) is normalized of original d_{jk} , we write it as dN_{jk} .

Thus formula (6) is written as:

$$p_j(z_k) = \sum_{k=1}^n dN_{jk} * z_k \tag{7}$$

In any case these can be viewed as mappings from R^n to C - the complex plane, i.e., $R^n \rightarrow C^2$.

Given a non-zero measure vector m in R^n , and a family of vectors P_j , then the projections of P_j against m according to formulas (4) and (7), we present our model HOV³ as the following equation (8):

$$p_j(z_k) = \sum_{k=1}^n dN_{jk} * m_k * z_k \tag{8}$$

where m_k is the k th attribute of measure m .

3.2.3 Discussion

In Star Coordinates, the purpose of scaling the weight of attributes of a particular axis (or α -mapping called in VISTA) is for adjusting the contribution of the attribute laid on a specific coordinate by the interactive actions, so that data miners might gain some interesting cluster information that automated clustering algorithms cannot easily provide [6], [16].

Thus, comparing the model of Star Coordinates, in equation (7), and our model HOV³ in equation (8), we may observe that our model covers the model of Star Coordinates, in that the condition of the angle of coordinates is the same in both models. This is because, any change of weights in Star Coordinates model can be viewed as changing one or more values of m_k ($k=1, \dots, n$) in measure vector m in equation (8) or (4). As a special case, when all values in m are set to 1, it is clear that HOV³ is transformed into Star Coordinates model (7), i.e., no measure case. In addition, either moving a coordinate axis to its opposite direction or scaling up the adjustment interval of axis, for example, from [0,1] to [-1,1] in VISTA, is also regarded as negating the original measure value.

Moreover, as a bridge between qualitative analysis and quantitative analysis, HOV³ not only supports quantified domain knowledge verification and validation, but also can directly utilize the rich statistical analysis tools as measures and guide data miners with additional cluster information. We demonstrate several examples running in MATLAB in comparison to the same dataset running in VISTA system [6] in the next section.

4 Examples and Explanation

In this section, we present several examples to demonstrate the advantages of using HOV³. We have implemented our approach in MATLAB running under Windows 2000 Professional. The results of our experiments with HOV³ are compared to those of VISTA, a Star Coordinates based system [6]. At this stage, we only employed several simple statistical methods on those datasets as measures. The datasets used in the examples are well known and can be obtained from the UCI machine learning website: <http://www.ics.uci.edu/~mlern/Machine-Learning.html>.

4.1 Iris

Iris dataset is perhaps the best-known in the pattern recognition literature. Iris has 3 classes, 4 numeric attributes and 150 instances.

The diagram presented in Fig. 3 (left) is the initial data distribution in Star Coordinates produced by the VISTA system. Fig 3 (right) shows the result of data distribution presented by HOV³ without any adopted measures. It can be observed that the shapes of data distribution are almost identical in the two figures. Only the directions for two shapes are little bit different, since VISTA shifted the appearance of data by 30 degrees in counter-clockwise direction.

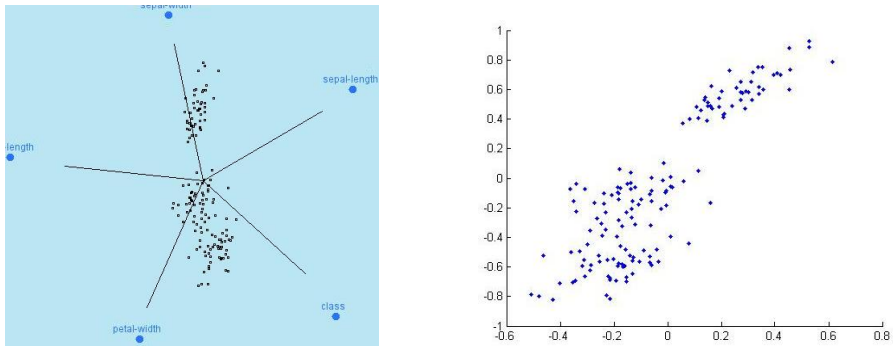


Fig. 3. The original data distribution in VISTA system (left) and its distribution by HOV³ in MATLAB (right)

Fig. 4 illustrates the results after several random weight adjustment steps. In Fig. 4, it can be observed very clearly that there are three data groups (clusters).

The initial data distribution cannot provide data miners a clear idea about the clusters, see Fig.3 (left). Thus, in VISTA the user may verify them by further interactive actions, such as weight scaling and/or changing angles of axes. However, though sometimes better results may appear, as shown in Fig.4, even users do not know where the results came from, because this adjustment process is pretty stochastic and not easily repeatable.

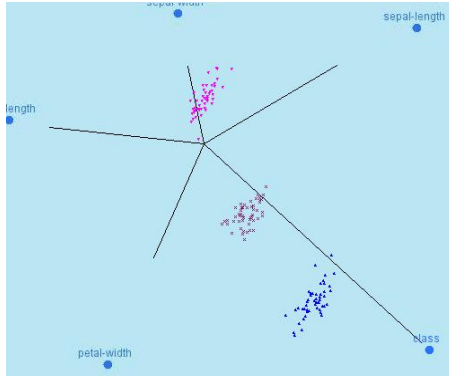


Fig. 4. The labeled clusters in VISTA after performing random adjustments by the system

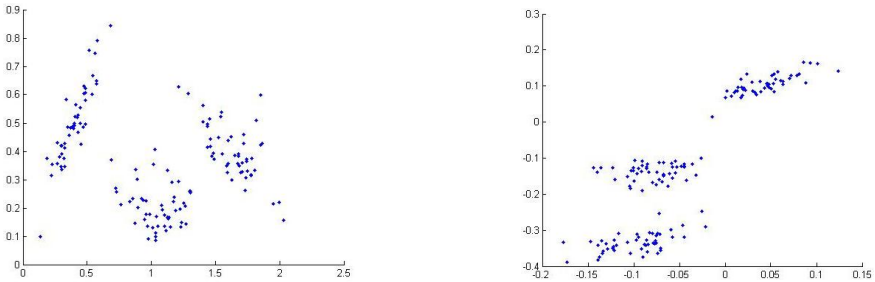


Fig. 5. Projecting iris data against to its mean (left), and Iris data projection against to its standard division (right)

We use simple statistical methods such as mean and standard division of Iris as measures to detect cluster information. Fig.5 gives the data projections based on these measures respectively. HOV^3 also provides three data groups, and in addition, several outliers. Moreover, the user can clearly understand how the results came about, and iteratively perform experiments with the same measures.

4.2 Shuttle

Shuttle dataset is much bigger both in size and in attributes than Iris. It has 10 attributes and 15,000 instances. Fig.6 illustrates the initial Shuttle data distribution, the same for both VISTA and HOV^3 .

The clustered data is illustrated in Fig.7 after performing manual weight scaling of axes in VISTA, where clusters are marked by different colours.

We used the median and the covariance matrix of Shuttle to detect the gaps of Shuttle dataset against its median and covariance matrix. The detected results are shown in Fig. 8. These distributions provide the user with different cluster information as in VISTA. On the other hand, HOV^3 can repeat the exact performance as VISTA did, if the user can record each weight scaling and quantified them, as mentioned in equation (8), HOV^3 model subsumes Star Coordinates based techniques.

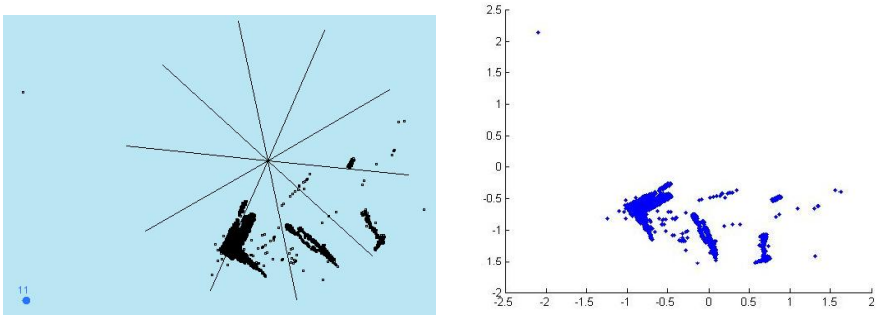


Fig. 6. Left: the initial shuttle data distribution in VISTA. Right: the initial shuttle data distribution in HOV³.

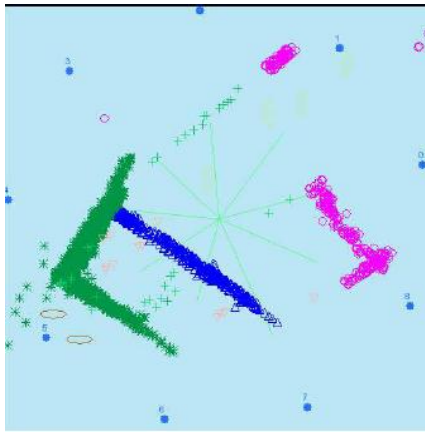


Fig. 7. Post adjustment of the Shuttle data with colored labels in VISTA

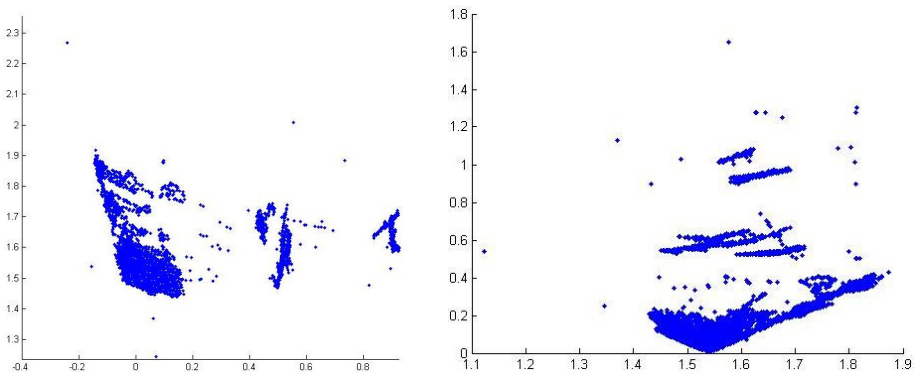


Fig. 8. Mapping shuttle dataset against to its median by HOV³ (left) and mapping shuttle dataset against to its covariance matrix (right)

The experiments we performed on the Shuttle dataset also show that HOV³ has the capability to provide users an efficient and effective method to verify their hypotheses by visualization. As a result, HOV³ can feed back more precise visual performance of data distribution to users.

5 Conclusions

In this paper we have proposed a novel approach called HOV³ to assist data miners in cluster analysis of high-dimensional datasets by visualization. The HOV³ visualization technique employs hypothesis oriented measures to project data and allows users to iteratively adjust the measures for optimizing the result of clusters.

Experiments show that HOV³ technique can improve the effectiveness of the cluster analysis by visualization and provide a better, intuitive understanding of the results. HOV³ can be seen as a bridging process between qualitative analysis and quantitative analysis. It not only supports quantified domain knowledge verification and validation, but also can directly utilize the rich statistical analysis tools as measures and give data miners an efficient and effective guidance to get more precise cluster information in data mining.

Iteration is a commonly used method in numerical analysis to find the optimized solution. HOV³ supports verification by quantified measures, thus provides us an opportunity to detect clusters in data mining by combining HOV³ and iteration method. This is the future goal of our work.

Acknowledgement

We would like to thank Kewei Zhang for his valuable support on mathematics of this work. We also would like to express our sincere appreciation to Keke Chen and Ling Liu for offering their VISTA system code, which greatly accelerated our work.

References

1. Alpern B. and Carter L.: Hyperbox. Proc. Visualization '91, San Diego, CA (1991) 133-139
2. Ankerst M., Breunig MM., Kriegel, Sander HP. J.: OPTICS: Ordering points to identify the clustering structure. Proc. of ACM SIGMOD Conference (1999) 49-60
3. Ankerst M., and Keim D.: Visual Data Mining and Exploration of Large Databases. 5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01), Freiburg, Germany, September (2001)
4. Berkhin P.: Survey of clustering data mining techniques. Technical report, Accrue Software (2002)
5. Cook D.R., Buja A., Cabrea J., and Hurley H.: Grand tour and projection pursuit. Journal of Computational and Graphical Statistics Volume: 23 (1995) 225-250
6. Chen K. and Liu L.: VISTA: Validating and Refining Clusters via Visualization. Journal of Information Visualization Volume: 13 (4) (2004) 257-270

7. Chernoff H.: The Use of Faces to Represent Points in k-Dimensional Space Graphically. *Journal Amer. Statistical Association*, Volume: 68 (1973) 361-368
8. Cleveland W.S.: *Visualizing Data*. AT&T Bell Laboratories, Murray Hill, NJ, Hobart Press, Summit NJ. (1993)
9. Ester M., Kriegel HP., Sander J., Xu X.: A density-based algorithm for discovering clusters in large spatial databases with noise. 2nd International Conference on Knowledge Discovery and Data Mining (1996)
10. Fienberg S. E.: Graphical methods in statistics. *American Statisticians* Volume: 33 (1979) 165-178
11. Guha S., Rastogi R., Shim K.: CURE: An efficient clustering algorithm for large databases. In *Proc. of ACM SIGMOD Int'l Conf. on Management of Data*, ACM Press (1998) 73--84
12. Hinneburg, A. Keim D. A., Wawryniuk M.: HD-Eye-Visual Clustering of High dimensional Data. *Proc. of the 19th International Conference on Data Engineering*, (2003) 753-755
13. Hoffman P. E. and Grinstein G.: A survey of visualizations for high-dimensional data mining. In Fayyad U., Grinstein G. G. and Wierse A. (eds.) *Information visualization in data mining and knowledge discovery*, Morgan Kaufmann Publishers Inc. (2002) 47-82
14. Inselberg A.: Multidimensional Detective. *Proc. of IEEE Information Visualization '97* (1997) 100-107
15. Jain A., Murty M. N., and Flynn P.J.: *Data Clustering: A Review*. *ACM Computing Surveys* Volume: 31(3) (1999) 264-323
16. Kandogan E.: Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. *Proc. of ACM SIGKDD Conference*, (2001) 107-116
17. Keim D.A. And Kriegel HP.: VisDB: Database Exploration using Multidimensional Visualization. *Computer Graphics & Applications* (1994) 40-49
18. Maria Cristina Ferreira de Oliveira, Haim Levkowitz: From Visual Data Exploration to Visual Data Mining: A Survey. *IEEE Transaction on Visualization and Computer Graphs* Volume: 9(3) (2003) 378-394
19. Pampalk E., Goebl W., and Widmer G.: Visualizing Changes in the Structure of Data for Exploratory Feature Selection. *SIGKDD '03*, Washington, DC, USA (2003)
20. Pickett R. M.: Visual Analyses of Texture in the Detection and Recognition of Objects. *Picture Processing and Psycho-Pictorics*, Lipkin B. S., Rosenfeld A. (eds.) Academic Press, New York, (1970) 289-308
21. Qian Y., Zhang G., and Zhang K.: FAÇADE: A Fast and Effective Approach to the Discovery of Dense Clusters in Noisy Spatial Data. In *Proc. ACM SIGMOD 2004 Conference*, ACM Press (2004) 921-922
22. Ribarsky W., Katz J., Jiang F. and Holland A.: Discovery visualization using fast clustering. *Computer Graphics and Applications*, IEEE, Volume: 19 (1999) 32-39
23. Sheikholeslami G., Chatterjee S., Zhang A.: WaveCluster: A multi-resolution clustering approach for very large spatial databases. *Proc. of 24th Intl. Conf. On Very Large Data Bases* (1998) 428-439.
24. Shneiderman B.: Inventing Discovery Tools: Combining Information Visualization with Data Mining. *Discovery Science 2001*, Proceedings. *Lecture Notes in Computer Science* Volume: 2226 (2001) 17-28
25. Zhang T., Ramakrishnan R. and Livny M.: BIRCH: An efficient data clustering method for very large databases. In *Proc. of SIGMOD96*, Montreal, Canada (1996) 103-114

A New Fuzzy Co-clustering Algorithm for Categorization of Datasets with Overlapping Clusters

William-Chandra Tjhi and Lihui Chen

Nanyang Technological University,
Republic of Singapore
william_chandra@pmail.ntu.edu.sg, elhchen@ntu.edu.sg

Abstract. Fuzzy co-clustering is a method that performs simultaneous fuzzy clustering of objects and features. In this paper, we introduce a new fuzzy co-clustering algorithm for high-dimensional datasets called Cosine-Distance-based & Dual-partitioning Fuzzy Co-clustering (CODIALING FCC). Unlike many existing fuzzy co-clustering algorithms, CODIALING FCC is a dual-partitioning algorithm. It clusters the features in the same manner as it clusters the objects, that is, by *partitioning* them according to their natural groupings. It is also a cosine-distance-based algorithm because it utilizes the cosine distance to capture the belongingness of objects and features in the co-clusters. Our main purpose of introducing this new algorithm is to improve the performance of some prominent existing fuzzy co-clustering algorithms in dealing with datasets with high overlaps. In our opinion, this is very crucial since most real-world datasets involve significant amount of overlaps in their inherent clustering structures. We discuss how this improvement can be made through the dual-partitioning formulation adopted. Experimental results on a toy problem and five large benchmark document datasets demonstrate the effectiveness of CODIALING FCC in handling overlaps better.

1 Introduction

Rapid advancement in various information technology related disciplines has made it possible for people to store huge amount of data in increasingly more complex data formats. One essential task in data mining is categorization of data, which enables us to reveal the potentially important but usually hidden grouping patterns of data. Of the two main methods to perform categorization, i.e. classification and clustering, the latter is often seen as the more practical one due to its unsupervised nature, where no or minimum prior knowledge is required to achieve the outcomes. By definition, clustering is an unsupervised process of categorizing data into several groups such that data belonging to one group are highly similar to one another, while data of different groups are highly dissimilar [1][2]. Due to its importance, various sophisticated clustering algorithms have been developed [2] and implemented for numerous practical applications such as: search results categorization in the Web [3], analysis of gene expression data [4], and intrusion detection [5].

One class of clustering algorithms has attracted a lot of attentions lately is co-clustering (or biclustering) [4][6][7]. If conventional clustering algorithms usually

only categorize one aspect of data, i.e. the objects, co-clustering algorithms simultaneously categorize both objects and their features (the objects' attributes). A co-cluster can then be defined as an associated object-cluster and feature cluster pair generated by a co-clustering process. In addition to being more informative, co-clustering is also argued to be able to capture the inherent groupings of objects more accurately than its conventional clustering counterpart [8]. This is because in most cases, not all features are relevant to the formation of object clusters. Unlike in the standard clustering approach, where all the features are considered equally important, co-clustering, through its feature clustering procedure, provides a possible mean to perform object clustering based only on a set of relevant features [4]. Furthermore, by clustering the features, co-clustering implicitly achieves dimensionality reduction, which in turn allows it to avoid the curse of high dimensionality [6]. Active researches, particularly in bioinformatics and information retrieval areas, are on the way in the pursuit of more effective and practical co-clustering algorithms. More detailed discussions on the concept of co-clustering can be found in [4].

The focus of this paper is on fuzzy co-clustering, which is a variant of co-clustering that incorporates concepts from the Fuzzy Set Theory [10] to generate fuzzy co-clusters. They are called fuzzy co-clusters because each one of them may have some degrees of overlap with other co-clusters. This is so because the assignment of an object (or a feature) to a co-cluster is represented by a membership function with value ranges from 0 to 1. This is as opposed to crisp co-clustering, which has membership value either 0 or 1 only. Compared to the crisp counterpart, fuzzy co-clustering offers a more realistic approach to co-cluster real world data, which are known to generally have significant amount of overlaps. Two well-known fuzzy co-clustering algorithms in the literatures are Fuzzy Simultaneous Keyword Identification and Clustering of Text Documents (FSKWIC) [11] and Fuzzy Co-clustering of Documents and Keywords (Fuzzy CoDoK) [12].

The objective of this paper is to introduce a new fuzzy co-clustering algorithm, Cosine-Distance-based & Dual-partitioning Fuzzy Co-clustering (CODIALING FCC), to improve the performance FSKWIC and Fuzzy CoDoK when dealing with datasets with high overlaps. Like its predecessors, the new algorithm essentially performs an iterative update of object and feature memberships to generate optimum fuzzy co-clusters. And similar to FSKWIC, it makes use of the cosine distance to decide the membership value of an object (or a feature) in a co-cluster. Different from the three existing algorithms however, CODIALING FCC imposes the same standard partitioning constraint (also known as the Ruspini's condition) [13] on both object and feature memberships. This results in dual partitioning of objects and features. This is in contrast with the approach adopted by FSKWIC and Fuzzy CoDoK, where only objects are partitioned, while features are ranked (or weighted) [11]. As we will see later, the fact that we use feature partitioning instead of feature ranking can result in CODIALING FCC being more effective than its predecessors when dealing with overlaps. The design on CODIALING FCC involves two steps. Firstly, the update membership equations are derived based on the minimization of a newly proposed objective function. Secondly, we perform a heuristic modification to the resulting update equations in order to achieve better performance.

The remaining of this paper is organized as follows. Section 2 gives some backgrounds by briefly reviewing related works on fuzzy co-clustering, which is then

followed by analyses on why overlaps can cause a problem in the existing formulations and on how to improve co-clustering when dealing with them. Section 3 discusses the design of the proposed algorithm. Section 4 presents some experimental results. Section 5 concludes the paper.

2 Background

We divide this section into three subsections. In the first subsection, we briefly discuss the key ideas behind some related works on fuzzy co-clustering. The second subsection details our analysis on why the existing algorithms may not be able to handle overlaps well. In the last subsection, a possible improvement is discussed.

2.1 Related Works

As indicated earlier, FSKWIC [11] and Fuzzy CoDoK [12] share similar frameworks where fuzzy co-clustering is seen as a process of optimizing an objective function subjects to certain constraints. Each algorithm then proceeds by iteratively updating the object and feature membership values in a steepest ascent (or descent) fashion until convergence is achieved. The resulting object and feature memberships reflect the generated fuzzy object and feature clusters (called fuzzy co-clusters).

In the case of FSKWIC, the optimization tries to minimize the aggregate of the “weighted” cosine distances between object and object clusters' centers (or prototypes). The cosine distance is said to be weighted due to the fact that the feature membership values are taken into account in the distance computation. In the case of Fuzzy CoDoK, the algorithm tries to maximize an entity called the degree of aggregation, which captures how objects and features are associated in a given co-cluster. Maximizing the degree of aggregation causes highly associated objects and features to be assigned to the same co-cluster.

Even though there are some significant differences between FSKWIC and Fuzzy CoDoK, both algorithms share common constraints. There are two constraints actually, one for object memberships and the other for feature memberships. Eqs. (1) and (2) show these two constraints respectively. Table 1 provides the list of all mathematical notations used in this paper.

$$\sum_{c=1}^c u_{ci} = 1 \text{ for } i = 1, 2, 3, \dots, N \quad (1)$$

$$\sum_{j=1}^k v_{cj} = 1 \text{ for } c = 1, 2, 3, \dots, C \quad (2)$$

Eq. (1) essentially states that the summation of all the memberships of each object in all the co-clusters must be equal to one. This constraint is the standard partitioning constraint [13] used in many popular fuzzy clustering algorithms such as Fuzzy C-means [14]. Eq. (2) on the other hand, states that the summation of all feature memberships in each co-cluster must be equal to one. Notice the orientation difference between (1) and (2). In (1), an object will have a high membership to a co-cluster if it is more relevant to this particular co-cluster than to the other co-clusters. Thus, the constraint (1) reflects object-partitioning process. In (2), a feature will have

high membership to a co-cluster if, of all the features, it is the one most relevant to the co-cluster. Therefore, the constraint (2) reflects feature-ranking process. The following section discusses how feature ranking may adversely affect the algorithms' performances when dealing with datasets with high overlaps.

Table 1. List of Mathematical Notations

Notation	Meaning
C, N, K	Number of co-clusters, objects, and features respectively
D_{cij}^o	Component-wise cosine distance between object and object center
D_{cij}^f	Component-wise cosine distance between feature and feature center
p_{cj}^o, p_{ci}^f	Object center and feature center respectively
T_u, T_v	Object's and feature's degrees of fuzziness respectively
u_{ci}, v_{ej}	Object and feature memberships respectively
a_{ij}	Object-feature association
a_{ij}^o	Object-feature association, normalized in object orientation
a_{ij}^f	Object-feature association, normalized in feature orientation
τ	Number of iterations
ϵ	Convergence indicator

2.2 Feature Ranking and the Problem in Dealing with Overlaps

Consider the problem of co-clustering the following object-feature association matrix A_1 into two diagonal co-clusters, assuming that objects and features are represented by rows and columns respectively:

$$A_1 = \begin{bmatrix} 1 & 0.8 & 0.6 & 0 & 0 \\ 1 & 0.8 & 0.6 & 0 & 0 \\ 0 & 0 & 0.6 & 0.2 & 1.265 \\ 0 & 0 & 0.6 & 0.3 & 1.245 \end{bmatrix}$$

The two rectangles denoted by $C1$ and $C2$ depict the ideal two co-clusters respectively. Notice how the two co-clusters overlap at feature (column) 3. Therefore, it should be natural if feature 3 has equal membership value in both $C1$ and $C2$. Applying FSKWIC and Fuzzy CoDoK to solve this problem however, resulted in the following object and feature membership matrices (U and V respectively), with rows representing co-clusters and columns representing objects and features respectively.

$$U_{FSKWIC} = U_{FCODOK} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$V_{FSKWIC} = \begin{bmatrix} 0.5 & 0.32 & 0.18 & 0 & 0 \\ 0 & 0 & 0.186 & 0.01 & 0.813 \end{bmatrix}$$

$$V_{FCODOK} = \begin{bmatrix} 0.417 & 0.333 & 0.25 & 0 & 0 \\ 0 & 0 & 0.285 & 0.119 & 0.596 \end{bmatrix}$$

Based on U_{FSKWIC} and U_{FCODOK} , the two algorithms assign objects (rows) 1 and 2 to $C1$ and 3 and 4 to $C2$, which is as intended. However, it can also be seen that in both V_{FSKWIC} and V_{FCODOK} , feature 3 has higher membership to $C2$ than to $C1$. This contradicts our earlier presumption that feature 3 would have equal memberships to both co-clusters. This example illustrates the existing problem in fuzzy co-clustering when dealing with dataset with overlaps such as on feature 3. This problem actually has roots in the feature-ranking processes in FSKWIC and Fuzzy CoDoK. It should be obvious that in the A_1 matrix above, feature 3 ranks the 3rd in $C1$ and the 2nd in $C2$. For this reason, it has higher membership to $C2$. When the features are well separated, such as in the case of features 1, 2, 4, and 5 in A_1 , feature rankings may be able to correctly capture the clustering structure of the features. This is because such features always have lower rankings in the non-relevant co-clusters than in the relevant counterparts. This is not the case however, when the features overlap among different co-clusters. In our simple illustration, there is only a small deviation in the membership values of feature 3 (i.e. 0.18 vs. 0.186 and 0.25 vs. 0.285). In the case of a dataset that has a large number of features however, an overlapping feature may have very different rankings in different co-clusters. This would increase the deviations in memberships and in turn, may affect the overall performance of the existing algorithms.

2.3 The Dual-Partitioning Approach: A Possible Improvement

One possible way to address the problem in dealing with overlapping datasets is by incorporating feature partitioning, instead of feature ranking, in the formulation of objective function. Thus, we have dual partitioning of both objects and features. To achieve this, the constraint in (2) should be changed to the following:

$$\sum_{c=1}^C v_{cj} = 1 \text{ for } j=1,2,3,\dots,K \tag{3}$$

This constraint, which is the standard partitioning constraint for feature memberships, would force the features to be partitioned among existing co-clusters. Similar to the case of object memberships, because of eq. (3), a feature will have high membership to a co-cluster if it is more relevant to this particular co-cluster than to the other co-clusters. Consequently, if a feature is equally relevant to more than one co-clusters, it will be assigned equal memberships to all these co-clusters. For this reason, if we incorporate this dual-partitioning approach, the deviation such as the one happens to the feature 3 memberships in the section 2.2 illustration will not take place. Thus, the problem in dealing with overlaps may be avoided. To show this, let us transpose the A_1 matrix so that the overlap occurs in the objects (rows) instead of features (columns).

$$A_1^T = \begin{array}{c|cc|cc} \hline & 1 & 1 & 0 & 0 \\ \hline & 0.8 & 0.8 & 0 & 0 \\ \hline & 0.6 & 0.6 & 0.6 & 0.6 \\ \hline & 0 & 0 & 0.2 & 0.3 \\ \hline & 0 & 0 & 1.265 & 1.245 \\ \hline \end{array}$$

Since the object memberships are constrained by (1), which conforms to the standard partitioning constraint, FSKWIC and Fuzzy CoDoK can capture the correct clustering structure of objects despite the overlap that occurs at object 3 (or equivalently feature 3 in A_1). The resulting object membership matrix below shows that object 3 has equal memberships to both co-clusters, which is what we desire.

$$U_{FSKWIC} = U_{FCODOK} = \begin{bmatrix} 1 & 1 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 1 & 1 \end{bmatrix}$$

3 CODIALING FCC Algorithm

The implementation of the dual-partitioning approach discussed in section 2.3 on CODIALING FCC is discussed here. The following shows the algorithm’s objective function. Again, all mathematical notations are listed in Table 1.

$$J = \sum_{c=1}^c \sum_{i=1}^N \sum_{j=1}^K u_{ci} v_{cj} (D_{cij}^o + D_{cij}^f) + \sum_{c=1}^c \sum_{i=1}^N u_{ci} \ln u_{ci} + \sum_{c=1}^c \sum_{j=1}^K v_{cj} \ln v_{cj} + \sum_{i=1}^N \lambda_i \left(\sum_{c=1}^c u_{ci} - 1 \right) + \sum_{j=1}^K \gamma_j \left(\sum_{c=1}^c v_{cj} - 1 \right) \tag{4}$$

where λ_i and γ_j are the Lagrange multipliers. Notice that the last two terms are due to the object and feature membership constraints in (1) and (3) respectively. Since both constraints conform to the standard partitioning constraint, the algorithm performs dual partitioning of object and features. In (4), D_{cij}^o denotes the component-wise cosine distance from object i to the object center in co-cluster c , while D_{cij}^f denotes the component-wise cosine distance from feature j to the feature center in co-cluster c . Based on the idea adopted in FSKWIC [11], they are defined as follows:

$$D_{cij}^o = 1/K - (a_{ij}^o p_{ci}^o) \tag{5}$$

$$D_{cij}^f = 1/N - (a_{ij}^f p_{ci}^f) \tag{6}$$

where p_{ci}^o denotes the object center in co-cluster c , p_{ci}^f denotes the feature center in co-cluster c , a_{ij}^o denotes the normalized a_{ij} in object orientation, and a_{ij}^f denotes the normalized a_{ij} in feature orientation. Note that there is no D_{cij}^f in FSKWIC because the presence of feature centers (or prototypes) does not really make sense in the feature-ranking context. Since we use the cosine distance, it is important that a_{ij}^o and a_{ij}^f are normalized at every iteration. The second and the third terms of (4) are the fuzzy entropy fuzzifiers [15]. They are needed to “fuzzify” the object and feature memberships in a similar manner as the parameter m “fuzzifies” the membership in the Fuzzy C-means algorithm [14]. Parameters T_u and T_v are used to adjust the degree of fuzziness of the resulting co-clusters. The higher they are, the fuzzier the results will be. Minimizing (4) using the Lagrange Multiplier method gives us the following update membership equations:

$$u_{ci} = \exp\left\{-\frac{1}{T_u} \sum_{j=1}^K v_{cj} (D_{cij}^o + D_{cij}^f)\right\} \left[\sum_{b=1}^C \exp\left\{-\frac{1}{T_u} \sum_{j=1}^K v_{bj} (D_{bij}^o + D_{bij}^f)\right\} \right]^{-1} \tag{7}$$

$$v_{cj} = \exp\left\{-\frac{1}{T_v} \sum_{i=1}^N u_{ci} (D_{cij}^o + D_{cij}^f)\right\} \left[\sum_{b=1}^C \exp\left\{-\frac{1}{T_v} \sum_{i=1}^N u_{bi} (D_{bij}^o + D_{bij}^f)\right\} \right]^{-1} \tag{8}$$

The term $\left\{-\sum_{j=1}^K v_{cj} (D_{cij}^o + D_{cij}^f)\right\}$ in the numerator (the first term) of (7) can be decomposed into two parts: $\left\{-\sum_{j=1}^K v_{cj} D_{cij}^o\right\}$ and $\left\{-\sum_{j=1}^K v_{cj} D_{cij}^f\right\}$. The former captures the cosine distance between object i and the object center in co-cluster c , with each distance dimension weighted by the membership of the feature in c . The latter captures the aggregate cosine distances, each between a feature and the feature center in co-cluster c , taking into account the dimension i (object i) only. Small values of these two measures indicate the *relevance* of object i to all the other objects and all the features in c . The denominator (the second term) of (7) serves as a mean to compare this relevance across all the existing co-clusters. The membership (u_{ci}) to the co-cluster that object i is most relevant to is assigned the highest value of all the existing co-clusters. The v_{cj} equation in (8) has the same principle.

One important thing that can be observed from (7) is that the membership assignment depends on the number of features in the co-cluster. In other words, the value of u_{ci} is influenced by the value of $\sum_{j=1}^K v_{cj}$. For illustration, consider the case where, given c and i , we have $\bigvee_{j=1}^K (D_{cij}^o + D_{cij}^f) \geq 0$. Then, as implied by the numerator of (7), there is a tendency that the value of u_{ci} would increase as the value of $\sum_{j=1}^K v_{cj}$ decreases (i.e. there are fewer features in the co-cluster). Since the value of $\sum_{j=1}^K v_{cj}$ does not reflect any relevance of object i to co-cluster c , the dependency on this entity induces a biased favoritism in the object partitioning process. In a similar fashion, the membership assignment using Eq. (8) depends on the number of objects in the co-cluster (i.e. the value of $\sum_{i=1}^N u_{ci}$), causing a bias in the feature partitioning process. Therefore, to ensure unbiased partitioning, we propose a heuristic that transforms the two update equations into the following:

$$u_{ci} = \exp\left\{\frac{\sum_{j=1}^K v_{cj} (D_{cij}^o + D_{cij}^f)}{T_u \sum_{j=1}^K v_{cj}}\right\} \left[\sum_{b=1}^C \exp\left\{\frac{\sum_{j=1}^K v_{bj} (D_{bij}^o + D_{bij}^f)}{T_u \sum_{j=1}^K v_{bj}}\right\} \right]^{-1} \tag{9}$$

$$v_{cj} = \exp \left\{ - \frac{\sum_{i=1}^N u_{ci} (D_{cij}^o + D_{cij}^f)}{T_v \sum_{i=1}^N u_{ci}} \right\} \left[\sum_{b=1}^c \exp \left\{ - \frac{\sum_{i=1}^N u_{bi} (D_{bij}^o + D_{bij}^f)}{T_v \sum_{i=1}^N u_{bi}} \right\} \right]^{-1} \tag{10}$$

By normalizing each argument in (7) and (8), we remove the dependencies to the numbers of objects & features in the co-clusters during the u_{ci} and v_{cj} computations respectively. The formulas for object and feature centers p_{cj}^o and p_{ci}^f are defined following the technique used in FSKWIC as follow.

$$p_{cj}^o = \begin{cases} 0, & \text{if } v_{cj} = 0 \\ \frac{\sum_{i=1}^N u_{ci} a_{ij}^o}{\sum_{i=1}^N u_{ci}}, & \text{otherwise} \end{cases}, \tag{11}$$

$$p_{ci}^f = \begin{cases} 0, & \text{if } u_{ci} = 0 \\ \frac{\sum_{j=1}^K v_{cj} a_{ij}^f}{\sum_{j=1}^K v_{cj}}, & \text{otherwise} \end{cases} \tag{12}$$

Table 2 shows the pseudo-code of CODIALING FCC. The new algorithm has a time complexity of $O(CNK\tau)$, where τ denotes the number of iterations. This is the same as the time complexities of some popular partitioning-based clustering algorithms such as K-means and Fuzzy C-means.

Table 2. CODIALING FCC Pseudo-code

CODIALING FCC Algorithm	
1.	Set parameters $C, T_u, T_v,$ and ϵ ;
2.	Normalize a_{ij} into a_{ij}^o and a_{ij}^f ;
3.	Randomly initialize $u_{ci}, p_{cj}^o, p_{ci}^f$;
4.	Normalize p_{cj}^o and p_{ci}^f ;
5.	REPEAT
6.	Update D_{cij}^o using (5) and D_{cij}^f using (6);
7.	Update v_{cj} using (10) and u_{ci} using (19);
8.	Update p_{cj}^o using (11) and p_{ci}^f using (12);
9.	Normalize p_{cj}^o and p_{ci}^f ;
10.	UNTIL $\max u_{ci}(\tau) - u_{ci}(\tau - 1) \leq \epsilon$

4 Experimental Results

We first show the membership matrices generated by CODIALING FCC when applied on the toy problem A_1 used in section 2.2. The membership matrices generated by FSKWIC and Fuzzy CoDoK are also re-captured to allow easier comparison.

$$A_1 = \begin{bmatrix} 1 & 0.8 & 0.6 & 0 & 0 \\ 1 & 0.8 & 0.6 & 0 & 0 \\ 0 & 0 & 0.6 & 0.2 & 1.265 \\ 0 & 0 & 0.6 & 0.3 & 1.245 \end{bmatrix}$$

$$U_{CODIALING} = U_{FSKWIC} = U_{FCODOK} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$V_{CODIALING} = \begin{bmatrix} 0.88 & 0.83 & 0.5 & 0.27 & 0.08 \\ 0.12 & 0.17 & 0.5 & 0.73 & 0.92 \end{bmatrix}$$

$$V_{FSKWIC} = \begin{bmatrix} 0.5 & 0.32 & 0.18 & 0 & 0 \\ 0 & 0 & 0.186 & 0.01 & 0.813 \end{bmatrix}$$

$$V_{FCODOK} = \begin{bmatrix} 0.417 & 0.333 & 0.25 & 0 & 0 \\ 0 & 0 & 0.285 & 0.119 & 0.596 \end{bmatrix}$$

By contrasting the matrices generated by CODIALING FCC and the other two algorithms, we can sense the dual-partitioning nature of the proposed algorithm. The results also indicate that even though all three algorithms are able to generate the ideal memberships for objects, only CODIALING FCC can closely capture the clustering

Table 3. Benchmark Datasets Summary

Dataset	No. Docs	No. Words	Clusters (No. Docs per Cluster)	Explanation
<i>Classic3</i>	3891	2176	Medical (1033), Aerospace (1398), Inform. Retrieval (1460)	Balanced and well-separated
<i>Binary</i>	500	3377	Politics (250), Middle-east (250)	Balanced and highly overlapping
<i>SM</i>	2000	5450	Soccer (1000), Motorsport (1000)	Balanced and highly overlapping clusters
<i>SS</i>	2000	6337	Soccer (1000), Sport (1000)	Balanced and highly overlapping. Sport is more general than Soccer.
<i>CAB</i>	2250	6980	Bank (1000), Astronomy (750), Biology (500)	Unbalanced & highly overlapping for Astronomy & Biology

structure of the features in A_1 . It can be seen in $V_{CODIALING}$ above that feature (column) 3 has equal memberships in $C1$ and $C2$, reflecting the reality that the feature is shared equally by both co-clusters; and that all the other features are correctly assigned to their respective co-cluster. The former, as we have discussed in section 2.2, is something FSKWIC and Fuzzy CoDoK unable to achieve in this case because both of them perform feature ranking, instead of feature partitioning like in the case of CODIALING FCC. This suggests that CODIALING FCC can handle overlap better than the two existing algorithms. In this small toy problem, this existing problem of handling overlapping datasets only shows impact on the representation of the feature clustering. On larger scale datasets however, given the nature of co-clustering where object clustering and feature clustering are tied together, deviation in feature clustering representation can eventually affect the final accuracy of the object clustering. This is something that can be observed in our following experiments involving five large benchmark document datasets summarized in Table 3. In this case, documents are objects and words are features.

Classic3 can be downloaded from <ftp://ftp.cs.cornell.edu/pub/smart>, while *Binary* can be downloaded from <http://people.csail.mit.edu/jrennie/20Newsgroups>. The rests of the datasets are the subsets of the large benchmark web document datasets found in [16]. As indicated by the last column of Table 3, four out of five datasets involve certain degree of overlap. This allows us to see how the existing problem in dealing with overlaps can affect the overall performances of the fuzzy co-clustering algorithms.

Again for comparison, we performed the simulations on three algorithms: CODIALING FCC, FSKWIC, and Fuzzy CoDoK. In our implementation of FSKWIC, we followed the technique in [11] to adjust the negative distance. To avoid *NaN* value in updating u_{cj} , every adjustment was offset by 0.01. If there was any negative v_{cj} , we performed the following adjustment:

$$v_{cj} = v_{cj} + \left| \min_{c=1}^c v_{cj} \left[\sum_{q=1}^K \left(v_{cq} + \left| \min_{c=1}^c v_{cq} \right| \right) \right]^{-1} \right|, \text{ for all } j=1,2,3, \dots, K, \text{ if any } v_{cj} < 0. \text{ Our implement-}$$

ation of Fuzzy CoDoK follows strictly the original procedure detailed in [12]. All three algorithms were run 10 times for each dataset and the average results were recorded. For pre-processing, we used the *Matrix Creation* toolkit [17]. Stopwords were removed but no stemming was performed. Words occurring in less than 1% and more 90% of the number of documents in the dataset were removed. The Binary dataset used is the version with the header files removed. For all other datasets, the experiments were conducted on the complete versions of the documents. We use the normalized TF-IDF [18] to capture the association between document and word. The “document to document cluster” assignment was based on the maximum document membership. In all CODIALING FCC experiments, we set $T_u = T_v = 0.0001$. In the case of FSKWIC, we fixed $m = 1.02$ and $\partial_c = 0.1$. And finally for Fuzzy CoDoK, we set $T_u = 0.00001$ and $T_v = 1.5$. All these parameter values were empirically found to be the most optimum for the datasets involved. Three performance measures were recorded: precision, recall, and purity [9][18]. Table 4 shows the performance comparison in %.

Table 4. Performance Comparison

Datasets	CODIALING FCC			FSKWIC			Fuzzy CoDoK		
	Prec.	Recall	Purity	Prec.	Recall	Purity	Prec.	Recall	Purity
<i>Classic3</i>	98.6	98.5	98.5	99.02	98.87	98.94	98.6	98.3	98.4
<i>Binary</i>	83.56	83.4	83.4	79.97	75.66	75.66	73.26	73.04	73.04
<i>SM</i>	69.66	68.75	68.75	59.67	59.13	59.13	62.66	61.66	61.66
<i>SS</i>	80.31	80.32	80.32	71	70.3	70.3	61.59	61.32	61.32
<i>CAB</i>	68.14	67.87	73.55	52.48	55.19	71.64	63.91	62.6	72.35

It can be seen from Table 4 that all three algorithms perform equally effective when dealing with well-separated dataset such as *Classic3*. But once some overlapping clusters are introduced into the datasets, such is the case for *Binary*, *SM*, *SS*, and *CAB*, the accuracies of FSKWIC and Fuzzy CoDoK drop more significantly compared to those of CODIALING FCC. This indicates how the existing problem in dealing with overlapping datasets can affect the eventual categorization accuracies of FSKWIC and Fuzzy CoDoK. At the same time, the results also suggest that our dual-partitioning formulation incorporated in CODIALING FCC is effective to address this problem, that is, as we can see from Table 4, CODIALING FCC consistently achieve better accuracies than the two existing algorithms when a dataset contains overlapping clusters.

5 Conclusions

We have introduced a new fuzzy co-clustering algorithm, CODIALING FCC. We have discussed how the dual-partitioning approach incorporated in the new algorithm can tackle the problems in dealing with highly overlapping datasets. Our experiments on toy problem and large benchmark document datasets demonstrate the effectiveness of the new algorithm. With the ability to thrive in highly overlapping environment, and the fact that the algorithm has a linear time complexity, CODIALING FCC gives a promising direction for fuzzy co-clustering in general to be effectively applied in the real-world tasks for data mining. For future work, further investigation to provide a theoretical basis for the heuristic we applied in formulating CODIALING FCC may be considered.

References

1. Mitra, S., Acharya, T.: Data Mining Multimedia, Soft Computing, and Bioinformatics. John Wiley & Sons Inc., New Jersey (2003)
2. Han, J., Kamber, M.: Data Mining Concepts and Techniques. Academic Press, London (2001)
3. Zamir, O., Etzioni, O.: Web Document Clustering: A Feasibility Demonstration. Proc. of the Twenty First Annual International ACM SIGIR Conf. on R&D in Information Retrieval, (1998) 46-54
4. Madeira, S.C., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE/ACM Trans. on Comp. Biology and Bioinf., 1 (2004) pp. 24-45

5. Ertöz, L., Steinbach, M., Kumar, V.: Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data. Proc. of SIAM International Conf. on Data Mining, (2003)
6. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-Theoretic Co-clustering. Proc of the Ninth ACM SIGKDD International Conf. on KDD, (2003) 89-98
7. Banerjee, A., Dhillon, I.S., Modha, D.S.: A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. Proc. of the Tenth ACM SIGKDD International Conf. on KDD, (2004) 509-514
8. Cho, H., Dhillon, I.S., Guan, Y., Sra, S.: Minimum Sum-squared Residues Co-clustering of Gene Expression Data. Proc. of the Fourth SIAM International Conf. on Data Mining, (2004)
9. Mandhani, B., Joshi, S., Kumnamuru, K.: A Matrix Density Based Algorithm to Hierarchically Co-Cluster Documents and Words. Proc. of the Twelfth Int. Conference on WWW, (2003) 511-518
10. Zadeh, L.A.: Fuzzy Sets. Information and Control, 8 (1965) pp.
11. Frigui, H., Nasraoui, O.: Simultaneous Clustering and Dynamic Keyword Weighting for Text Documents. In: Berry, M.W. (ed): Survey of Text Mining. Springer-Verlag (2004), 45-72
12. Kumnamuru, K., Dhawale, A., Krishnapuram, R.: Fuzzy Co-clustering of Documents and Keywords. IEEE International Conf. on Fuzzy Systems, 2 (2003) 772-777
13. Ruspini, E.: A new approach to clustering. Information and Control, 15 (1969) 22-32
14. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, (1981)
15. Oh, C.H., Honda, K., Ichihashi, H.: Fuzzy Clustering for Categorical Multivariate Data. Proc. of Joint 9th IFSA World Congress and 2nd NAFIPS Inter. Conf., (2001) 2154-2159
16. Sinka, M.P., Corne, D.W.: A Large Benchmark Dataset for Web Document Clustering. In: Abraham, A., et al (eds.): Soft Computing Systems: Design, Management and Applications. IOS Press, Amsterdam (2002) 881-892
17. Dhillon, I.S., Fan, J., Guan, Y.: Efficient Clustering of Very Large Document Collections. In: Grossman, R.L., et al (eds): Data Mining for Scientific and Engineering Applications. Kluwer Academic Publishers (2001) 357-382
18. Yates, R.B., Neto, R.R.: Modern Information Retrieval. ACM Press, New York (1999)

Quantum-Behaved Particle Swarm Optimization Clustering Algorithm

Jun Sun, Wenbo Xu, and Bin Ye

Center of Intelligent and High Performance Computing,
School of Information Technology, Southern Yangtze University,
No. 1800, Lihudadao Road, Wuxi, 214122 Jiangsu, China
{sunjun_wx, xwb_sytu}@hotmail.com

Abstract. Quantum-behaved Particle Swarm Optimization (QPSO) is a novel optimization algorithm proposed in the previous work. Compared to the original Particle Swarm Optimization (PSO), QPSO is global convergent, while the PSO is not. This paper focuses on exploring the applicability of the QPSO to data clustering. Firstly, we introduce the K-means clustering algorithm and the concepts of PSO and QPSO. Then we present how to use the QPSO to cluster data vectors. After that, experiments are implemented to compare the performance of various clustering algorithms. The results show that the QPSO can generate good results in clustering data vectors with tolerable time consumption.

1 Introduction

Roughly speaking, clustering procedures yield a data description in terms of clusters or groups of data points that possess strong internal similarities. Formal clustering procedures use a criterion function, such as the sum of the squared distances from the cluster centers, and seek the grouping that extremizes the criterion function. Clustering analysis has become an important technique in exploratory data analysis, pattern recognition, neural computing, data mining, image segmentation and mathematical programming. Generally, clustering algorithm can be grouped into two main categories, namely supervised and unsupervised. Among unsupervised clustering algorithms, the most important algorithms are K-means [6], [7], ISODATA [2], and Learning Vector Quantizers (LVQ) [9]. K-means algorithm is a very popular algorithm. It is used for clustering where clusters are of crisp and spherical. Here, clustering is based on minimization of overall sum of the squared error between each pattern and the corresponding cluster center. Although clustering algorithms are usually grouped into unsupervised and supervised, efficient hybrids have been developed that performs both supervised and unsupervised learning.

Recently, a novel variant of Particle Swarm Optimization (PSO), Quantum-behaved Particle Swarm Optimization (QPSO) has been proposed [11], [12], [13]. QPSO is a global convergent optimization algorithm. This paper explores the applicability of the QPSO to data clustering. In the process of doing so, the objective of the paper is to use original PSO and QPSO to cluster arbitrary data and make a performance comparison.

The rest part of the paper is structured as follows: Section 2 presents a brief overview of the K-means algorithm. PSO and QPSO are introduced in section 3. QPSO clustering algorithms are proposed in section 4. Section 5 presents experiment results on five widely used data sets and the paper is concluded in Section 6.

2 K-Means Clustering

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume c clusters) fixed a priori. For the purpose of this paper, we define the following symbols:

- D denotes the input dimension, i.e. the number of parameters of each data vector;
- N denotes number of data vectors to be clustered;
- c denotes the number of cluster centroids, i.e. the number of clusters to be formed;
- Z_p denotes the p^{th} data vector;
- \mathbf{m}_j denotes the centroid vector of cluster j ;
- $n_j = |C_j|$ is the number of data vectors in cluster j ;
- C_j is the subset of data vectors that form cluster j ;

Using the above notation, the standard K-means algorithm is summarized as follows.

- (1) Make initial guesses for the means $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_c$;
- (2) **Until** a stopping criterion is satisfied
 - (a) Use the estimated means to classify the samples into clusters, whose centroids are determined using

$$d(Z_p, \mathbf{m}_j) = \sqrt{\sum_{q=1}^D (Z_{pq} - m_{jq})^2} \tag{1}$$

That is, for each data vector, assign the vector to the cluster with the closest centroid vector;

- (b) **for j from 1 to k**
 Replace \mathbf{m}_j with the mean of all of the samples for cluster j , i.e.

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{\forall Z_p \in C_j} Z_p \tag{2}$$

endfor
enduntil

The K-means clustering process can be stopped when any one of the following criteria is satisfied: when the maximum number of iterations has been exceeded, when there is little change in the centroid vectors over a number of iterations, or when there are no cluster membership changes.

3 Quantum-Behaved Particle Swarm Optimization

Particle Swarm Optimization (PSO), a population-based random search technique, originally proposed by J. Kennedy and R. Eberhart [8], has become a most fascinating branch of evolutionary computation. The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm theory. In the Standard PSO [15] with population size M , each individual is treated as a volume-less particle in the n -dimensional, with the position and velocity of i^{th} particle represented as $X_i = (X_{i1}, X_{i2}, \dots, X_{in})$ and $V_i = (V_{i1}, V_{i2}, \dots, V_{in})$. The particle moves according to the following equation:

$$V_{id} = w \cdot V_{id} + c_1 \cdot r_1 \cdot (P_{id} - X_{id}) + c_2 \cdot r_2 \cdot (P_g - X_{id}) \tag{3}$$

$$X_{id} = X_{id} + V_{id}, \quad (d = 1, 2, \dots, n) \tag{4}$$

where c_1 and c_2 are called acceleration coefficients and r_1, r_2 are two numbers distributed uniformly in $[0,1]$, i.e. $r_1, r_2 \sim U(0,1)$. Parameter w is the inertia weight introduced to accelerate the convergence speed of PSO. Vector $P_i = (P_{i1}, P_{i2}, \dots, P_{in})$ is the best previous position (the position giving the best fitness value) of particle i called personal best position (***pbest***), and vector $P_g = (P_{g1}, P_{g2}, \dots, P_{gn})$ is the position of the best particle among all the particles in the population and called global best position (***gbest***).

In Quantum-behaved Particle Swarm Optimization (QPSO), the particle moves according to the following equation:

$$P_{id} = \varphi \cdot P_{id} + (1 - \varphi) \cdot P_{gd}, \quad \varphi = rand() \tag{5}$$

$$X_{id} = p_{id} \pm \alpha \cdot |mbest_d - X_{id}| \cdot \ln(1/u), \quad u \sim U(0,1) \tag{6}$$

where $mbest$ is the mean best position among the particles. That is

$$mbest = \frac{1}{M} \sum_{i=1}^M P_i = \left(\frac{1}{M} \sum_{i=1}^M P_{i1}, \frac{1}{M} \sum_{i=1}^M P_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M P_{in} \right) \tag{7}$$

The p_{id} , a stochastic point between P_{id} and P_{gd} , is the local attractor on the d^{th} dimension of the i^{th} particle, φ is a random number distributed uniformly in $[0,1]$, u is another uniformly-distributed random number in $[0,1]$ and α , called Contraction-Expansion Coefficient, is a parameter of QPSO. The Quantum-behaved Particle Swarm Optimization (QPSO) Algorithm in [12], [13] is described as follows.

1. Initialize an array of particles with random positions inside the problem space;
2. Determine the mean best position among the particles by Eq(7);
3. Evaluate the desired objective function (take minimization problems for example) for each particle and compare with the particle's previous best values: If the current fitness value is less than previous best value, then set the previous best value to the current value. That is, if $f(X_i) < f(P_i)$, then $X_i = P_i$;

4. Determine the current global position minimum among the particle's best positions. That is: $g = \arg \min_{1 \leq i \leq M} (f(P_i))$ (for maximization problem);
5. Compare the current global position to the previous global: if the fitness value of the current global position is less than that of the previous global position; then set the global position to the current global;
6. For each dimension of the particle, get a stochastic point between p_{id} and p_{gd} by Eq(5);
7. Update position by stochastic Eq(6);
8. Repeat steps 2-7 until a stop criterion is satisfied **OR** a pre-specified number of iterations are completed.

4 QPSO Clustering

In the QPSO clustering technique, a single particle represents the c cluster centroid vectors. That is, the position vector X_i of particle i is constructed as follows:

$$X_i = (\mathbf{m}_{i1}, \dots, \mathbf{m}_{ij}, \dots, \mathbf{m}_{ic}) \tag{8}$$

where \mathbf{m}_{ij} refers to the j^{th} cluster centroid vector of the i^{th} particle in cluster C_{ij} . Therefore, a swarm represents a number of candidate clustering schemes for the current data vectors. The fitness of the particle is measured as the quantization error,

$$J = \sum_{j=1}^c [\sum_{\forall Z_p \in C_{ij}} d(Z_p, m_j) / |C_{ij}|] / c \tag{9}$$

Using the QPSO algorithm, data vectors can be clustered as follows:

QPSO Clustering Algorithm

1. Initialize the swarm, in which each particle contain c randomly selected centroids;
2. **For** t=1 to MAXITER do
 - (1) Calculate the mbest of swarm using Eq(7);
 - (1) **For** each particle i **do**
 - (2) **For** each data vector Z_p
 - (I). Calculate the Euclidean distance $d(Z_p, \mathbf{m}_{ij})$ of Z_p to all cluster centroids C_{ij} ;
 - (II). Assign Z_p to cluster C_{ij} such that $d(Z_p, \mathbf{m}_{ij}) = \min_{\forall j=1,2,\dots,c} \{d(Z_p, \mathbf{m}_{ij})\}$;
 - (III). Calculate the fitness using Eq(9);
 - Endfor**
 - Endfor**
 - (3) Update the global best and the personal best positions;
 - (4) Update the cluster centroids (the position of the particle) using Eq(5) and Eq(6);
- Endfor**

where MAXITER is the maximum number of iterations.

Compared to the QPSO clustering, K-means algorithm tends to converge faster but usually with less accurate clustering. The QPSO clustering algorithms can be improved by seeding the initial swarm with the result of the K-means algorithm. The hybrid algorithm first executes the K-means once. In our proposed method, the K-means clustering executed in the hybrid algorithm is terminated when the maximum number of iterations t_{max} is exceeded. The result of the K-means algorithm is then used as one of the particles, while the rest of the swarm is initialized randomly. After that, the QPSO clustering algorithm described above is executed.

5 Experiment Results

This section presents comparison results of the K-means, the PSO, the Hybrid Clustering with the PSO and K-means (H-PSO-K), the QPSO and the Hybrid Clustering with the QPSO and K-means (H-QPSO-K) on five well-known classification problems.

The main purpose is to compare the quality of the respective clustering algorithms, where quality is measured according to the following three criteria: (1). The quantization error as defined in Eq(9), where the objective is to minimize the quantization error; (2).The intra-cluster distances, i.e. the distance between data vectors within a cluster, where the objective is to minimize the intra-cluster distances; (3). The inter-cluster distances, i.e. the distance between the centroids of the clusters, where the objective is to maximize the distance between clusters.

For all the results reported averages over 30 simulations are given. Each simulation runs for 1000 function evaluations, and the PSO or QPSO algorithm used 10 particles. For PSO, $w=0.72$ and $c_1=c_2=1.49$ [4], [10]. These values were chosen to ensure good convergence. For the experiments of QPSO, the value of α varies linearly from 1.0 to 0.5 on the course of running.

The classification problems used for the purpose of this paper are

(1). Artificial problem 1: The problem follows the following classification rules:

$$class = \begin{cases} 1 & \text{if } (z_1 \geq 0.7) \text{ or } (z_1 \leq 0.3) \text{ and } (z_2 \geq -0.2 \cdot z_1) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

A total of 400 data vectors were randomly created, with $z_1, z_2 \sim U(0,1)$.

(2). Artificial problem 2: The problem is 2-dimensional with 4 unique classes. A total of 600 patterns were drawn from four independent bivairate normal distribution, where classes were distributed according to

$$N_2 \left(\mu - \begin{pmatrix} m_i \\ 0 \end{pmatrix}, \Sigma = \begin{bmatrix} 0.50 & 0.05 \\ 0.05 & 0.50 \end{bmatrix} \right) \quad (11)$$

for $i=1, \dots, 4$, where μ is the mean vector and Σ is the covariance matrix. In the problem, $m_1 = -3, m_2 = 0, m_3 = 4$ and $m_4 = 6$.

(3). Iris plants database: This is a well-understood database with 4 inputs, 3 classes and 150 data vectors.

- (4). Wine: This is a classification problem with “well behaved” class structures. There are 13 inputs, 3 classes and 178 data vectors.
- (5). Breast cancer: The Wisconsin breast cancer database contains 9 relevant inputs and 2 classes. The objective is to classify each data vector into benign or malignant tumors.

Table 1. Comparison of K-means, PSO, H-PSO-K, QPSO and H-QPSO-K

Data Set	Clustering algorithms	Quantization Error	Intra-cluster Distance	Inter-cluster Distance
Artificial 1	K-Means	1.0185±0.0152	2.0372±0.0303	1.7481±0.0286
	PSO	0.9855±0.0236	1.9853±0.0763	1.6508±0.1495
	H-PSO-K	0.9836±0.0274	1.9736±0.0697	1.6721±0.1407
	QPSO	0.9538±0.0179	1.9187±0.0365	1.6985±0.0780
	H-QPSO-K	0.9488±0.0182	1.9008±0.0317	1.6762±0.0728
Artificial 2	K-Means	0.2681±0.0014	0.9079±0.0264	0.7843±0.0244
	PSO	0.2521±0.0011	0.8672±0.0236	0.8159±0.0195
	H-PSO-K	0.2514±0.0009	0.8647±0.0186	0.8147±0.0176
	QPSO	0.2405±0.0010	0.8538±0.0228	0.8235±0.0166
	H-QPSO-K	0.2343±0.0009	0.8332±0.0173	0.8179±0.0154
Iris	K-Means	0.6502±0.1223	3.3921±0.2647	0.9125±0.0915
	PSO	0.7456±0.0560	3.5645±0.1954	0.8952±0.0845
	H-PSO-K	0.6337±0.1333	3.3018±0.2013	0.8564±0.0976
	QPSO	0.6315±0.1126	3.2453±0.1952	0.9016±0.0812
	H-QPSO-K	0.5295±0.1523	3.3853±0.2416	0.8792±0.0706
Wine	K-Means	1.4214±0.1356	4.3854±0.2564	1.1346±0.1365
	PSO	1.3456±0.0524	4.8561±0.3216	2.8697±0.2157
	H-PSO-K	1.1794±0.1584	4.2597±0.4125	2.6151±0.1146
	QPSO	1.0452±0.0983	4.3513±0.4584	2.9135±0.1023
	H-QPSO-K	1.0849±0.1842	4.2387±0.5146	2.7172±0.1230
Breast-cancer	K-Means	2.0145±0.0631	6.4897±0.3354	1.7562±0.2194
	PSO	2.6538±0.1567	7.1431±0.3561	3.4267±0.2058
	H-PSO-K	1.9675±0.1146	6.3250±0.3860	3.3568±0.1250
	QPSO	1.8643±0.1253	6.0425±0.3874	3.4146±0.1168
	H-QPSO-K	1.6325±0.1846	6.2168±0.3946	3.5125±0.1219

Table 1 summarizes the results obtained from the three clustering algorithms as well as two hybrid algorithms for the problems above. Each value reported is averaged over 30 simulations, with standard deviation to indicate the range of values to which the algorithms converge.

As of the fitness of solutions, i.e. the quantization error, the Hybrid of QPSO and K-means (H-QPSO-K) has the smallest average quantization error. For Artificial problem 1 and artificial problem 2, H-QPSO-K has better results in quantization

errors. For Iris problem, K-means algorithm performs better than the PSO, but worse than three other algorithms. The QPSO and H-QPSO-K both have better quantization errors than the PSO and H-PSO-K, respectively. For Wine problem, K-means has worst quantization errors. The two algorithms with the QPSO have better results than the two with the PSO, but the QPSO performs better than its hybrid with K-means (H-QPSO-K). For Breast Cancer problem, the comparison results are similar to that of Iris problem, with H-QPSO-K having best quantization error.

As of inter- and intra-cluster distances, the latter ensures compact clusters with little deviation from the cluster centroids, while the former ensures larger separation between the different clusters. With reference to these criteria, the H-QPSO-K succeeded most in finding clusters with larger separation than other clustering algorithm in Breast cancer problem. For Iris problem, the clusters found by K-means algorithm have largest separation. The QPSO finds more separated clusters than any other clustering algorithms for Wine problem. K-means finds the clusters with largest separation for artificial problem 1, while QPSO generates the most separated clusters for artificial problem 2. It is the QPSO approaches that succeeded in forming the most compact clusters for all problems.

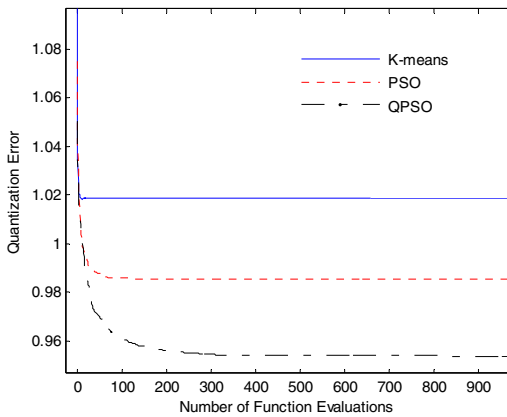


Fig. 1. Convergence processes of K-means, PSO and QPSO on Artificial problem 1

Figure 1 shows the convergence processes of three algorithms including K-means, PSO and QPSO clustering methods on artificial problem 1. It can be seen that although K-means converges rapidly, it always sticks into local optima. QPSO could find the global optima more efficiently and effectively than PSO and K-means. Thus we may conclude that QPSO will be a promising solver for clustering problems.

6 Conclusions and Future Work

In this paper, we explored the applicability of the QPSO to data clustering. For data clustering, it is natural that QPSO has overall better performance than K-means and

PSO clustering algorithms, because the QPSO is a global convergent optimization algorithm, while the PSO is not, according to the criteria used by Van de Bergh [3]. The experiment results also testified that hybrid of QPSO and K-means approach could improve QPSO clustering algorithm considerably. Our future work will focus on the application of the QPSO to more real world problems such as image segmentation. In doing so, reduction of time consumption for QPSO clustering is necessary. Parallization is a good technique for the computational time problem.

References

1. Andrews, H. C.: Introduction to Mathematical Techniques in Pattern Recognition. John Wiley & Sons, New York (1972)
2. Ball, G., Hall, D.: A Clustering Technique for Summarizing Multivariate Data. Behavioral Science, Vol. 12, (1967) 153-155
3. Van den Bergh, F.: An Analysis of Particle Swarm Optimizers. PhD Thesis. University of Pretoria, South Africa (2001)
4. Clerc, M.: The Swarm and Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. Proc. 1999 Congress on Evolutionary Computation. Piscataway, NJ (1999) 1951-1957
5. Fisher, D.: Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learning, Vol. 2. Springer Netherlands (1987) 139-172
6. Forgy, E.: Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classification. Biometrics, Vol. 21. (1965) 768-769
7. Hartigan, J.A.: Clustering Algorithms", John Wiley & Sons, New York (1975)
8. Kennedy, J., Eberhart, R. C.: Particle Swarm Optimization. Proc. 1995 IEEE International Conference on Neural Networks, Vol. IV. Piscataway, NJ (1995) 1942-1948
9. Kohonen, T.: Self-Organizing Maps. Springer Series in Information Sciences, Vol 30. Springer-Verlag (1995)
10. Van der Merwe, D.W., Engelbrecht, A.P.: Data Clustering Using Particle Swarm Optimization. Proc. 2003 Congress on Evolutionary Computation, Vol. 1. Piscataway NJ (2003) 215-220
11. Sun, J., Feng, B., Xu, W.-B.: Particle Swarm Optimization with Particles Having Quantum Behavior. Proc. 2004 Congress on Evolutionary Computation. Piscataway, NJ (2004) 325-331
12. Sun, J., Xu, W.-B., Feng, B.: A Global Search Strategy of Quantum-behaved Particle Swarm Optimization. Proc. 2004 IEEE Conference on Cybernetics and Intelligent Systems, Singapore (2004) 111-116
13. Sun, J., Xu, W.-B., Feng, B.: Adaptive Parameter Control for Quantum-behaved Particle Swarm Optimization on Individual Level. Proc. 2005 IEEE International Conference on Systems, Man and Cybernetics. Piscataway, NJ (2005) 3049-3054
14. Shi, Y., Eberhart, R.C.: Empirical Study of Particle Swarm Optimization. Proc. 1999 Congress on Evolutionary Computation. Piscataway, NJ (1999) 1945-1950
15. Shi, Y., Eberhart, R.C.: A Modified Particle Swarm. Proc. 1998 IEEE International Conference on Evolutionary Computation. Piscataway, NJ (1998) 69-73

Clustering Mixed Data Based on Evidence Accumulation

Huilan Luo^{1,2}, Fansheng Kong¹, and Yixiao Li¹

¹ Artificial Intelligence Institute, Zhejiang University, Hangzhou 310027, China
d051luohuilan@zju.edu.cn

² Institute of Information Engineering, Jiangxi University of Science and Technology, Gangzhou 341000, China

Abstract. An Evidence-Based Spectral Clustering (EBSC) algorithm that works well for data with mixed numeric and nominal features is presented. A similarity measure based on evidence accumulation is adopted to define the similarity measure between pairs of objects, which makes no assumptions of the underlying distributions of the feature values. A spectral clustering algorithm is employed on the similarity matrix to extract a partition of the data. The performance of EBSC has been studied on real data sets. Results demonstrate the effectiveness of this algorithm in clustering mixed data tasks. Comparisons with other related clustering schemes illustrate the superior performance of this approach.

1 Introduction

Traditional clustering methodologies assume features are numeric valued, and represent data objects as points in a multidimensional metric space. These classical approaches adopt distance metrics, such as Euclidean and Mahalanobis measures, to define similarity between objects. On the other hand, conceptual clustering systems use conditional probability estimates as a means for defining the relation between groups or clusters. Systems like COBWEB [9] and its derivatives use the Category Utility (CU) measure [14], which has its roots in information theory. The measure partitions a data set in a manner that maximizes the probability of correctly predicting a feature value in a given cluster. These measures are tailored for nominal attributes. As application areas have grown from the scientific and engineering domains to the medical, business, and social domains, a majority of the useful data is described by a combination of numeric and nominal valued features. Attempts to develop criterion functions for mixed data have not been very successful, mainly because of the differences in the characteristics of these two kinds of data [3].

To cluster mixed data sets COBWEB/3 [6] and ECOBWEB [7] use modifications of the CU measure to handle numeric attributes. However, the numeric form of the CU measure has a number of limitations. First, it assumes that feature values are normally distributed. Another limitation of the numeric CU measure is that it does not take into account the actual distance between object values in determining class structure. AUTOCLASS [8] imposes a classical finite mixture distribution model on the mixed data and uses a Bayesian method to derive the most probable class distribution for the data given prior information. The intra-class mixture probability distribution function (*pdf*) is a product of individual or covariant attribute *pdfs*, such as the Ber-

noulli distributions for nominal attributes, Gaussian distributions for numeric attributes, and Poisson distributions for number counts. AUTOCLASS also suffers from the over fitting problem associated with the maximum likelihood optimization methods for probabilistic models.

Huang [2] [1] presents two algorithms, k -modes and k -prototypes, which extend k -means paradigm to nominal domains and domains with mixed numeric and nominal values whilst preserving its efficiency. The k -modes algorithm uses a simple matching dissimilarity measure to deal with nominal data. It replaces the means of clusters with modes, and uses a frequency-based method to update modes in the clustering process. The k -prototypes algorithm integrates the k -means and k -modes algorithms through the definition of a combined dissimilarity measure to allow for clustering objects described by mixed attributes. In [3], a Similarity-Based Agglomerative Clustering (SBAC) algorithm is proposed to clustering mixed data, which adopts a similarity measure that gives greater weight to uncommon feature value matches in similarity computations. But experimental results in [3] show they produce better results for artificial data than they do for real data.

The focus of this paper is on developing unsupervised learning techniques that exhibit good performance with mixed data. The core of the methodology is based on evidence accumulation proposed by Fred [4]. First, we obtain N clusterings by running k -means N times with random initializations on the pure numeric subset and m clusterings for m nominal attributes. Then by taking the co-occurrences of pairs of patterns in the same cluster as votes for their association, the data partitions are mapped into a similarity matrix of patterns. Based on this similarity matrix, we apply a spectral clustering method NJW [5] to obtain the final clustering result.

2 The Similarity Measure Based on Evidence Accumulation

The idea of evidence accumulation clustering is to combine the results of multiple clusterings into a single data partition, by viewing each clustering result as an independent evidence of data organization [4].

Similarity between objects can be estimated by the number of clusters shared by two objects in all the partitions of a clustering ensemble. This similarity definition expresses the strength of co-association of objects by a matrix containing the values [10]:

$$S(i, j) = S(x_i, x_j) = \frac{1}{H} \sum_{k=1}^H \delta(\pi_k(x_i), \pi_k(x_j)) \quad (1)$$

Where H denotes the number of clusterings and $\pi_k(x_i)$, $\pi_k(x_j)$ are the cluster labels for x_i , x_j respectively in the k -th clustering.

2.1 Computing Similarity for Numeric Attributes

Initially the pure numeric data set is decomposed into a large number of compact clusters. The k -means algorithm performs this decomposition with N clusterings obtained by N random initializations of the k -means algorithm. Then we take the

co-occurrences of pairs of patterns in the same cluster as votes for their association. So the N clusterings are mapped into a $n \times n$ co-association matrix, as follows:

$$S_n(i, j) = \frac{n_{ij}}{N} = \frac{\sum_{l=1}^N C^l(i, j)}{N} \quad (2)$$

Where n_{ij} is the number of times the pattern pair (i, j) is assigned to the same cluster among the N clusterings, and $C^l(i, j) = 1$ if the pattern pair (i, j) is in the same cluster of the l -th clustering, else $C^l(i, j) = 0$. Evidence accumulated over the N clusterings, according to the above equation (2), induces a new similarity measure between patterns.

2.2 Computing Similarity for Nominal Features

For the nominal attributes, if we consider attribute values as cluster labels, each attribute with its attribute values gives a clustering on the data set without considering other attributes [12].

So if the data set has m nominal attributes, we can obtain m clusterings. As the above, we take the co-occurrences of pairs of patterns in the same cluster as votes for their association. So the m clusterings are mapped into a $n \times n$ co-association matrix, as follows:

$$S_c(i, j) = \frac{n_{ij}}{m} = \frac{\sum_{l=1}^m C^l(i, j)}{m} \quad (3)$$

Where n_{ij} is the number of times the pattern pair (i, j) is assigned to the same cluster among the m clusterings.

2.3 Aggregating Similarity Contributions from Numeric and Nominal Features

We combine these two matrixes to produce the final similarity matrix S .

$$S = S_n + \alpha S_c \quad (4)$$

where α is a user specified parameter. If $\alpha > 1$, then the algorithm gives a higher importance to the nominal features, and when $\alpha < 1$, it gives a higher importance on the numeric attributes. If we have some prior information about the data, we can use it to select the parameter. In the experimental results presented in section 5 we only simply set $\alpha = 1$.

The final data partition is obtained by applying the spectral clustering method that clusters points using eigenvectors of this similarity matrix S .

3 Spectral Clustering Technique

Spectral clustering techniques have seen an explosive development and proliferation over the past few years. They promise to become strong competitors for other clustering methods. Spectral methods are attractive because they are easy to implement and are reasonably fast (for sparse data sets up to several thousands). Also they do not intrinsically suffer from the problem of local optima and need not make harsh simplifying assumptions. (e.g., that the density of each cluster is Gaussian). In this paper we use the spectral clustering algorithm NJW in [5] and modified a little. For completeness of the text we briefly review their algorithm.

Given a set of n points $X = \{x_1, x_2, \dots, x_n\}$ in R^p , cluster them into C clusters as follows:

1. Set the diagonal elements $S_{ii} = 0$ in the similarity matrix $S \in R^{n \times n}$ of X .
2. Define D to be a diagonal matrix with $D_{ii} = \sum_{j=1}^n S_{ij}$ and construct the normalized similarity matrix $L = D^{-1/2} S D^{-1/2}$.
3. Manually select a desired number of groups C .
4. Find u_1, u_2, \dots, u_C the C largest eigenvectors of L , and form the matrix $U = [u_1, \dots, u_C] \in R^{n \times C}$.
5. Re-normalize the rows of U to have unit length yielding $Y \in R^{n \times C}$, such that $Y_{ij} = U_{ij} / (\sum_j u_{ij}^2)^{1/2}$.
6. Treat each row of Y as a point in R^C and cluster them into C clusters via k -means.
7. Assign the original point x_i to cluster j if and only if the corresponding row i of the matrix Y was assigned to cluster j .

4 Clustering Mixed Data Based on Evidence Accumulation

Let the collection of data $X = \{x_1, x_2, \dots, x_n\}$ can be represented as a set of points in a p -dimensional vector space, with m nominal attributes and d numeric attributes, $m + d = p$. Without loss of generality, write $x_i = \{x_i^{A_1}, \dots, x_i^{A_d}, x_i^{B_1}, \dots, x_i^{B_m}\}$, where the first d attributes are numeric attributes and the next m attributes are nominal attributes. Then the overall Evidence-Based Spectral Clustering (EBSC) algorithm for clustering mixed data is summarized below:

1. Find X_1, X_2, \dots, X_d the d numeric attribute columns of X , and form the matrix $A = [X_1, X_2, \dots, X_d]$;
2. Do N times for pure numeric data set A : Randomly select k cluster centers. Run the k -means algorithm on A with the above initialization, and produce a partition P . Update the S matrix: for each pattern pair, (i, j) , in the same cluster in P , set $S(i, j) = S(i, j) + 1/N$.

3. For each categorical attribute column X^B , According to the categorical attribute B_j update the S matrix: for each pattern pair (i, j) , if $x_i^{B_j} = x_j^{B_j}$, set $S(i, j) = S(i, j) + 1/m$.
4. Detect consistent clusters in the similarity matrix S using the NJW algorithm.

5 Experimental Results

Our goals for performing empirical studies with EBSC were twofold: 1) to gain a better understanding of the characteristics of the evidence accumulation similarity measure and 2) to compare the performance of EBSC with existing clustering mixed data algorithm. To achieve this, we used two kinds of data: artificially generated data, with mixed nominal and numeric features and two real data sets, the Australian Credit Approval Data Set and the heart disease data. We evaluated the performance of clustering algorithms by matching the detected and the known partitions of the datasets [15].

5.1 Artificial Data

To gain a better understanding of the characteristics of the evidence accumulation similarity measure, we follow the approach taken in [3] and graph the percent of the numeric feature values and the percent of the nominal feature values in each class were corrupted versus the error rate of the EBSC algorithm respectively.

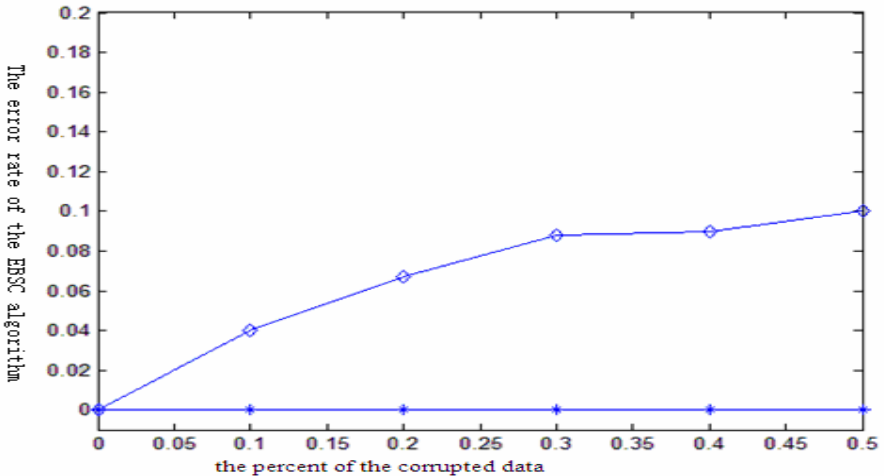


Fig. 1. The error rate of the EBSC algorithm versus the percent of the corrupted data (the star symbol presents the noisy nominal features but clean numerical features; the diamond symbol presents the noisy numeric features but clean nominal features)

The artificial data set had 180 data points, equally distributed into three classes. Each data point was described using four features: two nominal and two numeric features. In our experiments, we created data sets that had corrupted numeric features, or corrupted nominal features, but not both. Our primary goal here was to study how our algorithm was biased toward numeric and nominal features. Ten data sets were created as variations of the data set above by successively mixing up 10 percent to 50 percent of the numeric feature values or nominal feature values for one class with the other two classes.

The error rates of the EBSC algorithm for different data sets with different percents of corrupted data are plotted in Fig. 1. The star symbol “*” represent the effects of increasing noise added to the nominal attributes and the diamond symbol represent the effects of adding noise to the numeric attributes. It was observed that as the noise levels in the numeric features increased, our algorithm deteriorated a little, the error rate increased from 0 to 0.1. There was little deterioration when the nominal features were corrupted. From the graph, we can see the evidence that our algorithm is robust and showed very little deterioration in performance as the degradation levels were increased.

5.2 Real World Data

1. Australian Credit Approval Data Set

This data set concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. The data set has 690 instances, each being described by 6 numerical, 8 nominal attributes and 1 class attribute. The class labels available in the UCI repository [13] were used for post evaluation but they were not used as a feature in the clustering process. The numbers of data samples in the two classes are 307 and 383, respectively.

We set $k=28$, $N=100$ and use the proposed algorithm EBSC to cluster the dataset. The clustering accuracies by the proposed algorithm EBSC, SBAC [3] and k -prototypes [1] clustering algorithms are listed in Table 1. For the data set, the proposed algorithm achieved the best clustering result with clustering accuracy 0.82899.

Table 1. Performance Comparison of Different Clustering Algorithms on Credit Data Set

Algorithm	Clustering Accuracy
EBSC	0.82899
SBAC	0.76
k-prototypes	0.742

2. Heart Disease Data

The heart disease data, generated at the Cleveland Clinic, contains a mixture of nominal and numeric features. The data set consists of 303 patient instances defined by 13 features. Five of these are numeric-valued features, and eight are nominal-valued features. The data comes from two classes: people without heart disease and people with different degrees (severity) of heart disease. There are also a few missing values in this dataset. In our experiment the class labels available in the UCI repository were used for post evaluation but they were not used as a feature in the clustering

process. The 7 instances with missing values were removed in our experiment; therefore only 296 instances were used.

We use the proposed algorithm and set $k=40$, $N=100$, to cluster the data. The clustering accuracies by the proposed algorithm EBSC, SBAC [3], and k -prototypes [1] clustering algorithms are listed in Table 2. For the data set, the proposed algorithm achieved the best clustering result with clustering accuracy 0.81308.

Table 2. Performance Comparison of Different Clustering Algorithms on Heart Disease Data Set

Algorithm	Clustering Accuracy
EBSC	0.81308
k-prototypes	0.810
SBAC	0.752

From the experimental results on the two real data sets, we can see evidences that our algorithm can obtain comparable and reliable results.

6 Conclusions

Limitations of earlier methodologies and criterion functions in dealing with data described by mixed nominal and numeric features prompted us to look for a criterion function that would give better performance in clustering mixed data. We have demonstrated that the similarity measure based on evidence accumulation works well with data described by mixed type features. The measure makes no assumptions of the underlying distributions of the feature values. Illustrative experiments are presented to demonstrate the properties of the similarity measure. The similarity measure is then incorporated into a spectral clustering algorithm, NJW to obtain the final partition. SBAC and its performance is studied on real and artificially generated data. The results show that EBSC works well with mixed data and is equally robust to noisy numeric and nominal features.

References

1. Huang, Z.: Clustering Large Data Sets with Mixed Numeric and Categorical Values. Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining, (PAKDD), Singapore (1997) 21-34
2. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery* 2 (1998) 283-304
3. Li, C., Biswas, G.: Unsupervised Learning with Mixed Numeric and Nominal Data. *IEEE Trans. Knowl. Data Eng.* 14 (2002) 673-690
4. Fred, A.L.N., Jain, A.K.: Data Clustering using Evidence Accumulation. Proc. of the 16th Intl. Conference on Pattern Recognition ICPR 2002, Quebec City (2002) 276-280
5. Ng, A.Y., Jordan, M.I., Weiss, Y.: On Spectral Clustering: Analysis and an algorithm. *NIPS* (2001) 849-856

6. McKusick, K.B., Thompson, K.: COBWEB/3: A portable implementation Moffett Field, CA: NASA Ames Research Center (1990)
7. Reich, Y., Fennes, S.: The formation and use of abstract concepts in design. In: Fisher, D., Pazzani, M., Langley, P. (eds.): *Concept Formation: Knowledge and Experience in Unsupervised Learning*. Morgan Kaufmann, Los Altos, CA (1991) 323-353
8. Cheeseman, P., Stutz, J.: Bayesian classification (AutoClass): Theory and results. In: Fayyad, U.M., Dietetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): *Advances in Knowledge Discovery and Data Mining*. AAAI Press/The MIT Press (1996) 153-180
9. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. *Machine Learning* 2 (1987) 139-172
10. Topchy, A., Jain, A.K., Punch, W.: A Mixture Model of Clustering Ensembles. *Proceedings of the SIAM International Conference on Data Mining*, Lake Buena Vista, Florida (2004) 22-24
11. Jain, A.K., Fred, A.: Evidence accumulation clustering based on the k-means algorithm. *LNCS* 2396 (2002) 442-451
12. He, Z., Xu, X., Deng, S.: A cluster ensemble method for clustering categorical data. *Information Fusion* 6 (2005) 143-151
13. <http://www.ics.uci.edu/~mllearn/databases/>.
14. Gluck, M., Corter, J.: Information, Uncertainty, and the Utility of Categories. *Proc. Seventh Ann. Conf. Cognitive Soc.* (1985) 283-287
15. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2 (1955) 83-97

Mining Maximal Local Conserved Gene Clusters from Microarray Data*

Yuhai Zhao, Guoren Wang, Ying Yin, and Guangyu Xu

Institute of Computer System, Northeastern University
Shenyang 110004, China
wanggr@mail.neu.edu.cn

Abstract. In this paper, we explore a novel type of gene cluster called local conserved gene cluster or LC-Cluster for short. A gene's expression level is local conserved if it is expressed with the similar abundance only on a subset of conditions instead of on all the conditions. A subset of genes which are simultaneously local conserved across the same subset of samples form an LC-Cluster, where the samples correspond to some phenotype and the genes suggest all candidates related to the phenotype. Two efficient algorithms, namely FALCONER and E-FALCONER, are proposed to mine the complete set of maximal LC-Clusters. The test results from both real and synthetic datasets confirm the effectiveness and efficiency of our approaches.

1 Introduction

It is an important research problem in bioinformatics and clinical research to explore the patterns in microarray datasets [1, 2, 3]. While most of recent researches concentrate on the mining of co-regulated genes [3, 4, 5], they neglect another biologically significant analysis scheme. That is, identify “patterns” of gene expression that can be used to predict cell phenotype. This is also known as the cell phenotype prediction problem [6, 7].

When a gene's expression level is measured across a variety of samples, the expression values usually span a wide range. Biologically, the samples on which the gene is expressed similarly may correspond to a phenotype (e.g. a cancer tissue) while the gene may suggest the candidate gene correlated to the phenotype [6, 7]. However, if the gene keeps invariant or changes very little on all or a large majority of the samples, it becomes meaningless [8, 9].

Motivated by this problem, given a gene g_i , we say it to be *local conserved* if it is expressed similarly only across an appropriate subset of samples, e.g. S, but not across all or a large majority of the samples. Shortly, we say g_i is local conserved across S. Explicitly, it provides valuable hypothesis for biologists to identify a subset of genes G and a subset of samples S such that each gene $g_i \in G$ is local conserved across the samples in S since the samples may correspond to a phenotype and the genes may be all candidates related to the phenotype [6, 7].

* Supported by National Natural Science Foundation of China under grant 60573089 and 60473074.

We call such a subset of genes and samples a *local conserved gene cluster* or an *LC-Cluster* for short.

Trivially, given an LC-Cluster, D , any subset $D' \subseteq D$ is also an LC-Cluster. To avoid such redundancy, in this paper, we tackle the problem of *mining all maximal local conserved patterns* (see subsection 2.2) *from gene-sample microarray datasets* and make the following contributions: (1) we propose a model of local conserved gene cluster, i.e. LC-Cluster in gene-sample microarray datasets. We justify that the model is meaningful for biomedical research; (2) the clusters can be arbitrarily positioned anywhere in the input data matrix and they can have arbitrary overlapping regions since a gene may participate in several biological pathways; (3) we identify the computational challenges and conduct a systematic research on mining LC-Clusters via proposing two algorithms, namely FALCONER¹ and its enhanced version, E-FALCONER; and (4) we conduct an extensive empirical evaluation on both real datasets and synthetic datasets. Experimental results show that our proposed methods can find local conserved gene clusters interesting to biomedical research from real datasets and our algorithms outperform the existing enumeration tree-based method in performance.

The remainder of this paper is organized as follows. Section 2 gives a formal definition of the LC-Cluster model. Section 3 discusses our algorithms in detail. Experimental results and analysis are shown in Section 4. Finally, Section 5 concludes this paper.

2 The LC-Cluster Model

In this section, we define the LC-Cluster model for mining local conserved genes that are expressed similarly across a user-specified subset of samples.

2.1 Preliminary Concepts

Let $G = \{g_1, g_2, \dots, g_s\}$ be a set of s genes, and $S = \{s_1, s_2, \dots, s_t\}$ be a set of t samples. A microarray dataset is a real-valued $s \times t$ matrix $D = G \times S = \{d_{ij}\}$, with $i \in [1, s]$, $j \in [1, t]$. Each entry records the expression level of gene g_i on sample s_j . Table 1 shows an example of the dataset that we will look at in this paper.

Table 1. Running Dataset

gene	s_1	s_2	s_3	s_4	s_5
g_1	87.6	110	88.2	93	140.2
g_2	20	20.5	21.7	31.2	-10
g_3	50.2	56.7	65	-20	156.3
g_4	-18.6	-17.9	-20	-17.3	76.7
g_5	12.3	14	20	-50	126.5
g_6	-10	37.5	40.1	-10	86.9
g_7	31.2	32	31.2	33	31.6
g_8	72	72.6	71.2	67.5	67.9

We have introduced what is the local conserved gene briefly in section 1. For more rigorously, we formulate it in definition 1.

¹ FALCONER stands for Find All maximaL local conserved gene clustERS.

Definition 1. Given a complete microarray expression matrix $D=G \times S$, where $G=\{g_1, g_2, \dots, g_s\}$ and $S=\{s_1, s_2, \dots, s_t\}$, and three user-specific parameters α, β ($0 < \alpha \leq \beta < 1$) and δ , we say a gene $g_i \in G$ is **local conserved** across a subset of samples $S'=\{s_{j_1}, s_{j_2}, \dots, s_{j_m}\}$ ($S' \subseteq S$) iff g_i satisfies the following conditions: (1) $\forall x, y \in [1, m], |d_{ij_x} - d_{ij_y}| \leq \delta \times |MIN(d_{ij_x}, d_{ij_y})|$, where d_{ij_x} and d_{ij_y} are the expression levels of gene g_i under samples s_{j_x} and s_{j_y} respectively, and δ denotes the maximal tolerant fraction of difference between the two expression values, (2) $\alpha \times |S'| \leq |S'| \leq \beta \times |S|$, where $|S'|$ and $|S|$ denote the number of samples in S' and S respectively, and (3) S' cannot be extended to another sample set S'' with more than $\beta \times |S|$ samples such that condition (1) still holds across S'' .

2.2 Model Definition and Problem Statement

After giving the formal definition of local conserved gene, naturally we propose the definition of an LC-Cluster as follows:

Definition 2. Given a complete microarray expression matrix $D=G \times S$ and user-specified parameters α, β and δ , as defined in definition 1, an **LC-Cluster** is a submatrix $D'=G' \times S'$ such that $\forall g_i \in G', g_i$ is local conserved across S' .

Given this definition, a gene expression data matrix may contain many LC-Clusters. Among all LC-Clusters, we are interested in the maximal ones. An LC-Cluster with X genes and Y samples is called **maximal** iff (1) $X \geq min_g$, $min_s \leq Y \leq max_s$, where $min_s = \alpha \times |S|$ is the minimal number of samples in an LC-Cluster, $max_s = \beta \times |S|$ is the maximal number of samples in an LC-Cluster and min_g is the minimal number of genes in an LC-Cluster, (2) there doesn't exist another LC-Cluster with X' genes and Y' samples such that $X \subseteq X'$ and $Y \subseteq Y'$.

Problem Statement. Given: (1) a complete expression matrix $D = G \times S$, (2) α , a minimal fraction of samples, (3) β , a maximal fraction of samples, (4) min_g , a minimal number of genes, and (5) δ , a maximal tolerant fraction of difference between two expression values, the task is to find all maximal submatrices $D' = G' \times S'$ which satisfy definition 2, and $|G'| \geq min_g, \alpha \times |S'| \leq |S'| \leq \beta \times |S|$.

3 The LC-Cluster Algorithms

In this section, we present our algorithms, i.e. FALCONER and its enhanced version, E-FALCONER, to mine all maximal LC-Clusters from a given microarray dataset. For clarity, we take the dataset in Table 1 as a running example, where $\alpha = 0.4, \beta = 0.8, \delta = 0.15$ and $min_g = 3$.

3.1 Maximal Conserved Sample Sets

In both algorithms we proposed, to compute LC-Clusters, we need to check whether a subset of genes are local conserved across a subset of samples. To facilitate the tests, for each gene g_i , we compute the sets of samples S such that

(1) g_i is local conserved on S , and (2) there exists no superset $S' \supset S$ such that g_i is also local conserved on S' . S is called a **maximal conserved sample set** of g_i or **MCSS** for short.

After sorting all samples in ascending order based on their expression values on a gene g_i , we can find all *MCSSs* for g_i in a sliding window manner [4]. Differently, we can slide the left end of the window several positions at a time and locate the right end of the window in a binary search manner. For each gene in Table 1, *MCSSs* obtained by the above process are listed in Figure 1(a) Note: a gene may have no(such as g_7 and g_8) or more than one(such as g_3 and g_6) *MCSSs*, and some samples will not exist in any *MCSS*, e.g. s_5 .

Gene	Maximal conserved sample sets
g_1	$\{s_1, s_3, s_4\}$
g_2	$\{s_1, s_2, s_3, s_4\}$
g_3	$\{s_1, s_2\}, \{s_2, s_3\}$
g_4	$\{s_1, s_2, s_3, s_4\}$
g_5	$\{s_1, s_2\}$
g_6	$\{s_1, s_4\}, \{s_2, s_3\}$

(a) The maximal conserved sample sets for genes

Sample	The inverted index
s_1	$\{g_1.b_1, g_2.b_1, g_3.b_1, g_4.b_1, g_5.b_1, g_6.b_1\}$
s_2	$\{g_2.b_1, g_3.b_1, g_3.b_2, g_4.b_1, g_5.b_1, g_6.b_2\}$
s_3	$\{g_1.b_1, g_2.b_1, g_3.b_2, g_4.b_1, g_6.b_2\}$
s_4	$\{g_1.b_1, g_2.b_1, g_4.b_1, g_6.b_1\}$

(b) The inverted indices for samples

Fig. 1. The maximal conserved sample sets and the inverted indices

3.2 The Mining Algorithms

We label each *MCSS* in Figure 1(a) by the gene g_i , and the set-id, b_j , in the gene. For example, gene g_3 has two *MCSSs*, $g_3.b_1 = \{s_1, s_2\}$ and $g_3.b_2 = \{s_2, s_3\}$. For each sample s , we make up its *inverted index* as the list of all *MCSSs* containing s , as shown in Figure 1(b). Consequently, when we want to find all genes that are local conserved across a given sample set S , say $\{s_1, s_3, s_4\}$, we only need to get the intersection of the inverted indices of the samples s_1, s_3 and s_4 , i.e. $\{g_1.b_1, g_2.b_1, g_4.b_1\}$, instead of a complete scanning of the list in Figure 1(a)

The FALCONER Algorithm. Some enumeration tree-based methods, such as [7], can be referenced to discover all maximal LC-Clusters. However, previous methods must perform ‘ \subseteq ’ operation many a time to decide whether a cluster is maximal when it is found at a certain node of the enumeration tree. The complete set enumeration tree for samples in Figure 1 is shown in Figure 2.

It is well known that the operation ‘ \subseteq ’ is very time-consuming. In this subsection, we design a new algorithm, FALCONER, which finds all maximal LC-Clusters without the set operation ‘ \subseteq ’ and prunes the unpromising sample combinations substantially.

In the set enumeration tree, each node contains a unique subset of samples, which we refer to as *NOW*. At any time during the execution of the resursive depth-first algorithm, *NOW* is extended by a sample on branching, or reduced by a sample on backtracking. At each node, we also maintain the other two sample sets, i.e. *CA* and *EX*. *CA* is a subset of samples where any sample, say s_i , will

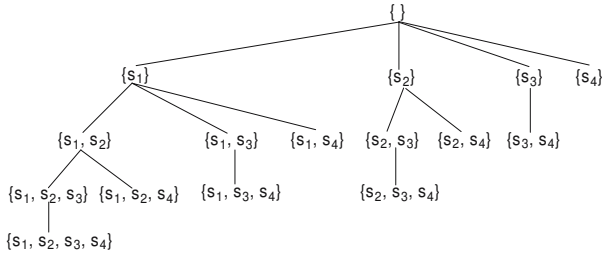


Fig. 2. Set enumeration tree of samples in Figure 1

be used to extend NOW to a larger subset of samples $S' = NOW \cup \{s_i\}$ such that at least min_g genes are local conserved across S' . EX contains all samples that were previously used to extend NOW and are now explicitly excluded from the extension.

Given the three sample sets, clearly, a necessary condition to form a maximal LC-Cluster on the sample set NOW is $CA = \emptyset$. However, this is not sufficient. Suppose $CA = \emptyset$ and $\exists s_i \in EX$, the LC-Cluster on NOW cannot be maximal if $G_{NOW} = G_{NOW \cup \{s_i\}}$, since $G_{NOW \cup \{s_i\}} \times NOW$ is also an LC-Cluster, which must have been detected before according to the definition of EX . If and only if $\forall s_i \in EX, G_{NOW} \neq G_{NOW \cup \{s_i\}}$, we can report $G_{NOW} \times NOW$ as an unsubsumed maximal LC-Cluster. Similarly, the *second necessary and sufficient condition* for reporting an unsubsumed maximal LC-Cluster at a node u is both $CA = \emptyset$ and $EX = \emptyset$. In conclusion, based on the states of three sets instead of operation ' \subseteq ', we can immediately know whether there is a new maximal LC-Cluster formed at a node u . Additionally, we adopts bit vector for the above computation, which further speeds up the decision of maximal LC-Cluster.

Next, we look at some pruning rules which are extremely important for the efficiency of FALCONER algorithm and give the formal algorithm with these pruning rules in Figure 3.

Pruning Rule 1. *Given a sample s_i and the sample sets NOW , CA and EX at a node in the enumeration tree, if $G_{NOW \cup \{s_i\}}$ contains less than min_g genes, then $s_i \notin CA$. Furthermore, $s_i \notin EX$.*

Pruning Rule 2. *For a node u in the enumeration tree, if its NOW and CA satisfy $|NOW| + |CA| < min_s$, then NOW cannot lead to any LC-Cluster with min_s or more samples, and thus the subtree of u can be pruned.*

Pruning Rule 3. *For a node u in the enumeration tree, whose $NOW = \{s_{i_1}, s_{i_2}, \dots, s_{i_k}\}$, if there exists a sample $s_j (j \notin \{i_1, i_2, \dots, i_k\})$ in EX of u , which always occurs in the MCSSs as a concomitant of $\{s_{i_1}, s_{i_2}, \dots, s_{i_k}\}$ (that is, every MCSS containing $\{s_{i_1}, s_{i_2}, \dots, s_{i_k}\}$ also contains $\{s_j\}$), then the recursive search rooted at node u cannot lead to any new maximal LC-Cluster, and thus can be pruned.*

The E-FALCONER Algorithm. Inspired by the pruning rule 3 above, we further present an enhanced version of the FALCONER algorithm, i.e.

Algorithm FALCONER**Input:** α, β, δ , set of genes G and samples S **Output:** maximal LC-Clusters set R **Method:**

```

1: Generate MCSSs and their inverted indices;
2: Initialization: Now= $\emptyset$ , CA= $S$ , EX= $\emptyset$ ;
3: Call MineLC(Now, CA, EX);

Subroutine: MineLC(Now, CA, EX)
Method:
1: if CA =  $\emptyset$  and EX =  $\emptyset$  then
2:   if  $|Now| \geq min_c$  then
3:     derive the intersection of inverted indices  $G_{Now}$  for samples in Now;
4:      $R \leftarrow G_{Now} \times Now$ ;
5:     return;
6: if CA =  $\emptyset$  and EX  $\neq \emptyset$  then
7:   if  $|Now| \geq min_s$  and  $\forall s_i \in EX, |G_{Now}| \neq |G_{Now \cup \{s_i\}}|$  then
8:     derive the intersection of inverted indices  $G_{Now}$  for samples in Now;
9:      $R \leftarrow G_{Now} \times Now$ ;
10:    return;
11: take the first candidate, c, in CA;
12:  $Now = Now \cup \{c\}$ ;
13:  $CA = CA - \{c\}$ ;
    //Apply Pruning 1 :
14: create CA' and EX' by removing irrelevant samples from CA and EX;
    //Apply Pruning 2 :
15: if  $|Now| + |CA'| < min_s$  then
16:   return;
17: if Now can be pruned by pruning rule 3 then
18:   return;
19: call MineLC(Now, CA', EX');
20:  $Now = Now - \{c\}$ ;
21:  $EX = EX \cup \{c\}$ ;
22: go to 1

```

Fig. 3. The FALCONER algorithm

E-FALCONER. Different from FALCONER's always selecting the first candidate from CA, the *basic candidate selection principle* of E-FALCONER is to choose a new candidate such that as more as possible unfruitful branches can be cut off at the earlier stage. Before describing it, We first introduce a basic concept that is useful for further discussion.

Definition 3. Given an expression matrix and its sample set $S = \{s_{i_1}, s_{i_2}, \dots, s_{i_k}\}$, for any sample $s_{i_j} \in S$, there exists a set $X = \{s_{i_x} | s_{i_x}$ in a MCSS infers s_{i_j} in the same MCSS and $x \neq j$ and $x \in [1, k]\}$ (every MCSS containing s_{i_x} must contain s_{i_j}). X is called the **leader set** of s_{i_j} and every sample in it is called a **leader** of s_{i_j} . The total number of samples in X is called the **leader count** of s_{i_j} in S .

At a node u , E-FALCONER first decides whether there exists a sample s_j in $CA \cup EX$ such that every MCSS containing all samples in Now also contains s_j . If s_j in EX , the recursive search backtracks immediately for the pruning rule 3. If s_j in CA , we select s_j as the first candidate such that it can emerge in EX as earlier as possible. If no such a s_j , we search s_k from $CA \cup EX$ with the largest Lcount in CA . If s_k in CA , we take it as the first candidate. On backtracking, s_k is moved into EX . At this node s_k in EX , we select a sample from CA which is not the leader of s_k as the first candidate. If all samples in CA which are not leaders of s_k have been moved to EX , the remaining searches from node u can be cut off for pruning rule 3 and definition 3. The E-FALCONER algorithm can be obtained by replacing step 11 with what we said above. Limited by space, we don't list it here.

4 Experiments

All experiments are done on a 2.0-GHz Dell PC with 512M memory running Window 2000 and the algorithms are coded in C++. Both synthetic and real microarray datasets are used to evaluate our algorithms. For the real dataset, we use AML-ALL dataset [8]. The synthetic datasets can be obtained by a data generator algorithm [7].

4.1 Scalability

We test the scalability of both FALCONER and E-FALCONER on synthetic datasets with $\alpha=0.4$, $\beta=0.8$, $\delta=0.1$ and $min_g=0.01*\#gene$. We first fix the number of samples to 30, and report the runtime w.r.t #genes (Figure 4(a)). We can see both approaches show an approximately linear scalability w.r.t #gene. Figure 4(b) shows the scalability for both approaches under different sizes of sample sets, when the number of genes is fixed to 3000. We can see both approaches scale well w.r.t the number of samples.

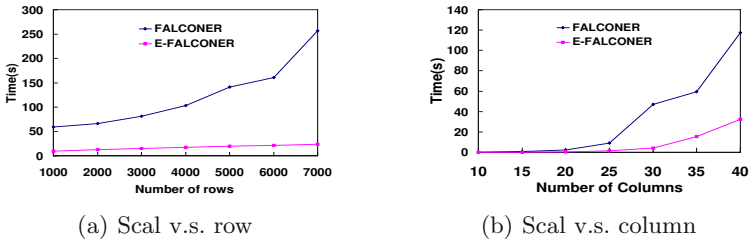
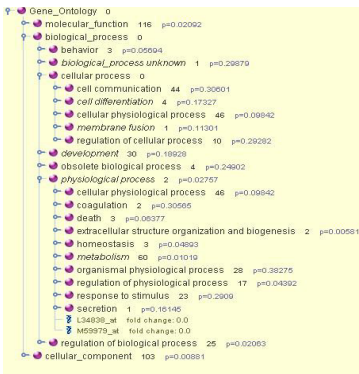
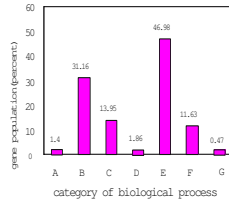


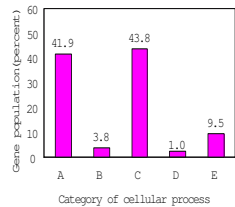
Fig. 4. Run time v.s. #genes and #samples



A-Behavior B-Cellular process C-Development
D-Obsolete biological process E-Physiological process
F-Regulation of biological process
G-Biological_process unknown



A-Cellular communication B-Cell differentiation
C-Cellular physiological process
D-Membrane fusion
E-Regulation of cellular process



(a) The gene ontology tree for genes in cluster C_3

(b) The distribution of biological process.

(c) The distribution of cellular process

Fig. 5. The gene ontology tree and the distribution of function for genes in cluster 3

4.2 Biological Significance Analysis

With the given parameters: $\alpha=0.2$, $\beta=0.6$, $\delta=0.05$ and $min_g=80$, some interesting observations from AML-ALL are found when we feed them to Onto-Express [7]. Figure 5(a) shows a feedback ontology tree for a discovered LC-Cluster. Figure 5(b) and 5(c) are the further analysis of genes in the cluster.

5 Conclusion

In this paper, we investigate a novel type of gene expression pattern, i.e. LC-Cluster, which is substantially different from current co-regulation pattern. We also develop two effective methods, FALCONER and E-FALCONER, to mine all such maximal patterns. The experimental results show that our approaches can effectively and efficiently find biologically significant LC-Clusters. Further, since E-FALCONER selects the next candidate sample that can make the unfruitful branches cut off as more as possible, it constantly outperforms FALCONER.

References

1. Hughes, T.R., Marton, M.J., Jones, A.R., et al: Function discovery via a compendium of expression profiles. *Cell* **102** (2000) 109–126
2. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. In: Proc. of the ISMB 2002 Conf., Canada. (2002) 136–144
3. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. of ISMB 2000 Conference. (2000) 93–103
4. Wang, H., Wang, W., Yang, J., Yu, P.S.: Clustering by pattern similarity in large data sets. In: Proc. of the 2002 ACM SIGMOD Conference, Wisconsin. (2002) 394–405
5. Liu, J., Wang, W.: Op-cluster: Clustering by tendency in high dimensional space. In: Proc. of ICDM 2003 Conference. (2003) 187–194
6. Murali, T., Kasif, S.: Extracting conserved gene expression motifs from gene expression data. In: In Pacific Symposium on Biocomputing. (2003) 77–88
7. Jiang, D., Pei, J., Ramanathan, M., Tang, C., Zhang, A.: Mining coherent gene clusters from gene-sample-time microarray data. In: Proc. 10th ACM SIGKDD Conference. (2004) 430–439
8. Golub, T.R., Slonim, D.K., Tamayo, P., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286** (1999) 531–537
9. Tang, C., Zhang, L., Zhang, A., Ramanathan, M.: Interrelated two-way clustering: An unsupervised approach for gene expression data analysis. In: Proc. Second IEEE Intl Symp. Bioinformatics and Bioeng. (2001) 41–48

A Novel P2P Information Clustering and Retrieval Mechanism

Huaxiang Zhang¹ and Peide Liu²

¹ Dept. of Computer Science, Shandong Normal University, Jinan 250014 China
huaxzhang@hotmail.com

² Dept. of Information Management, Shandong Economic University, Jinan 250014 China

Abstract. Information retrieval over peer-to-peer networks is an important task. In order to avoid query message flooding and improve information retrieval performance, clustering the nodes sharing the same kind of interests is a feasible approach. An interest crawling agent utilizing an incremental learning algorithm is proposed to calculate a crawled node's score, which is used for establishing a node cluster. An active time window is employed to accelerate the query. In order to utilize the node cluster efficiently, we present a \mathcal{E} -greedy query routing strategy. Experimental results show our approach performs well.

1 Introduction

The main challenge faced by information retrieval (IR) in peer-to-peer (p2p) networks is how to route the query messages and select the sources containing relevant answers. Message routing strategies used in completely decentralized p2p networks adopt flooding techniques to broadcast query messages to each node. Nodes waste the computational resources to handle irrelevant query messages that increase the network traffic. In very structured p2p overlay topology networks, shared documents are mapped to different specialized nodes by data hashed table (DHT) algorithms. This kind of file sharing mechanisms lacks scalability.

In order to improve the IR performance, a node should selectively route a query message to a set of relevant nodes rather than simply adopting the flooding technique. Kalogeraki et al [1] adopt the intelligent search mechanism (ISM) to help the querying node in finding the most relevant answers to the query efficiently. One problem of ISM is that the routing message may get locked into a cycle and fail to explore the rest of the networks. Triantafillou et al [2] propose the concept of node autonomy and shared document group based on document semantics. Their work ignores the dynamics of p2p as predefining the node categories. Each node may join in or drop out of the p2p networks randomly, and the document popularity just represents the time a document being visited. J. Lu et al [3] study the application of content-based resource selection and document retrieval in hybrid p2p networks, and the K-L divergence [4] between the query and the collection of documents stored in the leaf node is calculated in their approach. When a leaf node receives a query message, it uses the K-L divergence retrieval algorithm [4] to rank the documents in its collection and generates a query

hit message. Document relativity of the leaf nodes is ignored in calculating K-L divergence, and the query message is flooding among the directories also.

To minimize the number of query messages in p2p networks, recent studies [5] show that local awareness in p2p networks may improve the search performance. Some work has been done in this direction. Eisenhardt et al [6] present an approach to distributed clustering of documents. They use the time-consuming k-means clustering in p2p IR networks. Fessant et al [7] test the performance of interest-based clustering for video and audio files respectively, but the interest-based clustering lacks semantics consideration of the answers. Voulgaris et al [8] create a semantic overlay, and the linking nodes being “semantically close” are interested in similar documents. Each node, based on the query/responses history, creates a list of semantic neighbors to which queries are forwarded first. As they point out the list can be contaminated by some kinds of nodes. Lu et al [9] utilize the retrieval history information of a node’s neighbors to accelerate the IR. It can get high retrieval performance when answers are on the nearby nodes, but low performance when answers are on far away nodes.

Based on the above analysis, we propose a novel content-based node clustering and IR method. A crawling agent is proposed to cluster nodes, and the agent adopts an incremental learning algorithm to crawl nodes in p2p networks. The crawling agent calculates the score of each crawled node and cluster the nodes based on the scores. We also propose a concept of active time window to accelerate IR.

2 Managing Index Information

Index information is at the heart of p2p search methods, and it can be local, centralized or distributed [10]. We use a hybrid structured overlay topology in Psimulator. There are two kinds of nodes in the test platform Psimulator, one kind is for storing the index information and is called the indexing node (IN), another kind is for storing the shared documents of p2p networks, and is called the data node (DN). All INs are connected by a structured protocol such as CAN [11], and all DNs are connected by an unstructured protocol. Each IN duplicates its neighbors’ index information to lower the risk of single point failure. Unstructured protocol connection between DNs keeps the scalability and dynamics of p2p networks. Shared files are stored in different DNs and are easy to be managed.

The shared documents in one DN may belong to different interest groups, and the query message uploaded by a DN may not be the same interest as that stored in it. Most previous research just assumes the query message belongs to a changeless interest group. In fact, a node may upload any kind of query information.

According to the scientific taxonomy, we first divide the shared information into different large categories, and mapping the meta-data <node ip, category information, keywords of category> to an IN by DHT [11]. One DN may map different meta-data to several INs as it may have documents of different interests. When a node joins in the p2p networks, it transmits its different meta-data to different INs. If a node drops out of the networks, its meta-data is marked as stop-service, and when it joins the networks again, its meta-data is re-activated. Category information identifies the interest of a DN, and keywords of each category are provided by DNs.

3 Clustering a Node

We use a crawling agent to cluster the DNs that are “semantically close” in a node cluster. When a DN is idle, it sends a query message to a correlative IN. The IN searches the meta-data table to find the relevant DNs and sends these DNs’ meta-data to the message owner node. The node first checks whether these DNs are in its node cluster, if not, the node sends different crawling agents to those DNs still not jointing in the node cluster, and each crawling agent belongs to one interest group. The crawling agent calculates the similarities between the collection of the interest documents of its owner node and the documents stored in the node it crawled, and scores the crawled node. When the score is greater than a threshold, the crawled node is clustered in the owner node’s cluster.

3.1 Managing the Node Clustering

Each node cluster collects the online and offline information of its members by crawling agents, and divides the node members into several groups based on a time window. A time window is a series of time periods. The online time of a member node only lies in one period of the series. At each time, only one time window is active. We call it an active time window.

The node registers its cluster index information in the correlative IN. We call it the master node of the cluster. An online node with the highest score is noted as the slaver node of the cluster. The clustering information is duplicated in the slaver node. When the master node is down, the slaver node takes the role of the master node, and selects another node as its slaver node.

3.2 Similarity Ranking Based on Incremental Learning

When a node joins in p2p networks for the first time, its node cluster has not been built, and an algorithm should be employed by the node to establish the cluster. We use the topic-driven web resource discovery mechanism to solve this problem. A web crawler is used to crawl and collect relevant web pages [12]. A web crawler, commonly using the breadth first search, is commonly used in search engines to collect as many web pages as possible in a certain time cycle. This approach can be used to cluster the nodes. We call the crawling agent in the environment an interest crawling agent (ICA). The task of an ICA is not to download the shared documents in other nodes, but to calculate the similarities between documents. These similarities are used for establishing the node clusters.

The vector space model (VSM) [13] is commonly used in text categorization and clustering. In VSM, each document d_i is represented as a vector. We adopt *tf / idf* method to calculate the document term weight vector, and the cosine similarity between documents i and j (noted as $s(i, j)$, $s(i, j) \in [0,1]$). Nodes are connected with each other, and the connected nodes are called neighbors. If two nodes are connected directly, they are called immediate neighbors. Self-organization is one important feature of p2p networks, and each node can change to be an immediate neighbor of any other nodes. The score of a crawled node is calculated as

$$R_i(j) = (1 - \gamma)s(i, j) + \gamma \sum_{k=1}^N s(j, k) / N \tag{1}$$

$R_i(j)$ denotes the score of node j related to ICA owner's i th interest and N is the number of j 's immediate neighbors with similarity $s(j, k)$ greater than a threshold. $\gamma \in [0, 1]$ is a discounting rate used to adjust the weights of the two parts in the right of (1).

A node's score changes as an ICA crawls it. At first, the node cluster has not been established, and the immediate neighbors have not contributed to the score. As the number of nodes in a cluster increases, a node's immediate neighbors' contribution increases. We call the learning algorithm an incremental learning.

All the scores are collected and sorted in a descending order during the ICA's crawling processes. The ICA also has the ability to estimate the crawled nodes' online and offline time, and this information is used by the owner node to group its node cluster members.

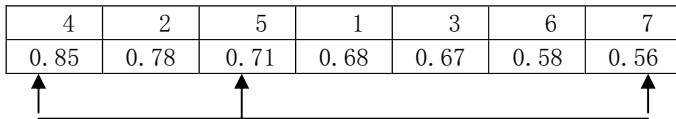


Fig. 1. Scores grouped with online time

Figure 1 shows a descending score list of a node cluster. The three arrow-pointed terms are inside of the active time window. If an IR message is routed to a node, the node first calculates the similarity between the query and the documents stored locally, and then broadcasts the query message to nodes inside the active window. No message is broadcast to the nodes outside of the active window, so the number of query messages is decreased greatly.

4 IR Mechanism

The resource location mechanism in Psimulator is as follows. When a query message is generated, the node first searches its local documents. If answers can be found locally, it can be sure that answers are in the node cluster. If this is not true, the query message is forwarded to the relevant IN through the structured layer. The IN searches the cluster information provided by different masters of clusters to find the proper cluster and forward the query message. If the query message is forwarded 3 times along a path in a node cluster without any answer returned, we stop the message forwarding along this path. The reason of limiting the forwarding time to 3 is just to minimize the number of the query messages in the networks.

Whether a query vector is inside a node cluster or not is determined by the distance between the query vector and the cluster center. If a vector lies in a cluster, it does not mean that other clusters don't have relevant answers. As shown in figure 2, the query

vector represented by the black triangle lies in the black dot cluster, but near the edge of the cluster. Some of the answers inside the dotted line circle belong to another cluster rather than the black dot cluster. We use \mathcal{E} -greedy to harness the query precision. For example, we set $\mathcal{E}=0.1$ and use a random number to determine whether to forward the query message to another cluster with the second greatest similarity as well as the most proper cluster.

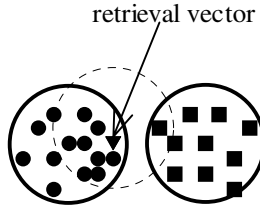


Fig. 2. The relation between a query vector and two node clusters

5 Evaluation Methodology

Both the retrieval accuracy and the efficiency of the query message routing are important in p2p networks. The retrieval accuracy is measured with three terms, and we use formula (2), (3), (4) to calculate them. The efficiency of the message routing is measured by the number of query messages forwarded in p2p network for a query, and is calculated by formula (5).

R_c (recall), P_c (precision), F-measure and $\overline{N_m}$ (averaged message number) are calculated as follows

$$R_c = \frac{Nr_{[0,T]}}{No_{[0,T]}} \quad (2); \quad P_c = \frac{Nr_{[0,T]}}{N} \quad (3); \quad \text{F-measure} = \frac{2R_c P_c}{R_c + P_c} \quad (4); \quad \overline{N_m} = \frac{M_{[0,T]}}{N_e} \quad (5)$$

Where, $Nr_{[0,T]}$ is the number of answers returned during time interval $[0, T]$, and $M_{[0,T]}$ the number query messages during time interval $[0, T]$. $No_{[0,T]}$ is the number of correct answers online during time interval $[0, T]$, N the returned answer number, and N_e the experimental times (we set 20 to it).

We obtain the results in three different cases: query in local node cluster, query among cluster and query in Gnutella. When the local documents and the query information are highly similar, the retrieval is always executed in the local node cluster, and we call it RIC (Retrieval In Cluster). When local documents are not similar to the query information, the query message is routed to a “remote” node cluster, and we call it RAC (Retrieval Among Cluster). We compare the above two cases with query in Gnutella called RIG (Retrieval In Gnutella). A same query is executed in RIC, RAC and RIG with execution time T .

6 Experimental Results

The Psimulator with 600 nodes is used as the test platform. No less than two interest file groups are stored in one node, and all files are divided into 96 interest groups. The number of files of one interest group changes from 31 to 294, and the total file number is 14560. Each file is indexed by 100 terms.

Table 1. Basic experimental data

Node number	600
IN number	96
Interest group number	96
Term number	100
Total file number	14560
Minimal file number in a group	31
Maximal file number in a group	294
Average file number	151
δ	0.15
The threshold	0.45

Table 2. Number of answers returned in a given time period

Number of answer	RIC time(ms)	RAC time(ms)	RIG time(ms)
1	42	45	48
2	43	50	61
4	46	54	70
6	51	59	84
8	54	61	92
10	56	67	104
12	61	68	120
14	65	72	150
16	68	75	190

We repeat the experiments 20 times and averaged the results. Table 2 shows that RIC takes the shortest time than both RAC and RIG do for returning the same number of answers, and RIG takes the longest time among the three cases. It can be explained that the retrieval is executed in the local node cluster in RIC, and query messages are routed among members of the node cluster. The time taken in RAC is close to that taken in RIC, and it is that the query message is first routed to an IN in RAC, after the IN returns the meta-data of the destination node cluster to the query uploading node, the query message is rerouted to the destination node cluster. Then the retrieval becomes a RIC. It takes little time to find the destination node cluster. The time taken in RIG is quite different from that taken in RIC and RAC. RIG uses the flooding technique to route the query messages to other nodes in p2p networks, the retrieval efficiency is very low, and longer time is taken for more answers.

Table 3. Measure terms in 3 cases ($\gamma=0.15$)

	R_c	P_c	F-measure
RIC	32.59%	78.28%	0.4602
RAC	29.62	77.95%	0.4292
RIG	5.37%	78.9%	0.1005

Table 4. Average message volume

	Average number of query message
RIC	43
RAC	45
RIG	188

Table 3 gives the values of the three measure terms averaged on 20 experiments. It's clear the three precisions are very close for the same similarity calculation in the three cases. But the recall in RIG is much smaller than that in RIC and RAC, and RIC gets the best recall. Nodes are clustered in RIC and RAC, and the query message hits the answers with a high probability. But in RIG, the hit rate is low. The value of F-measure shows this clearly. Table 4 gives the value of message volume in p2p networks, and it's clear the number of messages in RIG is the largest one.

7 Conclusions

This paper discusses the node clustering and information retrieval mechanisms in hybrid p2p networks. Most of the research on IR focuses on improving the robustness and efficiency of distributed information storage, and retrieval is limited to matches between query terms and document names or identifiers. These techniques are not sufficient for retrieval based on document content or semantics. If we adopt the successful techniques widely used in text categorization and clustering to IR in p2p, the match techniques can be extended to the content-based retrieval, and the retrieval accuracy may be increased. If nodes storing similar documents are clustered, the number of query messages can be greatly reduced and the retrieval efficiency can be increased to a high level. Applying the ideas to IR in p2p network, we use the vector space model to index shared documents, and adopt hybrid p2p architectures. INs are connected with structured protocols and manage the index information of documents belonging to one interest group. An ICA uses an incremental learning algorithm to calculate the score of a crawled node. ICA's owner node ranks the collected node in descendent order based on the scores, and groups the nodes according to the online times. When a retrieval request message is generated, the node first searches its local documents to determine whether the answers are inside the local node cluster. If it is true, the node forwards the query messages to all nodes inside the active time window.

References

1. V. Kalogeraki, D. Gunopulos and D. Zeinalipour-Yazti. A Local Search Mechanism for Node-to-Node Networks . Proc. of CIKM'02, McLean VA, USA, 2002
2. P. Triantallou, C. Xiruhaki, M. Koubarakis, N. Ntarmos. Towards high performance node-to-node content and resource sharing systems. In proceedings of the int. conf. on innovative data systems research(CDIR), 2003
3. J. Lu, J. Callan. Content-based retrieval in hybrid peer-to-peer networks. CIKM'03, Nov. 2003
4. P. Ogilvie, J. Callan. Experiments using the lemure toolkit. In proc. Of the 10th text retrieval conference(TREC-10), 2001
5. K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in node-to-node systems. In INFOCOM'03.
6. M. Eisenhardt, W. Müller, A. Henrich. Classifying documents by distributed p2p clustering. GI Jahrestagung (2) 2003: 286-291
7. L. Fessant, S. Handurukande, A. M. Kermarrec, L. Massoulié. Clustering in Peer-to-Peer File Sharing Workloads, 2004
8. S. Voulgaris, A. Kermarrec, L. Massoulié, M. V. Steen. Exploiting semantic proximity in node-to-node content searching, 2004
9. Z. Lu, K. S. McKinley. The effect of collection organization and query locality on information retrieval system performance and design. Book chapter in advances in information retrieval, Kluwer, New York, 2000. Bruce croft, editor
10. J. Risson, T. Moors. Survey of research towards robust peer-to-peer networks: search methods. Technical Report UNSW-EE-P2P-1-1, Univ. of New South Wales, Sydney, Australia. 2004
11. Can project home page. <http://www.icir.org/sylvia/>
12. F. Menczer, G. Pant, P. Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. ACM Transactions on Internet Technology, Forthcoming, online at <http://www.informatics.indiana.edu/fil/Papers/TOIT.pdf>, 2003
13. G. Salton. Automatic Information Organization and Retrieval. 1968, New York: McGraw-Hill

Keeping Track of Customer Life Cycle to Build Customer Relationship

Sung Ho Ha^{1,*} and Sung Min Bae²

¹ School of Business Administration, Kyungpook National University, 1370 Sangyeok-dong, Buk-gu, Daegu, Korea
Tel.: +82-53-950-5440; Fax: +82-53-950-6247
hsh@mail.knu.ac.kr

² Department of Industrial & Management Engineering, Hanbat National University, San 16-1, Duckmyoung-dong, Yusong-gu, Daejeon, Korea
loveiris@hanbat.ac.kr

Abstract. Using the CRM perspective to investigate customer behavior, this study differentiates between customers through customer segmentation, tracks customers' shifts from segment to segment over time, discovers customer segment knowledge to build an individual transition path and a dominant transition path, and then predicts customer segments' behavior patterns. Using real world data, this study evaluates the accuracy of derived customer knowledge. Concluding remarks discuss future research that can extend the work this study presents.

1 Introduction

A successful company today does well in both keeping and managing its customers through providing a bundle of attractive, personalized services that satisfy its customers' needs. This is due to the premise that it is less expensive to cross-sell an incremental product or service to existing customers, and that attracting new customers is expensive [10].

Therefore, a company needs to understand its existing customers and their needs more completely than ever before. Satisfying its customers' needs and building strong relationships with customers entail good customer relationship management (CRM). The goal of CRM is to forge closer and deeper relationships with customers and to maximize the lifetime value of a customer to the organization [11].

From this perspective, it is important to understand customer behavior through analyzing customer information to differentiate between customers, to identify the most valuable customers over time, and to increase customer loyalty by providing customized products and services [3]. Moreover, it is also important to predict the customer's purchase behavior.

In today's environment, most companies contact and serve customers or customer groups by utilizing a range of commercially viable channels. To understand their customers with unified view of the customer, companies try to integrate an abundance of data collected at the multiple channels, such as Web browsing, purchase

* Corresponding author.

behavior, complaints, and demographics. Furthermore, companies divide customers into numerous groups with similar preferences and examine the distinct characteristics of each group to determine the most profitable segments.

However, experience shows that business is ceaselessly changing and that customers continue to evolve over time. The customer segments and related knowledge discovered from multiple data sources change over time as the customer base changes [6]. Much research on the customer until now has assumed that customer segments and their members are stable. This means that the knowledge and predictions on customers are valid during a particular period. In addition, most existing prediction methods are fundamentally based on numerical, historical data patterns using a simple regression or neural network technique. However, in a real world situation, because of sudden fluctuations or peaks caused by internal and external events such as promotion, new product launching, and customer support policy, the assumption of status quo is not appropriate.

To resolve these problems and to focus on customers, this paper keeps track of customers' shifts among customer segments to monitor changes in the segments over time. It then investigates customer segment knowledge to build a customer life cycle and to predict customer segments' behavior patterns, which are helpful in responding appropriately in time and exercising customer-centric strategies.

2 Literature on Customer Segmentation

Segmenting customers assumes that customers exhibit heterogeneity in their preferences and buying behavior [7]. Focusing on customer segments with relatively homogeneous requirements can be a basis of satisfying these diverse customers more effectively. In this context, Bergeron (2002) defines customer segmentation as either a process of aggregating individual customers into groups of likely behavior, or an analysis method for identification and allocation of resources among identified segments [1].

According to both the academic and practitioner's literature, segmentation design schemes greatly depend on factors such as the measures used for segmentation, the number of resulting segments, the view about change over time, the segmentation techniques used, and the number of customers selected. The segmentation variables consist of either one or a combination of the following: demographic, geographic, psychographic, or behavioral purchase patterns [2]. Behavioral segmentation, including RFM (Recency, Frequency, and Monetary) or FRAT (Frequency, Recency, Amount, and Type) schemes, provides more knowledge of each customer's actual spending preferences and more accurate behavior predictions than other segmentations, which are generally useful [5]. This is because, as these researchers propose, behavioral measures provide information on what customers do.

In a customer segmentation design, the most common assumption is that a market is relatively stable, segments are unchanging, and the people who belong to them are unchanging over time. However, if the market is unstable, this assumption should be relaxed [4]. One way to predict the instability is through an occasion-based design, assuming that people vary in their needs across occasions of product purchase. Another way is to consider time-segmented customers through repeated measurements of the same customers at different points in time.

In general, methods for customer segmentation are divided into two distinct areas. One area deals with conventional statistical techniques, including *k*-means algorithm, discriminant analysis, and logistic regression. The other considers machine learning techniques, such as neural networks. West et al. (1997) suggested that neural networks are more accurate in classification than statistical techniques [12]. Especially a SOM does a good job in segmentation, compared with statistical methods [8].

Customer segmentation can incorporate all the customers or it can be limited to a sample thereof [13]. If a sample is used as the basis for segmentation, management should predict how other members of the universe of all customers fall into each group. Management then has to draw conclusions about universe via inferential statistics [9].

3 Time-Varying Customer Knowledge

Fig. 1 illustrates how to derive customer segment knowledge over time.

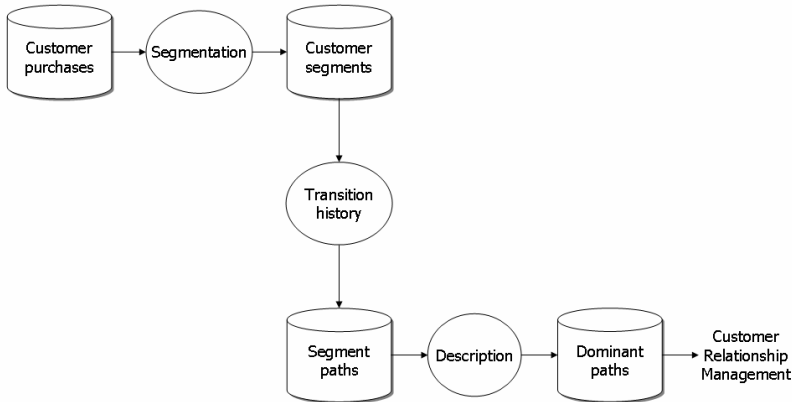


Fig. 1. Overall procedure for dynamic customer segmentation

This study focuses on discovering customer segment knowledge based on dynamic segmentation, so segmenting customers based on behavioral variables includes each customer's product usage. Segmentation by product usage uses a method called RFM analysis, which clusters customers based on recency, frequency, and monetary values. Given numerous transactions from a customer purchase database, customer segmentation divides customers into peer groups with common needs and identifies a few customer generalizations that represent the dominant characteristics present in the customer's purchase behavior. Some robust algorithms used to perform clustering include the self-organizing map (SOM), which uses a neural clustering method to divide the retailer's customers into numerous groups with similar RFM values, and to assign each customer to the resulting customer segments.

The buying behaviors of customers, however, vary greatly over time. Thus, the customer purchase database also changes over time, and the behavior patterns discovered

from that database are valid only within a certain period, and then become obsolete. This is why dynamic customer and customer segment analysis is conducted in the CRM. Customer segmentation based on product usage is performed five times and the history of the shifts from segment to segment can be observed.

Tracking each customer's segment shifts results in a sequential time series of segment shift behaviors for each customer. These data can compose an individual transition path that represents a single customer's history of shifts. Collecting individual transition paths together can reveal a dominant transition path, which explains the common histories most customers may pass through. Among others, two types of dominant paths are of interest in this study: one leading to a loyal segment and the other leading to a vulnerable segment. Deriving dominant paths that lead to a loyal segment is helpful in identifying customers who have similar patterns to the paths and in developing their careers to become loyal customers eventually. Identifying dominant paths leading to a vulnerable segment is helpful in preventing customers on the paths from leaving for a competitor. Predicting customer behaviors makes it possible to decide whether the loyal customer segment will shrink or not, which may indicate loss or increase in profit obtaining in the future, and to measure and evaluate the effectiveness and efficiency of targeted marketing campaigns or promotions.

4 Application to Retail Customers

A cluster analysis of product usage or purchase can be used to segment customers. Purchase transactions for segmentation purposes display four features: a customer number or customer ID, Recency value, Frequency value, and Monetary value.

4.1 Segmenting Customers

Customers from the target retailer were segmented five times during 15 months according to their purchase data from July 2003 to September 2004. The length of the time window to create new segmentation was set to three months (i.e., one quarter). Determining this re-creation point is difficult; however, some heuristics of a domain expert is helpful in determining the re-creating time windows. A self-organizing map is used to clustering customers.

A SOM, also called a Kohonen network, performs unsupervised clustering. It tries to uncover patterns in the set of input fields and clusters the data set into distinct groups with no target field. Records within a group or cluster tend to be similar to each other, and records in different groups are dissimilar. A SOM uses competitive learning (i.e., a winner-take-all algorithm). When an input pattern is imposed on the neural network, the algorithm selects the output node with the smallest Euclidean distance between the presented input pattern vector (\hat{X}) and its weight vector (\hat{W}_j).

Only this winning neuron generates an output signal from the output layer; all other neurons in the layer have a zero output signal. Because learning involves weight vector adjustment, only the neurons in the winning neuron's neighborhood can learn with this particular input pattern. They do this by adjusting their weights closer to the input vector, according to equation 1.

$$w_j(n+1) = \begin{cases} w_j(n) + \eta(n)[x(n) - w_j(n)], & j \in N(n) \\ w_j(n), & \text{otherwise} \end{cases} \quad (1)$$

where η is a learning rate and N is a neighborhood function.

Table 1 shows four dominant customer segments which are apparent in period T1. It shows the fraction of total customers assigned to each segment and the most significant characteristics such as average recency, frequency, and monetary values. After comparing the average RFM values of each output node with the total average RFM values of all customers, if each average is bigger than the overall average (mean), a character ‘↑’ is given to that value. If that is not the case, a character ‘↓’ is given.

Table 1. Characteristics of customer segments for the T1, derived by three-by-three SOM

Output node	Fraction of customers	Recency (Avg)	Frequency (Avg)	Monetary (Avg)	Customer segment
00	8,694(9%)	2.18	26.39	1,530,595	
10	20,221(21%)	4.67	9.77	885,755	R↓F↑M↑
11	1,881(2%)	21.95	8.18	762,462	
01	891(1%)	32.61	8.99	693,321	R↑F↑M↑
02	18,169(19%)	72.37	1.54	196,135	
12	11,784(12%)	39.05	2.27	268,372	R↑F↓M↓
20	15,115(15%)	2.79	2.99	344,091	
21	12,627(13%)	11.17	2.66	385,047	R↓F↓M↓
22	8,474(9%)	22.55	2.48	337,304	
Grand Avg	97,856(100%)	23.84	6.18	540,764	

Table 2. Results of the successive segmentation

# of customers	T1	T2	T3	T4	T5
R↓F↑M↑	30,796 (31%)	36,435 (29%)	39,408 (28%)	51,534 (35%)	64,196 (42%)
R↓F↓M↓	36,216 (37%)	41,525 (33%)	25,858 (18%)	17,306 (12%)	12,131 (8%)
R↓F↓M↑	-	-	11,789 (8%)	13,833 (9%)	-
R↑F↓M↓	29,953 (31%)	46,827 (38%)	63,159 (45%)	65,274 (44%)	77,337 (50%)
R↑F↑M↑	891 (1%)	-	-	-	-
Total	97,856 (100%)	124,787 (100%)	140,214 (100%)	147,947 (100%)	153,664 (100%)

Customers who reside in segment $R\downarrow F\uparrow M\uparrow$ can represent loyal ones who are frequent and big shoppers (loyal segment). Customers who belong to segments $R\uparrow F\uparrow M\uparrow$ or $R\uparrow F\downarrow M\downarrow$ are much more likely to become vulnerable customers, based on above-average value in recency (vulnerable segment). Segments $R\downarrow F\downarrow M\downarrow$ or $R\downarrow F\downarrow M\uparrow$ can represent new customers, given below-average values in recency and frequency (new-comer segment).

Table 2 summarizes the resulting segments and the fraction of customers assigned to each segment, after a five-time segmentation.

4.2 Deriving Customer Transition Knowledge

Five-time segmentation makes it possible to combine segment shift histories into a transition path. These shifts could result from changes in the natural life cycle of customers, or from external factors such as a rise in the standard of living.

Table 3 summarizes changes in the number of customers in the segments over successive quarters. The number of loyal customers and the number of vulnerable ones increase while the increasing rate of the vulnerable segment ($R\uparrow F\downarrow M\downarrow$) is steeper than that of profitable and loyal segment ($R\downarrow F\uparrow M\uparrow$). The increasing rates reverse at period T3. New acquisitions of customers ($R\downarrow F\downarrow M\downarrow$) show a tendency to shrink slowly after period T2: 41,525 (T2) \rightarrow 25,858 (T3) \rightarrow 17,306 (T4) \rightarrow 12,131 (T5).

Table 3. Changes in the number of customers for each segment over time

Segment	T1	$\Delta(T2-T1)$	$\Delta(T3-T2)$	$\Delta(T4-T3)$	$\Delta(T5-T4)$
$R\downarrow F\uparrow M\uparrow$	30,796	5,639 (18.31%)	2,973 (8.16%)	12,126 (30.77%)	12,662 (24.57%)
$R\uparrow F\downarrow M\downarrow$	29,953	16,874 (56.33%)	16,332 (34.88%)	2,115 (3.35%)	12,063 (18.48%)
$R\downarrow F\downarrow M\downarrow$	36,216	5,309 (14.66%)	-15,667 (-37.73%)	-8,552 (-33.07%)	-5,175 (-29.90%)

Table 4 tracks changes in the number of customers who remain on the same segments over time. No matter what the segments may be, the number of customers who remain on the same segments decreases over time. Among others, the decreasing rate of loyal customers is the smallest. This implies that a customer retention strategy, which keeps profitable customers longer, proves effective.

Table 4. Changes in the number of customers who remain on the same segments over time

Segment	T1	$\Delta(T2-T1)$	$\Delta(T3-T2)$	$\Delta(T4-T3)$	$\Delta(T5-T4)$
$R\downarrow F\uparrow M\uparrow$	30,796	-6,219 (-20.19%)	-5,044 (-20.52%)	-3,187 (-16.32%)	-1,109 (-6.78%)
$R\uparrow F\downarrow M\downarrow$	29,953	-12,165 (-40.61%)	-4,718 (-26.52%)	-3,052 (-23.35%)	-1,373 (-13.71%)
$R\downarrow F\downarrow M\downarrow$	36,216	-20,035 (-55.32%)	-10,243 (-63.30%)	-5,238 (-88.21%)	-602 (-86.0%)

To derive dominant paths, it is necessary to identify all the possible segment shifts and count the number of customers who follow each shift pattern. The larger the number of customers, the more dominant the path is. For example, a path, $R\downarrow F\uparrow M\uparrow \rightarrow R\downarrow F\uparrow M\uparrow \rightarrow R\downarrow F\uparrow M\uparrow$, is a dominant path of length three that leads to a loyal segment ($R\downarrow F\uparrow M\uparrow$), with a probability of 42.0%. A path, $R\uparrow F\downarrow M\downarrow \rightarrow R\uparrow F\downarrow M\downarrow \rightarrow R\uparrow F\downarrow M\downarrow$, shows dominant transition leading to a vulnerable segment ($R\uparrow F\downarrow M\downarrow$), with a probability of 20.9%.

5 Conclusion and Discussions

This study proposed a segment-based customer knowledge discovery method. This study tried to resolve one of the most fundamental problems that might arise from the customer segmentation analysis: the changing characteristics of customers in a segment. Through deriving descriptive knowledge from the customer segment knowledge, this study attempted to lessen numerical data's unexpected randomness and to make more stable predictions about customer segment transitions.

Future research can extend the work this study presents in several ways. First, building separate knowledge for each different segment and then combining results from multiple sources of knowledge can be a way to improve customer knowledge, since they can potentially offer complementary information about the derived patterns. Another way to improve knowledge accuracy can be considering alternative segmentation variables with better predictive performance. Second, there can be alternative ways to cluster customers. Because the choice of techniques influences the analysis results, a broader analysis of the methods used in this paper is needed to examine whether the results hold up with alternate methods.

References

1. Bergeron, B.: *Essentials of CRM: a guide to customer relationship management*. John Wiley & Sons, New York (2002)
2. Drozdenko, R.G., Drake, P.D.: *Optimal database marketing: strategy, development, and data mining*. Sage Publications, CAL (2002)
3. Gulati, R., Garino, J.: Get the right mix of bricks and clicks. *Harvard Business Review* 78 (2000) 107-114
4. Ha, S.H., Bae, S.M., Park, S.C.: Customer's time-variant purchase behavior and corresponding marketing strategies: an online retailer's case. *Computers & Industrial Engineering* 43 (2002) 801-820
5. Kohavi, R., Provost, F.: Applications of data mining to electronic commerce: a special issue of data mining and knowledge discovery 5(1-2). Kluwer, Dordrecht (2001)
6. Kracklauer, A.H., Mills, D.Q., Seifert, D.: *Collaborative customer relationship management: taking CRM to the next level*. Springer, Berlin (2003)
7. Linoff, G.S., Berry, M.J.A.: *Mining the web: transforming customer data into customer value*. John Wiley & Sons, New York (2001)
8. Mangiameli, P., Chen, S.K., West, D.: A comparison of SOM neural network and hierarchical clustering methods. *European Journal of Operational Research* 93 (1996) 402-417

9. Morwitz, V.G., Schmittlein, D.C.: Testing new direct marketing offerings: the interplay of management judgment and statistical models. *Management Science* 44 (1998) 610-628
10. Peppard, J.: Customer relationship management (CRM) in financial services. *European Management Journal* 18 (2000) 312-327
11. Peppers, D., Rogers, M., Dorf, R.: Is your company ready for one-to-one marketing. *Harvard Business Review* 77 (1999) 151-160
12. West, P.M., Brockett, P.L., Golden, L.L.: A comparative analysis of neural networks and statistical methods for predicting consumer choice. *Marketing Science* 16 (1997) 370-391
13. Wyner, G.A.: Segmentation design. *Marketing Research* 4 (1992) 38-41

Mining of Flexible Manufacturing System Using Work Event Logs and Petri Nets

Hesuan Hu, Zhiwu Li, and Anrong Wang

School of Electro-Mechanical Engineering, Xidian University,
Xi'an, Shaanxi 710071, P.R. China
hshu@mail.xidian.edu.cn

Abstract. One of buzzwords for modern manufacturing industry are flexible manufacturing systems (FMS), in which several machines are interlinked by an automated information and material flow system. Description and control upon these systems are of prominent significance. This paper is concerned with mining and construction of the established FMS from work event logs. A novel Petri nets based algorithm is developed to implement such an idea. When an FMS is mined and constructed, its corresponding Petri net is used to evaluate, analyze, and control the system. Theoretical and experimental results are illustrated to show the effectiveness and efficiency of this approach.

1 Introduction

The development and application of FMS, which ensure a fully automated implementation for manufacturing, have illustrated substantial competence in many high-tech industries [8][9]. However, FMS is not a magic that can do everything. Changing consumer preferences, fluctuating interest rates, and new materials may dwarf an FMS that is preplanned in detail. Thus reconstruction along with analysis, control, and optimization of the layout and work scheduling is indispensable when an FMS is in operation. To keep costs within controllable limits, reconstruction methods that are practicable for both the suppliers and the end users of FMS are strongly recommended. As mentioned previously, FMS is a product of information revolution, whose work events or transactions have been logged in various databases. Normally, these data are used as archives to guarantee production and quality management by tracing back to the operator of each product. In the area of total quality management, they are also referred as history records. Glancing at these data, one can see screen-shots with respect to the FMS. These logs typically contain activities referring to work steps during an operation. The resources exploited and the operators involved are also recorded to detail the process. The whole production process can be identified when all these records are sequenced together. The reconstruction of FMS is just to extract its corresponding information embedded in logs, thus modelling the architecture and configuration of them with various mathematical tools [2][7]. Reconstruction of an FMS from work event logs can benefit designers and end users in, at least, two aspects. To begin with, evaluation upon the performance of an FMS

can be carried out in spot. In many cases, managers and operators are eager to know what is the bottleneck with respect to their particular FMS components and how to reach the maximal output of the overall system [8]. They cannot rely on the given results provided by the designers because of the drastic discrepancy between the provider's simulation environment and the user's production conditions. Redesign and re-implementation of the FMS directly are always beyond either the user's ability or the provider's responsibility. Second, conformance test can be accomplished with the reconstructed model from work event logs. With the recent introductions of total quality management (TQM) and ISO9000/2000 quality qualification, all systems should be calibrated to conform to their original models [3]. Therefore, this conformance is required for the accomplishment of a quality-ensured manufacturing system. Although the concept to extract workflow model from event logs is not novel, the approach conducted in this paper is the initial one to construct the model of FMS and many newly arising difficulties are encountered during this stage. Compared with traditional researches, resources are introduced in our investigation for their prominent significance during the evaluation and analysis of the system.

2 Related Works and Our Contributions

The concept of process mining based on workflow logs is firstly introduced in [2]. According to it, a process can be divided into small, unitary actions, namely activities. The relationships between different activities are denoted as dependencies. In their approach, systems are modelled as directed acyclic or cyclic graphs. The graph vertices represent the activities while the edges represent dependencies between them. In [7], three different methods to model a process are discussed in the area of software engineering, which are the Rnet method, the Ktail method, and the Markov method, respectively. According to such an investigation, a process can be represented as a finite state machine (FSM). This method is infeasible as many systems are complex such that the state explosion problem occurs. In [3], the work processes are directly modelled as Petri nets, namely workflow nets. Since these previous works focus on the area of workflow management, the resource perspective is not taken into account. However, in many other applications such as FMS, resources play a significant role, if not all, to affect properties, i.e., deadlock, liveness, ratio of throughput, etc [8]. Furthermore, both the managers and the operators concern deeply with the utilization of each resource. Many problems and bottlenecks can only be detected and overcome with the resources involved in the resultant models. This phenomenon happens due to so many concurrencies and collaborations emerging in systems like FMS [8]. This paper, to the most knowledge of the authors, is the first one to introduce resources into process mining. Moreover, reconstruction of FMS when they have been designed and installed is also a novel idea that is a promising approach and prime concern for worldwide enterprisers.

3 Merged Enhanced Workflow Nets with Resources

Elements involved in an FMS are well classified as resources, activities, and events according to their respective functions during the evolution of systems [3]. Resources, just as their names mean, denote machines or tools contained in the systems. Activities are the atomic units of a work process, which cannot be divided further. Activities can only stay at one and only one of the following states, i.e., initiation, preparation, and execution [3]. Events mean occurrences of activities. Moreover, manufacturing systems cannot be described elaborately only with the introductions of independent activities, events, and resources. To formally describe manufacturing systems, four standard and basic interactive relationships should be specified, which are AND-join, AND-split, OR-join, and OR-split [4]. To establish the model of the event-driven system, a novel class of Petri net, called workflow nets (WF-nets) [5][6], is predefined.

Definition 1 ([3]). A Petri net $N=(P, T, F)$ is a workflow net (WF-net) if and only if P contains a source place i such that $\bullet i=\phi$; P contains a sink place o such that $o\bullet=\phi$; every node $x\in P\cup T$ is on the path from i to o ; a newly generated $\bar{N}=(P, T\cup\bar{t}, F)$ is strongly connected if and only if $\bar{t}=\bullet i=o\bullet$, where \bar{t} is a newly added transition; only one initial token is contained in i such that $M_0(i)=1$.

Definition 2 ([8]). A Petri net $N=(P, T, F)$ is an enhanced workflow net (EWF-net) if and only if T contains a source transition i such that $\bullet i=\phi$; T contains a sink transition o such that $o\bullet=\phi$; every node $x\in P\cup T$ is on the path from i to o ; a newly generated $\bar{N}=(P\cup\bar{p}, T, F)$ is a state machine, and strongly connected if and only if $\bar{p}=\bullet i=o\bullet$, where \bar{p} is a newly added place which should be initially marked such that $M(\bar{p})>0$.

Definition 3 ([8]). A Petri net $N=(P\cup P_R, T, F)$ is an enhanced workflow net with resources (EWFR-net) if and only if the subnet generated by $X=(P, T, F)$ is an EWF-net; $P_R\neq\phi$ and $P\cap P_R=\phi$; $\forall p\in P, \forall\bullet t_x\in P, \exists t_y\in P$, such that $\bullet t_x\cap P_R = P_R\cap t_y\bullet=\{r_p\}$; the following statements are verified: $\forall r\in P_R, \bullet\bullet r\cap P=r\bullet\bullet\cap P\neq\phi, \forall r\in P_R, \bullet r\cap r\bullet=\phi$.

Definition 4 ([8]). A Petri net $N=(P\cup P_R, T, F)$ is a merged enhanced workflow net with resources (MEWFR-net) if and only if an EWFR-net is also an MEWFR, let $N_i=(P_i\cup P_{R_i}, T_i, F_i), i\in(1, 2)$ be two EWFR-nets such that $P_1\cap P_2=\phi, P_{R_1}\cap P_{R_2}=P_c\neq\phi$, and $T_1\cap T_2\neq\phi$, then the net $N=(P\cup P_R, T, F)$ resulting from the composition of N_1 and N_2 via P_C can be defined as follows: $P=P_1\cup P_2, P_R=P_{R_1}\cup P_{R_2}, T=T_1\cup T_2$ and $F=F_1\cup F_2$. Such a merging process is denoted as $N_1 \circ N_2$.

Definition 5 ([8]). A Petri net $N=(P\cup P_R, T, F)=N_1\circ N_2$ is a merged enhanced workflow net with reasonably marked resources (MEWFR-net) if and only if $\forall i\in\{1, 2\}, \forall p\in P_1\cup P_2, m(p)=0; \forall i\in\{1, 2\}, \forall r\in P_{R_i}, m(r)=m(r_i); \forall i\in\{1, 2\}, \forall r\in P_C, m(r)=\max(m(r_i))$.

4 Analysis of Work Event Logs

Similar to the process to establish grammar based on sufficient number of sentences, it is of prominent significance to abstract model of a system out of its corresponding work event log. Such an approach is inspired in [2], whereas successful application is accomplished in [4]. In their research domains, i.e., e-commerce, a deep insight is given upon work event logs without the introduction of resources due to the aforementioned reasons. In our research, resources are indispensable elements that should be sufficiently involved in work event logs.

Definition 6 ([1]). *The work log is a list of activity records (P, A) , where P represents the work processes and A represents the activities. The orders of both P and A in the list correspond to their emerging sequences physically.*

Definition 7. *The enhanced work log is a list of activity records (P, A, T) with P representing the work processes, A representing the activities, and $T=(S, E)$ representing the time corresponding to the activities, where S means the start time, and E means the end time.*

Definition 8. *The enhanced work log with resources is a list of activity records (P, A, R, T) with P representing the work processes, A representing the activities, R representing the resources involved, and $T=(S, E)$ representing the time corresponding to the activities, where S means the start time, and E means the end time.*

Definition 9 ([1]). *Let A and B be two activities in a work log, activity A precedes B if and only if B starts only when A terminates in the work process where they both appear. Such a relationship is called sequential relation and denoted by $A \succ B$ or $B \prec A$.*

Property 1. The sequential relationship is transitive, whereas not self-reflexive and symmetrical.

Proof. Let A , B , and C be three different activities emerging in a work event log. When $A \succ B$ and $B \succ C$ hold, it is trivial that $A \succ C$ is also true, which means that operation \succ is transitive. Since $A \succ A$ does not hold, operation \succ is not self-reflexive. Since $A \succ B$ does not mean $B \succ A$, \succ is not symmetrical.

Definition 10 ([1]). *Let A , B , C , and D be four activities in a work log. Activity A is in parallel with B if and only if neither $A \succ B$ nor $B \succ A$ holds, whereas both $C \succ A \cap C \succ B$ and $A \prec D \cap B \prec D$ hold. Such a relationship is called parallel relation and denoted by $A \parallel B$.*

Property 2. The parallel relationship is transitive, self-reflexive and symmetrical.

Proof. Let A , B , and C be three different activities emerging in a work event log. When $A \parallel B$ and $B \parallel C$ hold, it is trivial that $A \parallel C$ is also true, which means that operation \parallel is transitive. Since $A \parallel A$ holds, operation \parallel is self-reflexive. Since $A \parallel B$ leads to $B \parallel A$, \parallel is not symmetrical.

Definition 11 ([1]). Let A, B, C , and D be four activities in a work log. Activity A is irrelative with B if and only if neither $A \succ B$ nor $B \succ A$ holds. Moreover, neither $C \succ A \cap C \succ B$ nor $A \prec D \cap B \prec D$ holds. Such a relationship is called irrelative relation and denoted by $A \# B$.

Property 3. The irrelative relation is neither self-reflexive nor transitive, whereas it is symmetrical.

Proof. Let A, B , and C be three different activities emerging in a work event log. Since $A \# B$ and $B \# C$ do not mean $A \# C$ hold, operation $\#$ is not transitive. Since $A \# A$ does not hold, operation $\#$ is not self-reflexive. Due to $A \# B$ does not lead to $B \# A$, $\#$ is not symmetrical.

5 Mining FMS Using Event Logs

In the sequel, when mentioning an event log, we mean a log that contains sufficient information about an FMS, which implies that every activity contained in the FMS emerges at least one time, and so does every work process.

Definition 12 ([1]). Let $M_{\alpha \times \beta}$ be a matrix derived from an event log, where $m_{\alpha \times \beta} \{ \succ, \prec, \parallel, \# \}$ is the element of M , and $\alpha, \beta \in \{A, B, \dots, Z\}$. The value of $m_{\alpha \times \beta}$ denotes the corresponding relationship between activities α and β , then $M_{\alpha \times \beta}$ is called the order relation matrix with respect to the given log.

Definition 13. Let $R_{\alpha \times \beta}$ be a matrix derived from an event log, where $r_{\alpha \times \beta} \in \{ \times, \surd \}$ is the element of R , and $\alpha, \beta \in \{A, B, \dots, Z\}$. The value of $m_{\alpha \times \beta}$ denotes the occupying relationship between activity α and resource β , where \surd and \times means the occupying and non-occupying relations, respectively. Then $R_{\alpha \times \beta}$ is called the occupying relation matrix with respect to the given log.

Given an event log with several processes and activities, assuming that every activity appears in the log at least once. The algorithm to find its corresponding MEWFR-net can be summarized as follows.

1. Generate the initial net system (P, T, F) with $P = \phi$ and $T = \{ \text{Source}, \text{Sink} \}$, where Source and Sink mean the source and sink transitions, respectively.
2. Add the places with respect to the logged activities, and the relations among them are established based on the order relation matrix such that Petri net component $\bigcirc \xrightarrow{A} \bigcirc$ or $\bigcirc \xleftarrow{A} \bigcirc$ is generated when $A \succ B$ or $A \prec B$ holds. Once activity A is detected to be followed or preceded by more than two activities, OR-split or Or-join relation is preferable.
3. For every generated place, e.g., activity A , a resource place, e.g., resource R , is allocated to it, e.g., $\bullet A = A \bullet = R$.
4. Return net system (P, T) .

6 Experimental Results

Table 1 shows an example of work event log whose respective Gantt chart is shown in Figure 1. According to them, there are 10 primitive activities involved in such a production cell. Four processes, i.e., processes 1, 2, 3, and 4, are sampled, which contain sufficient information on the whole production procedure. In Tables 2 and 3, the order relation matrix and occupying relation matrix corresponding to this log are derived. Figure 2 presents the resultant MEWFR-net mined from the given log, where R1, R2, R3, G, and L mean Robot 1, Robot 2, Robot 3, Grinder, and Lathe, respectively.

Table 1. Work event log

Process	Activity	Resource	Start	End	Process	Activity	Resource	Start	End
1	A	Robot 1	8:00	8:30	3	F	Robot 3	8:00	8:20
1	B	Grinder	8:30	8:40	4	F	Robot 3	8:05	8:25
2	A	Robot 1	8:10	8:20	4	G	Lathe	8:25	8:40
1	C	Robot 2	8:40	9:10	3	G	Lathe	8:20	8:30
2	B	Grinder	8:20	8:50	4	I	Robot 2	8:40	9:00
1	E	Robot 3	9:10	9:20	3	H	Grinder	8:40	9:00
2	D	Lathe	8:50	9:20	4	J	Robot 1	9:00	9:10
2	E	Robot 3	9:20	9:30	3	J	Robot 1	9:00	9:10

Table 2. Order relation matrix extracted from Table 1

A\A	A	B	C	D	E	F	G	H	I	J
A	#	γ	γ	γ	γ	#	#	#	#	#
B	γ	#	γ	γ	γ	#	#	#	#	#
C	γ	γ	#		γ	#	#	#	#	#
D	γ	γ		#	γ	#	#	#	#	#
E	γ	γ	γ	γ	#	#	#	#	#	#
F	#	#	#	#	#	#	γ	γ	γ	γ
G	#	#	#	#	#	γ	#	γ	γ	γ
H	#	#	#	#	#	γ	γ	#		γ
I	#	#	#	#	#	γ	γ		#	γ
J	#	#	#	#	#	γ	γ	γ	γ	#

Table 3. Occupying relation matrix between activities and resources

R\A	A	B	C	D	E	F	G	H	I	J
Robot 1	√	×	×	×	×	×	×	×	×	√
Robot 2	×	×	√	×	×	×	×	×	√	×
Robot 3	×	×	×	×	√	√	×	×	×	×
Lathe	×	×	×	√	×	×	√	×	×	×
Grinder	×	√	×	×	×	×	×	√	×	×

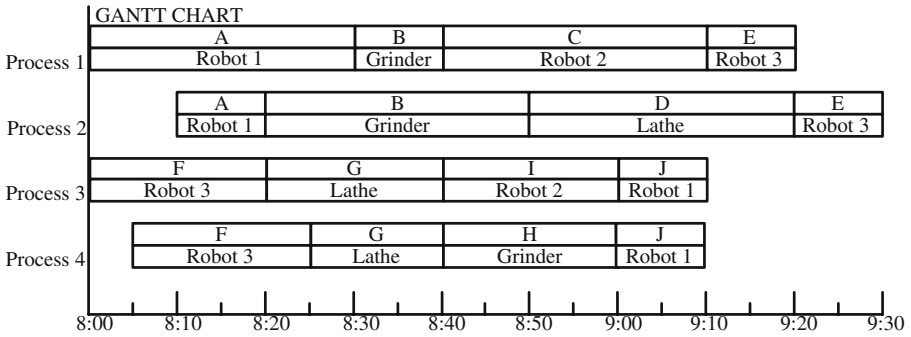


Fig. 1. Gantt chart derived from Table 1

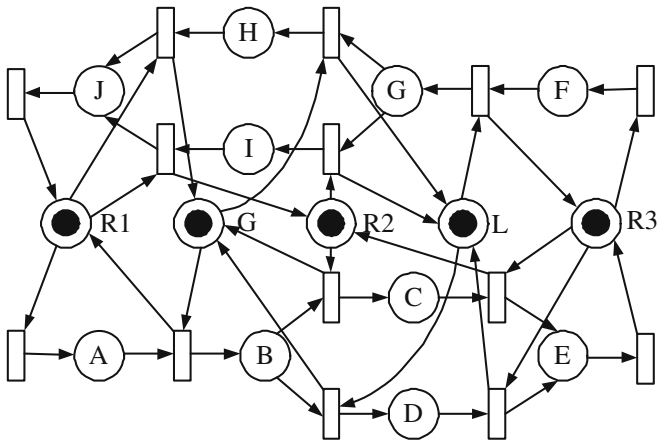


Fig. 2. EWFR-net generated from Tables 1 and 2

When the Petri net model of an FMS is mined from the work event log, many analysis methods and control policies based on Petri net theory can be applied to it. Thus a refined system is obtained. Quantitatively, the Petri net model can be converted to be a Markov chain to calculate parameters such as the ratio of utilization with respect to every resource and the throughput of the whole system. Qualitatively, properties such as deadlock-freeness and liveness can be detected and controlled [8][9].

7 Concluding Remarks

In this paper, a deep insight is given upon the mining of FMS based on their respective work event logs. Different from recently available process mining algorithms, resources are taken into consideration due to their indispensable roles during analysis of systems like FMS. Petri nets are used as mathematical tools

and a special class of net systems suitable to describe FMS is well defined. Further research will focus on the following areas. First, an algorithm should be developed to filter noise, i.e., error operations, missing of some records, contained in the work event system. Such a problem seems to be perplexed since no efficient method can work apart from the successful introduction of threshold in [2]. Second, a fast algorithm to abstract the models should be well investigated such that the whole abstraction process can be accomplished online. Such a research is promising since it ensures an adaptively controlled FMS, which implies that errors arising can be detected and controlled during the evolution of an FMS.

References

1. Wen L. J., Wang J. M., Van der Aalst W. M. P., Wang Z. and Sun J. G.: A novel approach for process mining based on event types. BETA Working Paper Series, Wp118, Eindhoven, University of Technology, Eindhoven, (2004).
2. Agrawal R., Gunopuls D., and Leymann F.: Mining process models from workflow logs. Proceedings of International Conference on Extending Database Technology, (1998) 469-483.
3. Van der Aalst W. M. P., Van Dongen B. B.: Workflow verification: finding control-flow errors using Petri-net-based techniques. Proceedings of International Conference on Business Process Management, Lecture Notes in Computer Science, **1806** (2000) 161-183.
4. Van der aalst W. M. P., Weijters A. J. M. M., and Maruster L.: Workflow mining: discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering, **16** (2004) 1128-1142.
5. Van der aalst W. M. P., Van Dongen B. F., Herbst J., Maruster L., Schimm G., Weijters A.J.M.M.: Workflow mining: a survey of issues and approaches. Data and Knowledge Engineering, **47** (2003) 237-267.
6. Van der aalst W. M. P., Van Dongen B. F.: Discovering workflow performacne models from timed logs. Proceedings of International Conference on Engineering and Deployment of Cooperative Information Systems, Lecture Notes in Computer Science, **2480** (2002) 45-63.
7. Jonathan E. C., Alexander L. W.: Discovering models of software processes from event-based data. ACM Transactions on Software Engineering and Methodology, **7** (1998) 215-249.
8. Ezpeleta J., Colom J. M., Martinez J.: A Petri net based deadlock prevention policy for flexible manufacturing systems. IEEE Transactions on Robotics and Automation, **11** (1995) 173-184.
9. Li Z. W., Zhou M. C.: Elementary siphons of Petri nets and their application to deadlock prevention for flexible manufacturing systems. IEEE Transactions on Systems, Man, and Cybernetics, **34** (2004) 38-51.

Improved Genetic Algorithm for Multiple Sequence Alignment Using Segment Profiles (GASP)

Yanping Lv¹, Shaozi Li^{1,*}, Changle Zhou¹, Wenzhong Guo², and Zhengming Xu¹

¹Intelligent Information Technology Lab., Department of Computer Science, Xiamen University, Xiamen, 361005, China

Catlet.lyp@gmail.com, {szlig, dozero}@xmu.edu.cn

²Department of Computer Science, Fuzhou University, Fuzhou, 350002, China
guowenzhong@fzu.edu.cn

Abstract. This paper presents a novel genetic algorithm (GA) for multiple sequence alignment in protein analysis. The most significant improvement afforded by this algorithm results from its use of segment profiles to generate the diversified initial population and prevent the destruction of conserved regions by crossover and mutation operations. Segment profiles contain rich local information, thereby speeding up convergence. Secondly, it introduces the use of the norMD function in a genetic algorithm to measure multiple alignment. Finally, as an approach to the premature problem, an improved progressive method is used to optimize the highest-scoring individual of each new generation. The new algorithm is compared with the ClustalX and T-Coffee programs on several data cases from the BALiBASE benchmark alignment database. The experimental results show that it can yield better performance on data sets with long sequences, regardless of similarity.

1 Introduction

Multiple sequence alignment (MSA) has become an essential tool in molecular biology. It has been used for the analysis of protein families, comprehension of their evolutionary trends and detection of remote homologues, genome annotation and analysis and a host of other tasks. When sequences are similar to each other, virtually any alignment method will produce good results. However, evolutionary divergence in families can result in the pair similarity between family members being so low as to be indistinguishable from chance [1]. The development of accurate, reliable multiple alignment programs capable of handling large numbers of very divergent sequences, is therefore of major importance.

Unfortunately, accurate multiple alignments can be difficult to build. The optimization algorithms largely fall into two categories: progressive and iterative algorithms. In progressive methods, an MSA is built up gradually by aligning the closest se-

* Corresponding author.

This work is supported by the Natural Science Fund, Science & Technology Project of Fujian Province (Project Number: A0310009, 2001J005), China, the 985 Innovation Project on Information Technique of Xiamen University(2004-2007), China.

quences first and successively adding in the more distant ones. A typical program is ClustalX [2]. It constructs a global alignment over the entire length of the sequences. It has the advantages of speed and simplicity. However, due to its 'greediness', errors made in the first alignments cannot be rectified later as the rest of the sequences are added in.

Iterative strategies have been applied to refine and improve the initial alignment. DIALIGN [3] constructs multiple local alignments based on segment-to-segment comparisons. Other iterative algorithms aim at building global alignments, two examples are SAGA [4], based on a genetic algorithm, and HMM [5]. For low-identity (low-similarity) sequences, DIALIGN will produce low quality MSAs due to its local nature. HMM does not correctly align structurally similar regions existing in some, but not all, sequences.

SAGA has been demonstrated to obtain better MSAs than other programs for divergent sequences [1]. It succeeds in aligning critical motifs and conserved core structure of protein families. However, the length and size of sequences it can handle is restricted due to its limit speed and it may sometimes tend to diverge away from the correct alignment in the presence of an 'orphan' sequence aligned to a family of closely related sequences, as in ref2 of the BALiBASE database.

The paper is organized as follows. Section 2 proposes an improved genetic algorithm for multiple sequence alignment. Results of experimental evaluation are given in Section 3, which contains the description of benchmark database used for comparison of algorithms, the experimental setting for each algorithm, and discussions about the results. Section 4 gives conclusions and future work.

2 The GASP Algorithm

We call our algorithm GASP, for alignment based on a genetic algorithm using segment profiles. The outline of the procedure follows.

2.1 Encoding and Initialization

For genetic algorithms, each individual in the population is a possible solution to the problem. Different encoding methods can be chosen for different problems. Here, each individual is an alignment, in SAGA. Intuitively, an alignment of the population is expressed as a string matrix consisting of characters from a given alphabet.

The challenge in initialization is to generate an diverse initial population. However, a diversified population simultaneously increases computational complexity. In existing GA-based methods, individuals in the initial population are constructed randomly. The lengths of initial alignments are bounded by a value. For highly similar sequences, it is reasonable to limit the number of gaps. For divergent sequences, however, it is likely to result in the optimal alignment being missed. In our algorithm, the diversified initial population is generated, centered on different SPs.

We have designed a simple and efficient method for finding SPs. Here, a SP is defined as a string set in which every string from every sequence is highly similar. The first step is to find all segment pairs of equal length within a finite position distance d , with sum-of-pairs score (using the PAM250 substitution matrix [6]) higher than a

threshold T_{sp} . Two similar segments always get a higher score, since PAM250 considers the similarity of residue pairs. Therefore, it is necessary to set this threshold. The position distance restricts the number of gaps.

Next we construct an SP whose segments are from different sequences. The number of segments in the SP must be greater than half of the sequence size and its norMD [7] score must be higher than a cutoff T_{md} . (The sum-of-pairs score is sensitive to the length and size of sequences, whereas the norMD score is not affected by these factors.) Finally, we extend the SP to both sides until its norMD score is less than the cutoff mentioned above, since when a SP corresponds to a structurally similar region of alignment, there is a high probability that there will be another SP located nearby.

To create one of these individuals, we randomly align two substring sets on both sides of an SP, then build up an MSA by integrating the two subalignments and the SP. As a result, each MSA is centered on a different SP. If the number of MSAs is less than the population size, the remaining individuals are randomly generated as in other GA-based methods. The final result is a diversified initial population with different individuals, most of which are centered on different SPs.

2.2 Fitness Function and Its Scaling

In this algorithm, norMD is introduced to measure the quality of an MSA. The goal of MSA is to align structurally similar regions of all sequences and to succeed in aligning regions that are structurally similar in some sequences. Sum-of-pairs can't reasonably evaluate the quality of an MSA, for it is sensitive to the length and size of sequences. NorMD was therefore suggested here for comparison purposes. Simulation experiments show that it is not sensitive to the factors mentioned above and delineates an MSA better, since it combines column scores with residue similarity scores.

Because the variance of the fitness value given by norMD is so low, a corresponding function scaling method has been used in this algorithm. The NorMD scores of most alignments obtained during the iterative procedure range between 0 and 1. This algorithm also calculated the expected offspring (EO) of an alignment on the basis of the fitness value.

$$EO_i = \frac{f_i}{\sum_j f_j / Num}$$

Here, f_i is the norMD score of the i^{th} individual and Num is the population size.

2.3 Operators

Selection: In this algorithm, an individual is selected as a parent simply based on the proportional probability of its EO.

$$P_i = EO_i / \sum_j EO_j$$

One-point crossover: The crossover can be very disruptive at the junction point. Positions in SPs are chosen as crossover sites on the basis of zero probability, to prevent destruction due to crossover. As a result, SPs as conserved regions in the initial population will be kept down until the iterative process terminates. If an SP is an excellent gene, a MSA which contains it will get a high norMD score. Otherwise, it will get a low score and be abandoned in a later generation. However, SPs also bring the problem of premature convergence. To overcome this problem, we optimize the highest-scoring MSA by rearranging it after crossover and mutation operations.

Mutation: Some positions are conserved more than others during the process of generation [8]. For this reason, we found it useful to bias the choice of the mutation site. In this algorithm, the positions in SPs are chosen for zero probability and other positions are selected as mutation sites for equal probability.

2.4 Rearrangement

A very stable local minimum makes it difficult for operators to generate an optimal MSA. To avoid being trapped in local minima resulting from SPs, we rearrange the highest-scoring MSA of every generation. During the iterative procedure, we extract all substrings from two adjacent SPs in the MSA, align them using a progressive method and incorporate all subalignments and SPs into it. We align pair sequences using an improved SPA [9] for proteins, if substrings are long or the sequence size is large; otherwise, traditional dynamic programming is used here. We simply need to rearrange one MSA. As a result, most of the conserved residues can be aligned in the same columns, without sacrificing too much time.

3 Experimental Results

3.1 Reference Alignments

In order to demonstrate the feasibility of our algorithm, we used version 3 of the BALiBASE benchmarks database [10]. BALiBASE is designed for the evaluation and comparison of MSA programs. The alignments in BALiBASE are divided into eight reference sets. Here, we used only the first two reference sets. Ref1 contains alignments of a small (<6) number of sequences which are equidistant, meaning that the percent identity between two sequences is within a specified range. Alignments in Ref2 combine three ‘orphan’ sequences (<25% identical) from ref1 with a family of at least 15 closely related sequences. Ref1 and Ref2 are divided into groups of short, medium and long sequences. For clarity of comparison, a single ‘orphan’ sequence is aligned to a family in ref2.

3.2 Alignment Quality Scoring

BALiBASE provides a module (BaliSore) that defines two scores. The sum-of-pairs score, SPS, is the ratio of the number of correctly aligned pairs of positions in the test alignment to the number of aligned pairs in the reference alignment structurally informed. The column score, CS, is the ratio of the number of correctly aligned columns in the test alignment to the number of aligned columns in the reference alignment.

Both SPS and CS range from 0.0 for no agreement to 1.0 for perfect agreement. The designers recommend SPS as the best quality score for Ref1,2,3.

While the BALiBASE scores are useful, they have limitations as measures of alignment quality. On the one hand, they only take core blocks into account and give no credit for positions between core blocks. Neither of them penalizes columns between core blocks in the test alignment that are not structurally aligned. On the other hand, neither of them measures the alignments excluded from BALiBASE benchmarks. (In Ref2, we only align one orphan to a family.) As a complementary measure of alignment quality, we also evaluate alignments using the norMD measure, where both each residue pair and each column are compared between the two alignments.

3.3 Algorithm Parameters

GASP has the following parameters: Num , the population size; P_c , the probability of crossover; P_m , the probability of mutation. Three additional parameters are d , the maximal position distance between segment pairs; T_{sp} , the sum-of-pair score threshold and T_{md} , the NorMD score cutoff. The parameter d involves the size of the alignment search space. If the d value is too small, a segment profile containing rich local sequence information cannot be found. If it is too large, too many gaps must be inserted into an alignment centered on the SP found and more time is required to find the optimal alignment. The value of d used in these experiments depends on the variance in the length of the sequences. For sequences of similar length, it is set to one-quarter the sequence length, to avoid having to insert too many gaps into an optimal alignment; otherwise, the proportion is one-half. T_{sp} and T_{md} are obtained empirically. For most experiments, good results can be produced with $T_{sp}=0$ and $T_{md}=0.5$. P_c and P_m are two main parameters in GAs. We performed some experiments to find the optimal values of these parameters. In our experiment, we empirically chose 15 values of P_c and 20 values of P_m . For each parameter composition, we ran the program 30 times and the average SPS (ASPS) of the results was obtained. We first determined the optimal P_c value as follows. For each of the 15 values, we averaged all ASPS values with different P_m values. We selected the optimal result of the 15 results and subsequently the optimal P_c could be chosen. Good performance is obtained with $Num=91$ (more individuals increases the computer load without reducing the number of algorithm iterations before convergence), and with $P_c=0.7$ and $P_m=0.065$. With these parameters, the iteration terminated at a point beyond which no better solution would usually be found.

3.4 Experiments

Using the parameters set above, GASP constructed the alignments by extracting sequences from the BALiBASE reference alignments, (60 out of 123 cases, 45 in ref1 and 15 in ref2). For the alignments selected, we also downloaded ClustalX and, T-Coffee [11] for comparison. ClustalX is one of the most commonly used tools; we used version 1.83. T-Coffee is one of the most recent tools, which generates a MSA faster and sacrifices less accuracy than SAGA, which runs too slowly for long sequences. Figures 1 and 2 show the SPS and NorMD scores, respectively, for DASP, ClustalX, and T-Coffee on benchmarks with low, medium, and high similarity.

From Figure 1, when medium or long sequences are considered regardless of similarity, GASP outperforms other tools. For medium benchmarks, on average, GASP gets a score of 80.3%, which is better than 79.1% for T-Coffee and 76.2% for ClustalX. GASP finds the best results for 11 out of 15 medium reference benchmarks. For long benchmarks, GASP is again superior to the other two tools. Its average score of 85.5% is the highest of the three, and it performs best in 13 cases out of 15. On short sequences, however, it gets an average accuracy of 73.5%, worse than 84.7% for T-Coffee and 79.2% for ClustalX, since it constructs the initial population with few alignments centered on segment profiles, ultimately resulting in premature convergence. In conclusion, for medium and long benchmarks regardless of similarity, our method performs best.

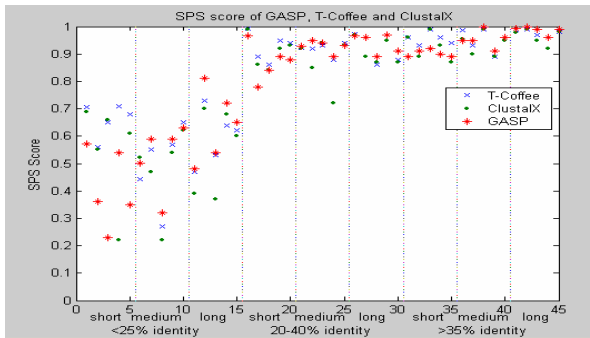


Fig. 1. The SPS scores of GASP and the other tools. Here, the five test cases are chosen from ref1 with varied percent identity and varied length of equidistant sequences.

Figure 2 shows the norMD scores of GASP and the two other programs on ref2 alignments. Neither of the BALiBASE scores measure the alignments excluded from the BALiBASE benchmarks. Here, for clearer comparison, we align only one ‘orphan’

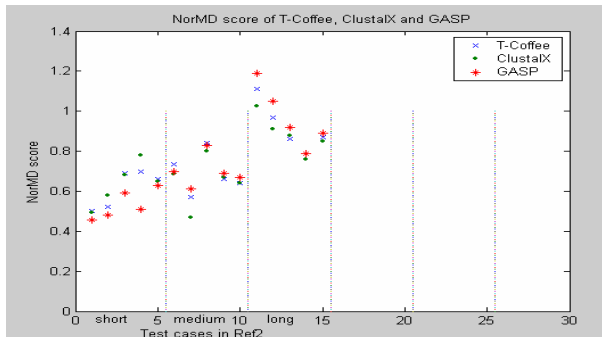


Fig. 2. The norMD scores of GASP and the other tools on test cases. Here, each of the five test sets is chosen from ref2 with varied length of sequences but only one, rather than three ‘orphan’ sequences, is aligned to a family of at least 15 closely related sequences on ref2 alignments.

sequence to a family of closely related sequences extracted from ref2 benchmarks. Figure 2 shows that, on average, GASP has comparable norMD scores to the other two programs for medium and long sequences but still obtains the worst norMD score for short sequences. However, the superiority is more pronounced for medium and long sequences, and the difference less for short ones, using the norMD measure. This means there are few gaps inserted into columns between core blocks in GASP, since norMD also considers the columns between core blocks.

Figure 3 shows one subalignment of GASP and the corresponding one for T-Coffee. Here, an ‘orphan’ sequence, *lgowA*, is aligned to a family of 15 closely related sequences. In this alignment, 8 of 16 aligned subsequences are laid out below. In GASP, an alpha helix from each sequence is aligned in common columns, for it was first found as a segment profile and then kept down. The corresponding alpha helix diverges by some gaps in T-Coffee.

```

      2myr .FESDGGDGSSNIYYYPKG IYSVMDYFKNKYNN----PLIYVTENGISTPG---:
bgl2 trirp ----PRAASIWIIYVYPYMF IQEDFEIFCYILKINITILQFSITENG MNEFNDAI
bgl2_maize ----PPMGNPWIIYMYPEGLKDLLMIMKNKYGN----PPIYITENGIGDVTKE
bgl2_bacsu -PHLITSNWDW-TIDPIGLRIGLRRI TSRYQ-----LPVFITENGLGEFDK--
lacg_staau t-VDVPRTDWDW-MIYPQGLYDQIMRVVKDY---PNYHKIYITENGLGYKDEFI
lacg_lacac PDGIETTDWDW-LIYPQGLYDKIMRVKNDY---PNIHKVYITENGLGKDTVP
lacg_lacca .PDGIETTDWDW-SIYPRGMYDILMRIHNDY---PLVPVYVTENGIGLKSPL
lgowA PTSDFG----WEFF-PEGLYDVLTKYWNRYH---L--YMYVTENGIADDA---
--PLF-ESDGGDGSSNIYYYP----KG IYSVMDYF-KNKYNN-PLIYVTENGISTP:
-----PRAASIWIIYVYPYMF IQEDFEIFCYILKINITI-LQFSITENG MNEF
-----PPMGNPWIIYMYP---EGLKDLLMIM-KNKYGN-PPIYITENGIGDV
KTKKN-PHLITSNWDW-TIDP----IGLRIGLRRI-TSRYQ---LPVFITENGLGEF:
QREFD-VDVPRTDWDW-MIYP---QGLYDQIMRV-VKDYPNYHKIYITENGLGYK:
EEKLP-DGIETTDWDW-LIYP---QGLYDKIMRV-KNDYPNIHKVYITENGLGFK:
EEKLP-DGIETTDWDW-SIYP---RGMYDILMRI-HNDYPLVPVYVTENGIGLK:
-NSVSLAGLPTSDFGW-EFFP----EGLYDVLTKY-WNRYH---LYMYVTENGIADDA:

```

Fig. 3. A subalignment of GASP and the corresponding one for T-Coffee. Here, the red region is a secondary structure (an alpha helix); the green, a beta strand.

The average running time of GASP and the other programs for long sequences is calculated in Figure 4. Here, time is measured in milliseconds and short sequences are not taken into consideration, for the alignments constructed by GASP are less accurate than those yielded by the other two. GASP is not suitable for building alignments for short sequences. In Figure 4, we conclude that ClustalX performs best for long sequences. However, ClustalX achieves this at the expense of low accuracy. Our algorithm is slower than ClustalX, but faster than T-Coffee.

Test case	GASP	ClustalX	T-Coffee
Ref1 long<25% identity	1881	752	3109
Ref1 long<20-40% identity	2263	869	4187
Ref1 long >35% identity	2902	915	4984
Ref2 long	3859	2073	6735

Fig. 4. Average running time of GASP and the other tools for long sequences

4 Conclusion and Future Work

GASP was developed to structurally align similar regions of multiple long proteins. GASP is based on a genetic algorithm, but differs from existing GA-based multiple sequence alignment methods in that it builds up the initial population by SPs. It first constructs the initial population in which the most individuals are centered on different SPs, then keeps SPs down, finally rearranges the highest-scoring individual of each new generation to avoid being trapped in local minima. The experimental results show that GASP achieves high accuracy and still maintains a competitive running time. For medium and long sequences, GASP yields the best result with appropriate parameters and the running time of GASP is comparable to that of representative tools. For short sequences, GASP can be improved by incorporating other computational methods during the iterative procedure.

References

1. Thompson, J.D., Plewniak, F.: A comprehensive comparison of multiple sequence alignment programs. *Nuc. Acids. Res.*, 1999, 27:2682–2690.
2. Thompson, J.D., Gibson, T.J.: The CLUSTAL_X windows interface: flexible strategies for MSA aided by quality analysis tools. *Nuc. Acids. Res.*, 1997, 25(24):4876-82.
3. Brudno, M., Chapman, M.: Fast and sensitive multiple alignment of large genomic sequences. *Bioinformatics*, 2003, 4:66.
4. Notredame, C., Higgins, D.G.: SAGA: sequence alignment by genetic algorithm. *Nuc. Acids. Res.*, 1996, 24:1515-1524.
5. Eddy, R.: *Biological Sequence Analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998, pp: 51-68.
6. Dayhoff, M., Schwartz, R.M.: A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 1978, 5:345–352.
7. Thompson, J.D., Plewniak, F.: Multiple Sequence Alignment Objective Function. *J. Mol. Biol.*, 2001, 314(4):937-951.
8. Benner, S.A., Cohen, M.A.: Amino acid substitution during functionally constrained divergent evolution of protein sequences. *Protein Eng.*, 1994, 7:1323–1332.
9. Shiyi, Shen., Jun, Yang.: Super Pairwise Alignment (SPA): An Efficient Approach to Global Alignment for Homologous Sequences. *J. Com. Biol.*, 2002, 9(3):477-486.
10. Thompson, J.D.: BALiBASE: A benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 1999, 15:87-88.
11. Notredame, C., Higgins, D., Heringa, J.: T-Coffee: A novel method for multiple sequence alignments. *J. Mol. Biol.* 2000, 302:205-217.

A Novel Visual Clustering Algorithm for Finding Community in Complex Network

Shuzhong Yang¹, Siwei Luo¹, and Jianyu Li²

¹ School of Computer and Information Technology, Beijing Jiaotong University,
100044, Beijing, China

² School of Computer and Software, Communication University of China,
100024, Beijing, China
yang_shu_zhong@163.com

Abstract. Complex network is an active research field in complex system in recent years. In this paper, we investigate the topological structure of complex networks and present a novel unsupervised visual clustering algorithm for finding community in complex networks. We firstly introduce a new distance between nodes to measure the dissimilarity between nodes and obtain the distance matrix. Then the rows (columns) of distance matrix are reordered according to the dissimilarity and the reordered matrix is displayed as an intensity image. Clusters are indicated by dark blocks of pixels along the main diagonal. The experiments show that our algorithm has good performance and can find the community structure hidden in complex networks.

1 Introduction

Clustering problems arise in many different applications, such as data mining and knowledge discovery [1], data compression and vector quantization [2], and pattern recognition and pattern classification [3] in the last decades. The general problem of clustering can be described very simply: we wish to partition a data set into a number of groups, or clusters, according to some criterion of similarity or dissimilarity such that the data within a cluster have high similarity comparison to one another, but are very dissimilar to data in other clusters. The criterion determines whether two data are similar or not. Similarity or dissimilarity is only an opposite concept and in general one is the inverse of the other. Various clustering algorithms [12, 13, etc] have been researched in the last decades which used different criteria of similarity or dissimilarity such as Euclidean distance. In this paper, the clustering problem of graph will be discussed. The difference between the data clustering and graph clustering is that there does not exist general meaningful distance in graph. So it is an important problem to define a meaningful distance in order to realize the clustering of graph.

There is a growing interest in evolving complex networks in recent years. Complex networks exhibit many different properties such as small world effect [4], scale-free effect [5] and community structure [6] compared to regular and random networks. Many real networks emerge the above common properties of complex networks. Since the real networks are continuously evolutionary, the researches to properties of network and prediction of network evolution are very important. To

predict the tendency of network evolution we must firstly build the complex network models to simulate the real networks. The most earliest and prominent small world model was presented by Watts and Strogatz [4] in 1998 according to random cut and rewiring of regular network. In 1999, Barabasi and Albert presented the first scale-free model according to growth and preferential attachment. After above two prominent works, various complex network models [9, 10, 11] were presented. Having networks, we also need design algorithms to find out the properties hidden in networks. Visual clustering of complex networks is an effective way. Visual clustering of complex networks can help us find the community structure hidden in the networks, understand the networks better and predict the behavior of networks in future. In this paper, a novel visual clustering algorithm for finding community in complex network will be presented to realize the above aims.

The rest of the paper is organized as follows. In Section 2, we describe the related works about visual clustering and why we use the defined distance and visualization technique. In Section 3, we give the detailed visual clustering procedure and method. In Section 4, some performance experiments are presented. Finally, discussions are given in Section 5.

2 Related Works

The aim of visual clustering is not only to realize the unsupervised classification of data but also to display the clustering result to the researchers. The general clustering algorithms [12, 13] only realize the clustering of data and denote a cluster by its clustering center. As we know, the initialization of clustering centers can influence the quality of clustering and a clustering algorithm can also be regarded as the initialization of another one [14]. So in the literature [7], the authors presented a tool for visual assessment of cluster tendency (VAT). Euclidean distance is used to describe the similarity (dissimilarity) between the data. The tool simply reorders the distance matrix according to the similarity such that the more similar the two data is, the nearer their indexes is and the cluster tendency indicated by dark blocks of pixels along the main diagonal. The tool is not only a good visual clustering method without knowing the number of clusters in advances but also can serve for another clustering algorithm as its initialization procedure. Furthermore, VAT tool has much lower time complexity than clustering algorithms since the general clustering algorithms need compute pairwise distances many times, but VAT tool only need compute pairwise distances once.

Different from [7], the clustering problem of graph instead of data will be discussed in this paper. The difference between the data clustering and graph clustering is that there does not exist general meaningful distance. So the definition of distance and clustering algorithm are both important issues to graph visual clustering. In the literature [8], the authors introduced a new method to realize the multi-scale visualization of small world networks. It firstly described a metric that had been designed in order to identify the weakest edges in a small world network. The metric leads to an easy and low cost filtering procedure that breaks up a graph into smaller and highly connected components. We called the metric used in that paper “edge strength”. We find that the edge strength is related to the distance between nodes. The stronger the edge

strength between nodes is, the smaller the distance between them is. Suggested by this, a transform of edge strength is used to define the distance between nodes. In the next section, the detailed definition of distance and visual clustering procedure will be described.

3 Visual Clustering Algorithm of Complex Network

In previous section, we described the technique we will use to realize visual clustering of complex network. In this section, we will give the detailed procedure which mainly consists of three steps: the quantification of dissimilarity, reordering the dissimilarity matrix and display the reordered matrix as an intensity image. Then clusters are indicated by dark blocks of pixels along the main diagonal.

3.1 The Quantification of Dissimilarity

A graph $G = (V, E)$ generally consists of a finite set V of nodes and a finite set E of edges with $E \in V^{(2)}$, where $V^{(2)}$ is the set of all subsets of V which have exactly two elements.

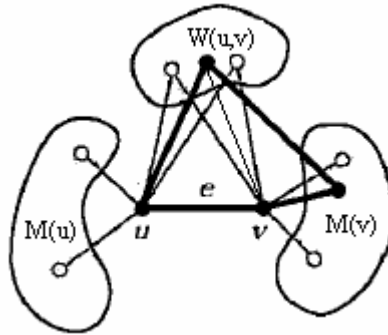


Fig. 1. Dividing neighbors of edge (u, v) [8]

According to the above graph definition and edge strength definition in [8], the pairwise distances between nodes can be computed through the following steps. Given an edge $(u, v) \in E$ (Fig. 1), we can compute its strength by dividing neighbors of u or v into three distinct subsets. Firstly, denote by $M(u)$ the set of neighbors of u that are not neighbors of v (excluding v). Secondly, denote by $M(v)$ the set of neighbors of v that are not neighbors of u (excluding u). Finally, denote by $W(u, v)$ the set of common neighbors to u and v . Denote by $r(A, B)$ the number of edges linking nodes in the set A to nodes in the set B such that $s(A, B) = r(A, B) / |A||B|$ ($|A|$ is the number of nodes in A) defines the proportion of existing edges among all possible edges connecting nodes of A and B . By above definition, any edge connecting two of the subsets $M(u)$, $M(v)$ or $W(u, v)$ is part of a cycle of length 4 passing through the edge (u, v) (see Fig. 1, the cycle labeled by thick lines). By this way, all cycles of length 4 are

captured. Finally, we define the ratio $|W(u, v)|/(|M(u)|+|W(u, v)|+|M(v)|)$ as a ratio related to the proportion of cycles of length 3 containing the edge (u, v) (see Fig. 1, the triangles labeled by thin lines). The number of cycles equals to the number of nodes in $W(u, v)$. Then we can compute $strength(u, v)$ using the following equation:

$$strength(u, v) = \frac{s(M(u), W(u, v)) + s(W(u, v), M(v)) + s(W(u, v)) + s(M(u), M(v)) + |W(u, v)|/(|M(u)|+|W(u, v)|+|M(v)|)}{(1)} \tag{1}$$

(Set $s(A) = 2r(A,A)/(|A|(|A|-1))$ when computing the proportion of edges connecting a set to itself).

Then we can use the inverse of edge strength to define the pairwise distances between nodes. If the strength between two nodes is zero, the distance between them is infinity; otherwise the distance equals to $D_{uv} = 1 / strength(u, v)$.

3.2 Reordering the Dissimilarity Matrix

According to (1) we can obtain the dissimilarity matrix given the adjacent matrix of complex network. Denote by \mathbf{D} the dissimilarity matrix and then we can see that \mathbf{D} satisfies the following conditions for all $1 \leq i, j \leq n$ (n is the dimension of \mathbf{D}):

$$D_{ij} \geq 0, \quad D_{ij} = D_{ji}, \quad D_{ii} = 0 \tag{2}$$

Next we present two strategies called ‘‘SetNearest’’ and ‘‘PointNearest’’ to reorder the dissimilarity matrix. The ‘‘PointNearest’’ strategy is similar to that used in [7]. In fact, the procedure of reordering is the procedure of permutation of indexes so that the smaller the dissimilarity between two nodes is, the nearer the indexes of two nodes are.

The reordering procedure is as follows which is similar to the algorithm in [7]:

- 1) Set $R = \{1, 2, \dots, n\}; S = T = \emptyset; P = (0, \dots, 0)_n$.
- 2) Select $(i, j) \in \arg \max_{p \in R, q \in R} \{D_{pq}\}$. Set $P(1)=i; S = \{i\}$; and $T = R - \{i\}$.
- 3) For $k=2, \dots, n$
 Select $(i, j) \in \arg \min_{p \in S, q \in T} \{D_{pq}\}$. Set $P(k)=j; S = S \cup \{j\}$ and $T=T - \{j\}$.
 End for.
- 4) Obtain the ordered dissimilarity matrix $\tilde{\mathbf{D}}$ using the ordering array P as:
 $\tilde{D}_{ij} = D_{P(i)P(j)}, 1 \leq i, j \leq n$.

In step 2) the strategy we use is ‘‘SetNearest’’ in which we select an index from T which has smallest dissimilarity to all indexes of S . In the following experiments, we will also use the strategy ‘‘PointNearest’’ in which we select an index from T which has smallest dissimilarity to the last element of S .

3.3 Display the Reordered Matrix as Intensity Image

After obtaining the reordered dissimilarity matrix, we can display it as an intensity image, which we call a dissimilarity image. The intensity or gray level g_{ij} of pixel

(i, j) depends on the value of D_{ij} . The value $D_{ij} = 0$ corresponds to $g_{ij} = 0$ (pure black); the value $D_{ij} = D_{\max}$, where D_{\max} denotes the largest dissimilarity value in \mathbf{D} , corresponds to $g_{ij} = 255$ (pure white). Intermediate intensity value g_{ij} produced by intermediate values of D_{ij} can be computed as

$$g_{ij} = 255 * D_{ij} / D_{\max} \quad (3)$$

After this, we obtain the intensity image $\mathbf{G} = \{g_{ij}\}$ which will often indicate cluster tendency in the data by dark blocks of pixels along the main diagonal. The ordering is accomplished only by processing elements in the dissimilarity matrix \mathbf{D} rather than using the data directly, so it has much lower time complexity than other algorithms.

4 Experimental Analysis

Since no former similar algorithm is used to implement the visual clustering of complex networks, we can't give the direct comparisons between the formers and ours. But in order to validate the performance of our method, we will design four experiments and discuss the experimental results. The complex networks we used include ER network, WS small world network [4], BA scale-free network [5] and the structured network [11] with small world effect, scale-free effect and community structure. ER network has small clustering coefficient and small characteristic path length. WS small world network has big clustering coefficient and small characteristic path length compared to ER network. BA scale-free network has small clustering coefficient and small characteristic path length but has power-law degree distribution of nodes. These three networks all have no distinct community structure. The structured network has small clustering coefficient, small characteristic path length and power-law degree distribution of nodes. When reordering the quantified dissimilarity matrix, we use two strategies called "SetNearest" and "PointNearest", respectively. The four networks are listed in table 1 and Fig. 2, Fig. 3, Fig. 4, Fig. 5 are experimental results, respectively.

Table 1. The four types of simulated complex work

type of complex network	the number of nodes	the number of clusters
ER network	100	not defined
WS small world network	200	not defined
BA scale-free network	100	not defined
The structured network	106	4

From Fig. 2 we can see that nearly every node belongs to a different cluster in ER networks. It is right as we know that every node connects to others with equal probability in random ER networks, so it can't form a cluster with others. From Fig. 3 we can see that although only few edges are rewired in small world network, our method can cluster the nodes according to our new distance. It is consistent with what we have known that small world network has clustering property and has big clustering

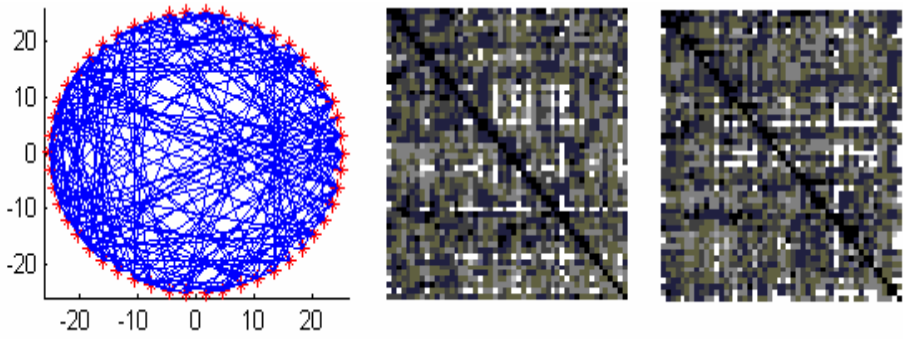


Fig. 2. Visual Clustering of ER random network, left: original network; middle: result using “PointNearest”; right: result using “SetNearest”

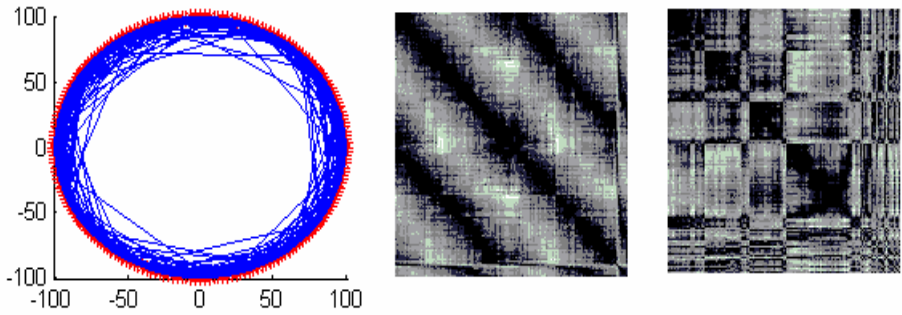


Fig. 3. Visual Clustering of WS small world network, left: original network; middle: result using “PointNearest”; right: result using “SetNearest”

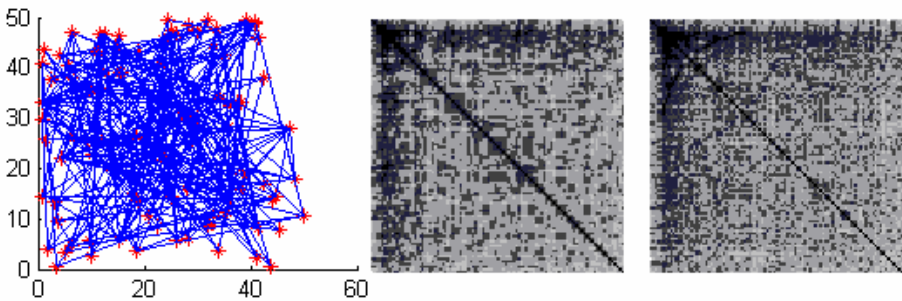


Fig. 4. Visual Clustering of BA scale-free network, left: original network; middle: result using “PointNearest”; right: result using “SetNearest”

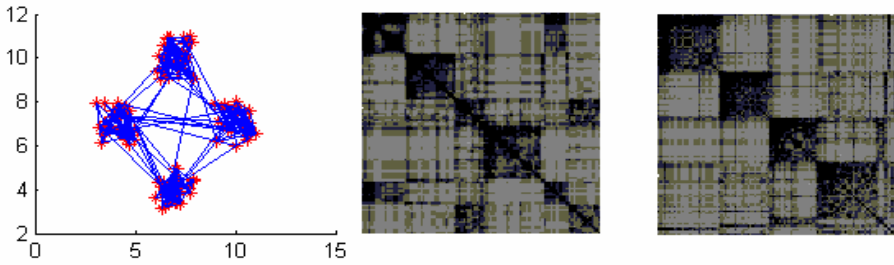


Fig. 5. Visual Clustering of structured network, left: original network; middle: result using “PointNearest”; right: result using “SetNearest”

coefficient. From Fig. 4 we can see that the nodes in scale free network don’t have clustering property, too. This is consistent with what we have known that scale free network has small clustering coefficient and has no clustering property. From Fig. 5 we can see that our method has good performance on structured network. It is also noted that our algorithm can find out hierarchical structure hidden in complex networks (Fig. 3 and Fig .5). In a word we can conclude that as long as the complex networks have clustering property, our visual clustering algorithm can find out the community hidden in them. Furthermore we can see that the strategy “SetNearest” has better performance than the strategy “PointNearest” in reordering the dissimilarity distance from experiments.

5 Conclusions

In this paper, we present a novel unsupervised visual clustering algorithm for finding community in complex networks. Our main contribution includes two aspects: Firstly, we introduce a new measurement to measure the dissimilarity between nodes and secondly we use the method of reordering the dissimilarity matrix to realize the visual clustering of the networks. Our method has low time complexity $O(N^2)$. The experiments prove that as long as the networks have community structure, our method can find out it correctly. Furthermore our method can also be used as the preprocess procedure of other clustering methods to decide the number of clusters. Certainly the data sets which we used in experiments are all synthetic, so the massive real world data set need be used to validate our visual clustering algorithm in future.

Acknowledgements

The research is supported by the National Natural Science Foundation of China under Grant Nos. 60373029 and the National Research Foundation for the Doctoral Program of Higher Education of China under Grant Nos. 20050004001.

References

1. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy.: *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.
2. A. Gersho and R.M. Gray.: *Vector Quantization and Signal Compression*. Boston: Kluwer Academic, 1992.
3. R.O. Duda and P.E. Hart.: *Pattern Classification and Scene Analysis*. New York: John Wiley & Sons, 1973.
4. Watts, D., and Strogatz, S.: Collective dynamics of 'small-world' networks. *Nature* 393 (1998) 440-442.
5. Albert-Laszlo Barabasi, Reka Albert.: Emergence of scaling in random networks, *Science* 286 (1999) 509-512.
6. MEJ Newman.: Detecting Community Structure in Networks. *Eur. Phys. J. B* 38, (2004) 321-330.
7. Bezdek, JC, and RJ Hathaway.: "VAT: A Tool for Visual Assessment of (Cluster) Tendency," *Proc. IJCNN 2002*, IEEE Press, Piscataway, NJ, (2002) 2225-2230.
8. Auber, D., Chiricota, Y., Jourdan, F., and Melancon, G.: Multiscale visualization of small world networks. In *Proceedings of the 2003 IEEE Symposium on Information Visualization*. (2003) 75-81.
9. Holme P. and Kim B.J. Growing scale-free networks with tunable clustering. *Phys. Rev. E* 65, 026107 (2001).
10. K.Klemm and V.M.Eguíluz.: Growing Scale-Free Networks with Small World Behavior. *Phys. Rev. E* 65, 057102 (2002).
11. Chunguang Li, Philip K. Maini.: An Evolving Network Model with Community Structure. *Journal of Physics A: Mathematical and General*, Vol. 38, No. 45, (2005) 9741-9749.
12. Kanungo et al.: An Efficient k-Means Clustering Algorithm: Analysis and Implementation. *PAMI* (24), No. 7, (2002) 881-892.
13. Yu, J. [Jian]: General C-Means Clustering Model. *PAMI* (27), No. 8, (2005) 1197-1211.
14. P.S.Bradley, Usama M.Fayyad, Refining Initial Points for K-Means Clustering. *Proceedings of the Fifteenth International Conference on Machine Learning*, (1998) 91-99.

Self-Organizing Network Evolving Model for Mining Network Community Structure

Bo Yang

College of Computer Science and Technology & Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, 130012, P.R. China
ybo@jlu.edu.cn

Abstract. Community structure is an important topological property of network. Being able to discover it can provide invaluable help in exploiting and understanding complex networks. Although many algorithms have been developed to complete this task, they all have advantages and limitations. So the issue of how to detect communities in networks quickly and correctly remains an open challenge. Distinct from the existing works, this paper studies the community structure from the view of network evolution and presents a self-organizing network evolving algorithm for mining communities hidden in complex networks. Compared with the existing algorithm, our approach has three distinct features. First, it has a good classification capability and especially works well with the networks without well-defined community structures. Second, it requires no prior knowledge and is insensitive to the build-in parameters. Finally, it is suitable for not only positive networks but also signed networks containing both positive and negative weights.

1 Introduction

Many systems in the world take the form of networks^[2,3] such as human societies^[4,5], natural ecosystems^[6,7,8] and technological systems^[9,10]. The networks that include only positive weights is called positive networks, and the networks with both positive and negative weights are called signed networks^[1,14]. Although different networks have distinct structures, they share some common statistical properties including small world effect^[3], network transitivity^[3,11] and power-law distribution of degree^[2,12]. In this paper, we will focus on one of topological properties, community structure, which is shared by many networks.

A network community refers to a group of vertices with similar link-based properties such as link density and link sign. In positive networks, communities are decided by link density, which are defined as the groups of vertices within which the links are dense but between which they are sparse^[13], as illustrated in the left pane of Fig.1. While in signed networks, communities are decided not only by link density but also by link sign, which are defined as the groups of vertices, within which positive links are dense and negative links are sparse, while between which negative links are dense and positive links are sparse, as illustrated in the right pane of Fig.1.

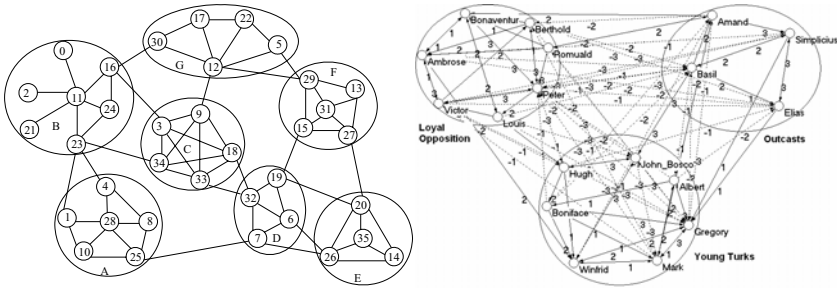


Fig. 1. Two schematic representations of network communities. Left pane shows a simple positive network containing 7 communities denoted by A to G respectively. The links in each community are dense, while the links between them are sparse. Right pane shows a signed social network built by Sampson in 1969^[15], which describes the friendship relations of a group of monks in New England monastery. Solid lines denote the positive links (like relations) and dashed lines denote negative links (dislike relations). Three communities, Young Turks, Outcasts and Loyal Opposition, are detected. Within community the positive links are dense while the negative links are sparse. On the contrary, between communities the positive links are sparse while the negative links are dense.

Different kinds of networks may have different kinds of communities. Communities in social networks can be real social groups within which people share common interests and have more contacts than they do without; communities in natural ecosystem networks can represent types of fauna, flora or other functional groupings; communities on the WWW may consist of collections of web pages related to common topics. The ability to mine community structure hidden inside networks can help us effectively exploit and sufficiently understand such networks. For instance, in the case of WWW, the ability to find web communities is helpful for content filtering and automatic classification, which can improve search engines significantly^[16].

Many algorithms to solve this issue have been developed, which can be generally divided into three main categories: (1) bisection methods, mainly including spectral methods^[17–20], the Kernighan-Lin algorithm^[21] and the Wu-Huberman algorithm^[22]; (2) hierarchical methods, including agglomerative methods based on some similarity measure^[4,23,24] and divisive methods such as the GN algorithm^[25], the Tyler algorithm^[26] and the Radicchi algorithm^[27]; (3) methods for detecting web communities mainly including the MFC algorithm^[16], the HITS algorithm^[28,29], the SAE algorithm^[30] as well as others^[31,32].

However the issue of discovering communities has not been solved satisfactorily due to three main reasons. First, the clustering accuracy requires being improved further. Especially, most of the existing algorithms have a low clustering accuracy when they deal with the networks without well-defined community structures. Second, most of the existing algorithms require prior knowledge such as the number of communities, the approximate size of each community or a predefined criterion for cutting dendrogram, but all of which are hard to be obtained beforehand. Finally, most existing algorithms are exclusively designed for positive networks, and thus they are not suitable for signed networks. This paper aims to address these problems.

The remainder of the paper is organized as follows: section 2 presents a self-organizing network evolving algorithm for mining communities in social networks. In section 3 we test this algorithm and evaluate its performance against some benchmark networks. Section 4 concludes the paper by highlighting the major advantages of our work and some future extensions.

2 Self-Organizing Network Evolving Model

2.1 Basic Idea

The Self-Organizing Network Evolving Model (SONE model) can be described as a two-tuple (N, O), where N denotes the network in question, and O denotes the set of evolutionary operators. The network in the SONE model is considered as a dynamic system containing a group of vertices with a set of changeable relations among them. Like other dynamic systems such as human societies and ecosystems, the network N can evolve continuously, from its original state to its convergent state, regulated by some predefined operators. Inspired by the friendship network of human society in which each individual prefers choosing new friends from his old friends' friends, the evolutionary operators of the SONE model are defined as $O=\{o_1,o_2,o_3,o_4\}$, where o_1 is the *similarity computing operator*, which computes the similarity between each adjacent pair of vertices of N; o_2 is the *neighbors selecting operator*, which selects the nearest neighbors and deleting rest ones for each vertex of N; o_3 is the *neighbors making operator*, which produces new neighbors for each vertex of N; and o_4 is the *balancing operator*, which averages the similarity of each pair of adjacent vertices.

Based on these four evolutionary operators, the evolving process of the network N can be described as an iterative course, in which four evolutionary operators, as the sequence of o_1, o_2, o_3 and o_4 , are applied to N repeatedly until N get into its convergent state. This evolving process is actually a positive feedback course, in which the links between communities will gradually disappear, while those within communities will gradually emerge. Finally, the original network N will become the one being made up of a set of separated cliques, and each of them corresponds to a community of N.

2.2 Evolutionary Operators

2.2.1 Similarity Computing Operator

In an unigned, unweighted and undirected network, the similarity of vertex i and vertex j can be defined as:

$$s_{ij} = \begin{cases} 0 & (i, j) \notin L \\ \frac{|K(i) \cap K(j)|}{|K(i)|} \cdot \frac{|K(i) \cap K(j)|}{|K(j)|} & (i, j) \in L \end{cases} \quad (2.1)$$

where $K(i)$ is the set of neighbors of vertex i .

The idea behind this definition is that two individuals sharing more acquaintances are more likely in the same community. s_{ij} is actually the probability of two individuals belonging to the same community. The higher the value of s_{ij} the denser the links

appear among vertex i , vertex j and their respective neighbors, and thus the more likely these vertices belong to the same community due to their highly clustered relations. We can extend the Equation 2.1 for getting a more general similarity measure suitable for all kinds of networks including signed, weighted and directed networks.

Definition 1. Let A be the adjacency matrix of a network, the clustering similarity of vertices i and j is defined as:

$$S(i, j) = \begin{cases} 0 & A(i, j) \leq 0 \\ \frac{\sum_{k \in \Gamma_i^+(A) \cap \Gamma_j^+(A)} A(i, k) \cdot \sum_{k \in \Gamma_i^+(A) \cap \Gamma_j^+(A)} A(j, k)}{\sum_{k \in \Gamma_i^+(A) \cup \Gamma_j^-(A)} |A(i, k)| \cdot \sum_{k \in \Gamma_j^+(A) \cup \Gamma_i^-(A)} |A(j, k)|} & A(i, j) > 0 \end{cases} \quad (2.2)$$

where $K_i^+(A) = \{k \mid A(i, k) > 0\}$, $K_i^-(A) = \{k \mid A(i, k) < 0\}$, $\Gamma_i^+(A) = \{i\} \cup K_i^+(A)$, $\Gamma_i^-(A) = \{i\} \cup K_i^-(A)$.

S is called the *clustering similarity matrix* of a network, and the *similarity computing operator* is defined as:

$$o_1 : A \rightarrow S \quad (2.3)$$

The time complexity of similarity computing operator is $O(m^2/n)$, where m and n are the numbers of links and vertices respectively. Let $d = m/n$ is the average degree of a network. The total time of figuring out the Equation 2.2 is $O(4d+1) = O(d)$. Because the clustering similarity of non-adjacent vertices is always zero, and thus at most m pairs of adjacent vertices should be considered. So the entire clustering similarity matrix can be figured out within the time of $O(md) = O(m^2/n)$.

Euclidean distance^[23,24] and *correlation coefficient*^[23] are two commonly used similarity measures in bottom-up clustering methods. Actually, the concepts of *edge betweenness* presented in GN algorithm^[25] and *edge clustering coefficient* presented in Radicchi algorithm^[27] also are two kinds of similarity measures used in top-down clustering methods, which are respectively defined as the number of geodesic paths running through a given edge and the number of triangles or squares which a edge belongs to. Compared with them the clustering similarity presented here has two distinct features. First, it is a more general similarity measure suitable for both positive networks and signed networks; second, it can be figured out locally, that is, this similarity between vertices only depends on the information of their respective neighbors and is unrelated to the rest. This feature is especially significant for designing distributed algorithm in our future work.

2.2.2 Neighbors Selecting Operator

Neighbors selecting operator is defined as follows:

$$o_2 : S \rightarrow S^{(2)} \quad (2.4)$$

where $S^{(2)}$ is defined as:

$$S^{(2)}(i, j) = \begin{cases} S(i, j) & S(i, j) > f_i(S) \\ 0 & \text{else} \end{cases} \quad (2.5)$$

where the *threshold function* $f_i(S)$ is defined as follows:

$$f_i(S) = \omega_1(\mu_i(S) + \omega_2\sigma_i(S)) \quad (2.6)$$

where ω_1 and ω_2 are two constants, $\mu_i(S)$ and $\sigma_i^2(S)$ are respectively defined as follows:

$$\mu_i(S) = \frac{1}{|K_i^+(S)|} \sum_{k \in K_i^+(S)} S(i, k), \quad \sigma_i^2(S) = \frac{1}{|K_i^+(S)|} \sum_{k \in K_i^+(S)} (S(i, k) - \mu_i(S))^2$$

The time complexity of neighbors selecting operator is $O(m)$. Again let $d = m/n$ is the average degree of a network. μ and σ^2 can be computed within the time of $O(d)$ respectively. So, for each vertex i , all its $S^{(2)}(i, j)$ can be figured out within the time of $O(3d) = O(d)$. So, entire matrix $S^{(2)}$ can be figured out within the time of $O(nd) = O(m)$.

2.2.3 Neighbors Making Operator

This operator is inspired by the friendship network of human society in which each individual prefer choosing new friends from his old friends' friends. *Neighbors making operator* is defined as follows:

$$o_3 : S^{(2)} \rightarrow S^{(3)} \tag{2.7}$$

where $S^{(3)}$ can be figured out by the procedure described in Table 1, in which the procedure $rank(K)$ randomly selects one element from the set K .

Table 1. Neighbors making operator o_3

1.	$S^{(3)} \leftarrow S^{(2)}$;
2.	for $i = 1 : n$ do
3.	$j \leftarrow rand(K_j^+(S^{(2)}))$
4.	for $\forall k(k \in K_j^+(S^{(2)}) \wedge k \neq i)$ do
5.	$S^{(3)}(i, k) \leftarrow \max(S^{(3)}(i, k), S^{(2)}(i, j) \cdot S^{(2)}(j, k))$
6.	end
7.	for $\forall k(k \in K_i^+(S^{(3)}))$ do
8.	if $S^{(3)}(i, k) < f_i(S^{(3)})$ then $S^{(3)}(i, k) \leftarrow 0$
9.	end
10.	end

In fact, the course of repeatedly applying operators of o_2 and o_3 to networks is a positive feedback process in which the links between communities gradually decrease, but those within communities gradually increase. Let d is the average degree of a network. Step1 takes $O(m)$ time to copy at most m nonzero elements from $S^{(2)}$ to $S^{(3)}$; step3 takes $O(1)$ time; steps of 4,5 and 6 take $O(d+1)$ time to figure out the i -th raw of $S^{(3)}$; steps of 7,8 and 9 take $O(d)$ time to select the new neighbors of vertex i ; so the total time of steps 2-10 is $O(n(2d+1)) = O(nd)$. So the time complexity of neighbors making operator is $O(nd+m) = O(m)$.

2.2.4 Balancing Operator

This operator is defined as follows:

$$o_4 : S^{(3)} \rightarrow S^{(4)} \tag{2.8}$$

where $s^{(4)}$ is defined as:

$$S^{(4)}(i, j) = \frac{1}{2}(S^{(3)}(i, j) + S^{(3)}(j, i)) \tag{2.9}$$

After applying this operator, the matrix $S^{(4)}$ become symmetry. Obviously, this operator can be implemented within a time of $O(n^2)$.

2.3 SONE Algorithm

With the evolutionary operators introduced in previous section, the self-organizing network evolving algorithm for mining network communities is presented in Table 2.

Table 2. SONE algorithm

Algorithm SONE(<i>A, S</i>)
<i>A</i> : Input, the matrix of the initial network to be mined
<i>S</i> : Output, the matrix of the finally evolved network
1. $t \leftarrow 0$;
2. $S \leftarrow o_4(o_3(o_2(o_1(A))))$;
3. if $S = A$ or $t > T$
4. return ;
5. else
6. $A \leftarrow S$; $t \leftarrow t + 1$; goto 2;

It is easy to see the time complexity of the SONE algorithm is $O(T \times (m^2/n + m + m + n^2)) = O(\max\{m^2/n, n^2\})$. Let k is the number of communities in network, and n/k is the average number of vertices in each community. At the finally evolved network, there are k separated cliques, in which each vertex has links to all the others, thus we have $m = O(n^2/k)$. So, the time complexity of the SONE algorithm is $O(\max\{n^3/k^2, n^2\})$, or $O(n^2)$ at the best case and $O(n^3)$ at the worst case.

3 Evaluation on the SONE Model

In this section we will test the performance of the SONE algorithm against some benchmark social networks.

Fig.2(a) shows the karate club network. In the 1970s, Wayne Zachary observed the social interactions between members of a karate club at an American university^[5]. He then constructed the karate club network based on its members' social interactions within the club. During the two years of his study, by chance a dispute arose between the club's administrator and its principal karate teacher, and as a result the club eventually split into two roughly equal-sized clubs. One was led by its administrator, represented by squares, and the other by its teacher represented by circles, as shown in

Fig.2(a). In this experiment, we set $\omega_1 = 0.2$ and $\omega_2 = 0.2$, and the evolving process converges after six iterative steps. Fig.2(b) presents the evolved karate club network separated into two cliques. Compared with the actual division shown in Fig.2(a), only vertex 10 is misclassified.

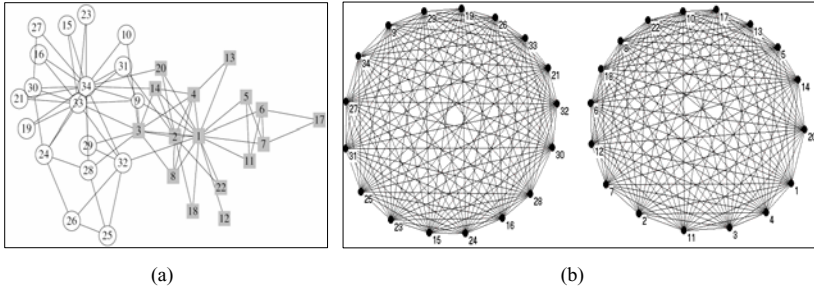


Fig. 2. Mining communities from the karate club network

As another test, we turn to the network of US college football association in 2000 season^[25]. The network includes 115 nodes and 616 edges, which represent football teams and games among those teams respectively. All of 115 teams are divided into 12 conferences. Games are more frequent between members of the same conference than between members of different conference. So each conference can be naturally considered as one community of the network. Fig.3(a) is the adjacency matrix of the initial foot association network. In this experiment, we set $\omega_1 = 0.3$ and $\omega_2 = 0.2$, and the evolving process converges after seven iterative steps. Fig.3(b) shows the adjacency matrix of the finally evolved network, in which 12 cliques are formed, and each of them corresponds to one conference respectively. Compared with real communities, most associations are detected correctly except for a few teams, such as five teams of IA Independents (No.5), teams 28 and 58 of Western Athletic (No.11), and team 110 of Texas Christian (No.4). The reason for misclassification is that these teams play more matches with teams in other associations than with those in their own association.

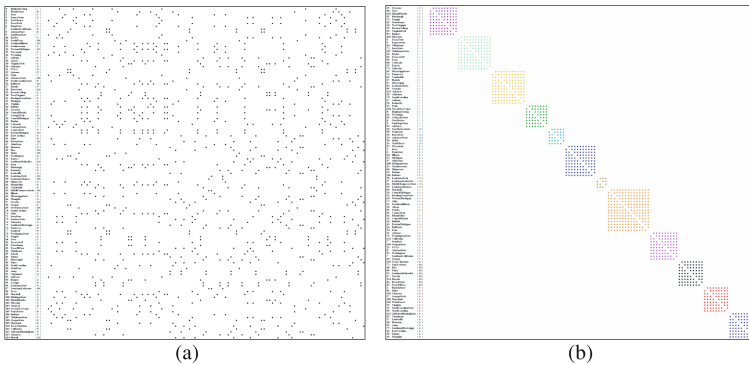


Fig. 3. Mining communities from the football association network

Fig.4(a) shows the dolphin network, which described the social relationship of 62 bottlenose dolphins living in Doubtful Sound of New Zealand. This network was established by D.Lusseau^[36] based on his observations of these dolphins for seven years. During his studies, he found these dolphins once separated into two groups due to some reasons. Fig.4(b) presents the finally evolved dolphin network directed by the SONE algorithm, in which two groups A and group B are correctly divided. Furthermore, four latent subdivisions of group B are also predicted by the SONE algorithm. We set $\omega_1 = 0.25$ and $\omega_2 = 0.2$, and the evolving process converges after nine iterative steps.

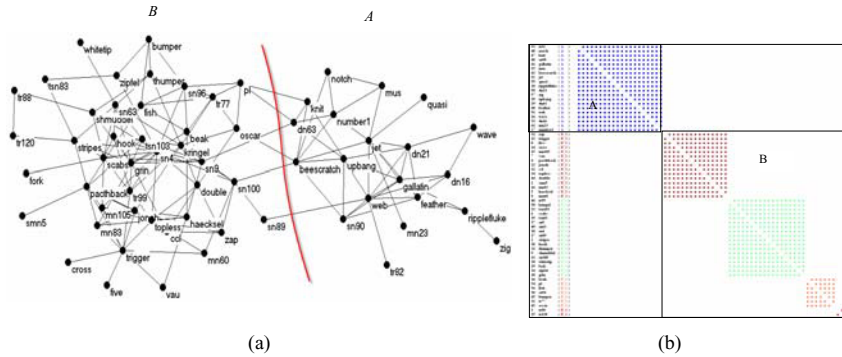


Fig. 4. Mining communities from the dolphin network

In 1969 Sampson studied social relationship among a group of monks in a New England monastery. In order to rank the friendship relations among the monks, Sampson required each monk to give only his top three choices for both like and dislike ones, and built the Sampson network as shown in the right pane of Fig.1. Based on the network, Sampson predicted that these 18 monks will be divided into 3 groups of Young Turks, Loyal Opposition and Outcasts. During his stay, a political "crisis in the cloister" resulted in a split of the monks, which was fairly close to Sampson's prediction. Fig.5 presents the finally evolved Sampson network directed by the SONE algorithm, in which three communities, Young Turks, Outcasts and Loyal Opposition, are correctly detected. In this experiment we also set $\omega_1 = 0.3$ and $\omega_2 = 0.2$, and the evolving process converges after five iterative steps.

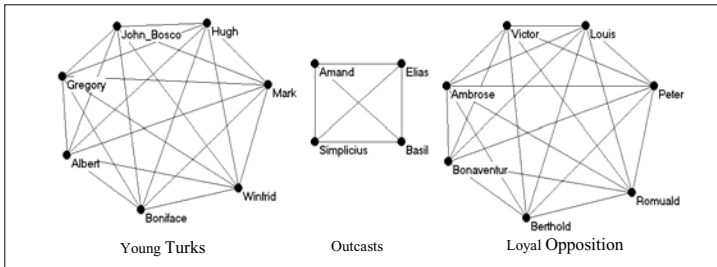


Fig. 5. Mining communities from the Sampson network

Fig.6 presents the comparison of classification accuracy among three algorithms including GN algorithm^[25], Newman algorithm^[34] and SONE algorithm. Experiment network is a benchmark network used by many papers^[25,27,34], which contains 4 communities. In the network, each vertex has z_{in} edges connecting it to members of the same group and z_{out} edges to members of other groups, with the sum $z_{in} + z_{out} = 16$. Algorithm is considered to be successful if each vertex is classified in the right community, and the four communities are not further subdivided. In Fig.6, y-axis denotes the fraction of vertices correctly identified by these three algorithms, and each point in the curves is obtained by running corresponding algorithm over 100 graphs. In this experiment we set $\omega_1 = 0.25$ and $\omega_2 = 0.2$.

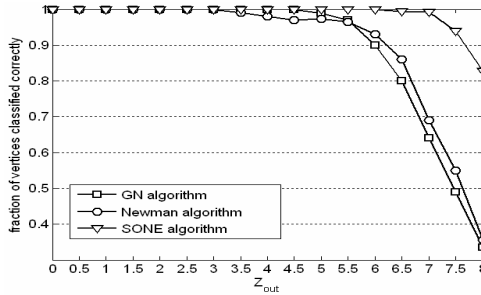


Fig. 6. Classification accuracy of three algorithms

From Fig.6 we can see that all algorithms work very well when $z_{out} \leq 5.5$, correctly identifying more than 95% of vertices. In the case of $6 \leq z_{out} \leq 8$ the classification accuracy of the SONE algorithm is much better than GN algorithm and Newman algorithm.

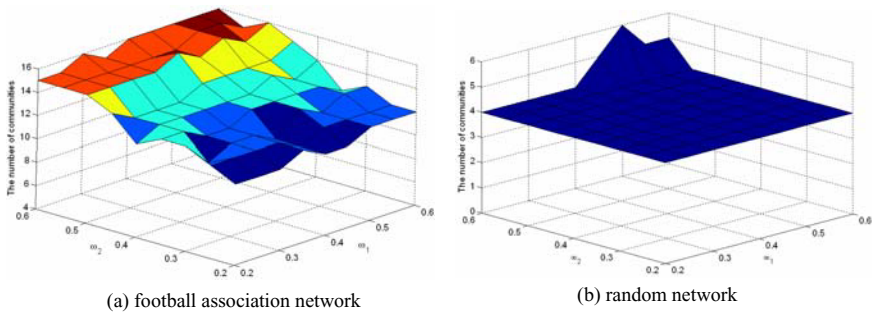


Fig. 7. The number of communities obtained by the SONE algorithm with different parameters

Only two parameters, ω_1 and ω_2 in Equation 2.6, are involved in the SONE model, which are used to control the granularity of communities. More communities with smaller size will be obtained under larger value of ω_1 and ω_2 . Otherwise, fewer

communities with larger size will be obtained. Fig.7 illustrates this fact using two networks, in which the z -axis denotes the number of communities obtained under different values of two parameters. From this figure, we can see that the number of communities obtained by the SONE algorithm is related to but not sensitive to parameters when $0.2 \leq \omega_1, \omega_2 \leq 0.6$. In Fig.7(a), the number of communities obtained in football association network falls into the interval [11,16], and its average is 13, very close to the actual number of football teams. In Fig.7(b), the number of communities obtained in a random network with four communities varies within the interval [3,6], and its average is 4.

The concept of *modularity* presented by Newman^[35] is used to evaluate the quality of a particular division of a network. Consider a particular division of a network into k communities. The modularity Q of this division is defined as follows:

$$Q = \sum_i (e_{ii} - a_i^2) \tag{3.1}$$

where e is a $k \times k$ symmetric matrix whose element e_{ij} is the fraction of all edges in the network that link vertices in community i to vertices in community j , and $a_i = \sum_j e_{ij}$.

As argued by Newman, higher modularity value indicates better division for a network.

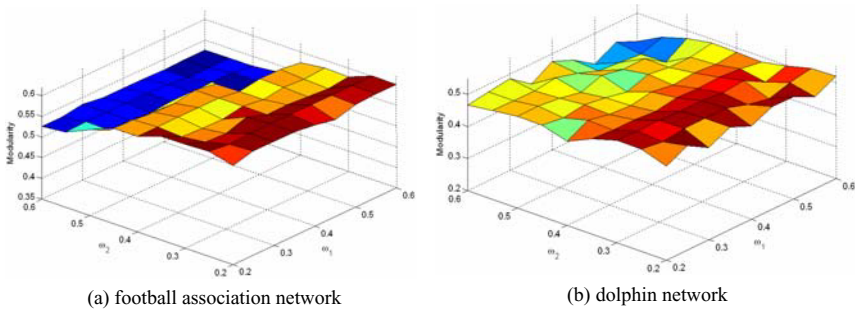


Fig. 8. The modularity obtained by the SONE algorithm with different parameters

From Fig.8 in which the z -axis denotes the Q -value of the divisions obtained under different parameters, we can see that the modularity obtained by the SONE algorithm is related to but not sensitive to parameters when $0.2 \leq \omega_1, \omega_2 \leq 0.6$. In Fig.8(a), the obtained modularity of the football association network falls into the interval [0.52, 0.603], and gets 0.601 at $\omega_1 = 0.3$ and $\omega_2 = 0.2$ as used in the experiment shown in Fig.3. In Fig.8(b), the obtained modularity of the dolphin network falls into the interval [0.43, 0.525], and gets 0.521 at $\omega_1 = 0.25$ and $\omega_2 = 0.2$ as used in the experiment shown in Fig.4.

4 Conclusions

In this paper, we have presented a new approach to mining communities from complex networks. The key idea behind this approach rests on a self-organizing network

evolving model presented here, in which a network is considered as an evolutionary system, and its community structure actually corresponds to a stable state of the evolving network directed by some predefined evolutionary rules. We test these models against several benchmark networks, and experimental results show their good performance.

First, two clustering attributes, the signs and the density of the links, are both considered by our approach, and thus it is suitable for both positive networks and signed networks. Second, our approach has a good classification capability and especially works well with the networks without well-defined community structures. Finally, our approach needs no prior knowledge and is insensitive to the built-in parameters.

Restricted by the space, in this paper we only discuss some social networks with moderate size. In our future work we will apply our approach to the networks with large size such as WWW or P2P networks and focus on the problems of community structure prediction and network stability analysis.

Acknowledgements

The research is supported by the National Natural Science Foundation of P.R.China under Grant No. 60503016.

References

1. Batagelj, V.: Semirings for Social Network Analysis. *Journal of Mathematical Sociology* 19 (1994) 53-68
2. Strogatz, S.H.: Exploring Complex Networks. *Nature* 410 (2001) 268-276
3. Watts, D.J., Strogatz, S.H.: Collective Dynamics of Small-World Networks. *Nature* 393 (1998) 440-442
4. Scott, J.: *Social Network Analysis: A Handbook*. 2nd edn. Sage Publications, London, (2000)
5. Zachary, W.W.: An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropological Research* 33 (1977) 452-473
6. Williams, R.J., Martinez, N.D. Simple Rules Yield Complex Food Webs. *Nature* 404 (2000) 180-183
7. May, R.M., Lloyd, A.L.: Infection Dynamics on Scale-Free Networks. *Physical Rev. E*. 64 (2001) 066112
8. Jeong, H., Tombor, B., Albert, R., Oltvai, Z., Barabasi, A.: The large-scale organization of metabolic networks. *Nature* 406 (2000) 651-654
9. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On Power-Law Relationships of the Internet Topology. *Computer Comm. Rev.* 29 (1999) 251-262
10. Albert, R., Jeong, H., Barabasi, A.L.: Diameter of the World-Wide Web. *Nature* 401 (1999) 130-131
11. Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Random Graphs with Arbitrary Degree Distributions and Their Applications. *Physical Rev. E* 64 (2001) 026118
12. Barabasi, A.L., Albert, R.: Emergence of Scaling in Random Networks. *Science* 286 (1999) 509-512
13. Newman, M.E.J.: Detecting Community Structure in Networks. *European Physical J.B.* 38 (2004) 321-330

14. Doreian, P., Mrvar, A.: A partitioning approach to structural balance. *Social Networks* 18 (1996) 149-168
15. Sampson, S.: Crisis in a cloister. Unpublished doctoral dissertation, Cornell University, (1969)
16. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.M.: Self-Organization and Identification of Web Communities. *IEEE Computer* 35 (2002) 66-71
17. Fiedler, M.: Algebraic Connectivity of Graphs. *Czechoslovakian Math. J.* 23 (1973) 298-305
18. Pothen, A., Simon, H., Liou, K.P.: Partitioning Sparse Matrices with Eigenvectors of Graphs. *SIAM J. of Matrix Analysis and Application* 11 (1990) 430-452
19. Fiedler, M.: A Property of Eigenvectors of Nonnegative Symmetric Matrices and Its Application to Graph Theory. *Czechoslovakian Math. J.* 25 (1975) 619-637
20. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. *IEEE Trans. On Pattern analysis and machine Intelligent* 22 (2000) 888-904
21. Kernighan, B.W., Lin, S.: An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical* 49 (1970) 291-307
22. Wu, F., Huberman, B.A.: Finding Communities in Linear Time: A Physics Approach. *European Physical J. B.* 38 (2004) 331-338
23. Burt, R.S.: Positions in Networks. *Social Forces* 55 (1976) 93-122
24. Wasserman, S., Faust, K.: *Social Network Analysis*. Cambridge Univ. Press, Cambridge (1994)
25. Girvan, M., Newman, M.E.J.: Community Structure in Social and Biological Networks. *Proc. Nat'l Academy of Science* 9 (2002) 7821-7826
26. Tyler, J.R., Wilkinson, D.M., Huberman, B.A.: Email as Spectroscopy: Automated Discovery of Community Structure within Organizations. *Proc. 1st Int'l Conf. Communities and Technologies*, Kluwer (2003)
27. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and Identifying Communities in Networks. *Proc. Nat'l Academy of Science*, 101 (2004) 2658-2663
28. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. *Proc. 9th Ann. ACM-SIAM Symp. Discrete Algorithms* (1998) 668-677
29. Gibson, D., Kleinberg, J., Raghavan, P.: Inferring Web Communities from Link Topology. *Proc. 9th ACM Conf. Hypertext and Hypermedia* (1998)
30. Pirolli, P., Pitkow, J., Rao, R.: Silk from a Sow's Ear: Extracting Usable Structures from the Web. *Proc. ACM Conf. Human Factors in Computing Systems, CHI*. ACM Press (1996)
31. Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.: Trawling the Web for Emerging Cyber-Communities. *Proc. 8th Int'l World Wide Web Conf.* (1999)
32. Chakrabarti, S., van der Berg, M., Dom, B.: Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery. *Proc. 8th Int'l World Wide Web Conf.* (1999)
33. Golub, G.H., Van L.C.F.: *Matrix Computations*. Johns Hopkins Univ. Press (1989)
34. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. *Phys. Rev. E* 69(2004) 066133
35. Newman, M.E.J., Girvan, M.: Finding and Evaluating Community Structure in Networks. *Physical Rev. E* 69 (2004) 026113
36. Lusseau, D.: The Emergent Properties of a Dolphin Social Network. *Proceedings of the Royal Society of London, Series B* 270 (2003) S186-S188

An Interactive Visualization Environment for Data Exploration Using Points of Interest

David Da Costa^{1,2} and Gilles Venturini²

¹ AGICOM 3 degré Saint Laumer 41000 Blois, France
ddacosta@agicom.fr

² Laboratoire d'Informatique 64, Avenue Jean Portalis 37200 Tours, France
{david.dacosta, venturini}@univ-tours.fr

Abstract. We present in this paper an interactive method for numeric or symbolic data visualization that allows a domain expert to extract useful knowledge and information. We propose a new approach based on points of interest (POI) but in the context of visual data mining. POIs are located on a circle, and data are displayed within this circle according to their similarities to these POI. Interactive actions are possible: selection, zoom, dynamical change of POI. We evaluate the properties of such visualization with standard data with known characteristics. We describe an industrial application which explores results from satisfaction inquiries.

1 Introduction

The methods of "Visual data mining" (VDM) try to solve the problems of interpretation and interaction in the knowledge discovery process by using dynamic visualizations and graphical requests on the represented data and knowledge [5], [11], [12]. By way of traditional examples, we can mention Chernoff's faces [4] which encodes data into icons while being based on the fact that the human mind easily analyzes the resemblances and differences between faces. We can also mention the "scatter plots" [2] which make it possible to obtain multiple views on the data and to observe the data using graphical techniques such as the "brushing" (which gives the possibility to select data in a view while underlining these same data in the other views).

These methods bring innovations and pursue goals which are promising for the field of VDM: the use of visual perception and often of preattentive perception [6], dynamic interaction with the data, easiness of use, direct use of the results. However, these methods also have limits as far as the VDM is concerned: the visualized data are generally numerical, visualizations and their handling requires user training (as it is the case for example to interpret graphs like "parallel coordinates" [8]), the dynamic interaction requires many resources of calculation (real time modifications) and must thus need the fastest possible algorithms (but which must in addition provide as much information as possible).

In this work, we suggest a new method of VDM, adapted itself from the methods involving points of interest and which are used for visualization of textual

data. Our objectives, in addition to those of VDM, are: to be able to represent all types of data on the basis of the existence of a similarity function (or distance) between the data, to have a very fast display when working with dynamic interactions and, if possible, to handle large volumes of data (algorithms with linear temporal and spatial complexities w.r.t. the number of data), to use a visualization requiring the shortest possible training time (thus understood by the majority of the potential users who are not regarded as experts in data mining).

2 Background of the Visualization Methods Involving Points of Interests

These methods are named by the terms "points of interests" or "points of references". They consist in positioning some specific icons (POIs) on a circle, and then to display the data icons within this disc at locations determined by the similarity between the POIs and the data. For example, this visualization was used as a method to display the documents resulting from a search engine request, which made it easier to navigate within all these returned documents. The selected POIs are generally keywords used in the request and the data are the documents which location is determined according to the proportion of keywords they contain. The choice of the keywords depends on their frequency in the documents. To visualize these data, one uses in general graphs displaying techniques involving springs and forces. The force being exerted between a POI and a data is proportional to the similarity between this POI and this data. The VIBE System [9], SQWID [10], Radviz [7] or the radial visualization of the system Information Navigator [1] use these principles. Sometimes it is difficult to see exactly toward which point of interest a data is attracted. In this case, these systems then make it possible to remove or add points of interest on the circle to obtain a better representation of the data. These are the principal interactive operations suggested by these methods.

Radial [1] is a generic example of such visualization methods. Initially, after extraction of the result of the request, Radial identifies a series of key terms relating to these results. Then the first 12 most important terms are arranged all around a circle. It is possible to modify the list of the displayed terms, the choice being done on two lists placed on the left of the screen. Then documents are displayed inside the disc. Only the data in connection with the keywords aligned around the circle are displayed. A document is like being suspended by springs connected to the keywords. It is thus impossible to move a document while clicking on it, because of the forces exerted by the springs. On the other hand, while clicking on a point, the keywords in connection with this point are lit and a bubble displays information on this data. It is possible to move the terms outside the circle and thus to move all the nodes of the data in connection with these terms. This makes it possible to make a manual classification of the results in different categories.

All these systems showed that this type of dynamic visualization brings a great interest for the user who can extract information quite easily. The speed of

display coupled with the interaction possibilities bring more to these methods. In addition, as we will show in this paper, they can visualize data of various types. To our knowledge, they have not yet been used for data mining as we will present it in the following sections.

3 Using Points of Interest in Data Mining

3.1 Basic Principles of Visualization

One considers n data D_1, \dots, D_n and a matrix of similarity Sim between these data. $Sim(i, j)$ is the similarity between the data D_i and D_j , this matrix being symmetrical and with a diagonal with 1s. If $Sim(i, j) = 1$ then the D_i and D_j data are identical, and if $Sim(i, j) = 0$ then they are completely different.

Initially, we will consider that the POIs are a subset of these data which are denoted by D_1, \dots, D_k . We display these k data on a circle with an arc of constant length between each POI (see figure 1).

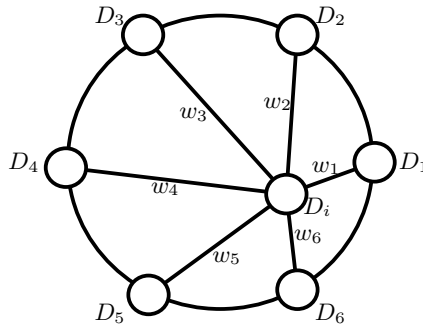


Fig. 1. Basic principles of our visualization with the positioning of a data D_i according to k POIs

One wants then to position the $n - k$ remaining data according to their similarity to the POIs D_1, \dots, D_k . We use the following formula to calculate the display coordinates (XD_i, YD_i) of data D_i :

$$XD_i = w_1 \times XD_1 + \dots + w_k \times XD_k \tag{1}$$

$$YD_i = w_1 \times YD_1 + \dots + w_k \times YD_k \tag{2}$$

with $w_1 + \dots + w_k = 1$. For this data D_i and a POI D_j , the weight w_j is computed in the following way:

$$w_j = \frac{Sim(D_i, D_j)}{\sum_{p=1}^k Sim(D_i, D_p)} \tag{3}$$

If D_i is identically similar to all of the POIs, it will be displayed in the middle of the disc. On the opposite, if it is completely similar to one POI and completely different from the others, its position will be confounded with that POI. If its similarity is biased towards certain POIs, then it will tend to approach these POIs.

More generally, our method is such that two data close to one another in the initial representation will thus be also close with respect to POIs, and they will thus be close in 2D space. The visualized space thus becomes a space of distances between selected points (POIs) and the data. It is in this manner that this method can deal with any type of data. We use a Euclidean distance for numerical data or a Hamming distance for symbolic data. On the other hand, the reciprocity of this property is not true: two data close in the 2D space are not necessarily close in the original space (all the points at equal distance of two POIs in the initial space form a mediating line and are thus displayed at the same 2D location). It will be necessary to use other methods to remove these ambiguities (see the last section).

Finally, displaying requires very little calculation and only needs to compute a part of the similarities ($k \times (n - k)$).

Several questions are raised by this method. First of all, the initial choice of the POIs must be carried out. Initially, we consider that if the data are supervised (a class label is available), then we take the first representative of each class as initial POIs. There will thus be as many POIs as classes in the first visualization suggested to the user. If the data are not supervised, we choose the first k data. Other automatic choices are possible (and certainly more judicious) as we describe it in the last section, and we try here to suggest initial choices that the user will be able to interactively and dynamically modify according to what is displayed (see the following section). A second question comes from the order of the POIs: if a great number of data are attracted by two POIs, then it is desirable that these POIs are close to each others on the circle. A critical situation would consist in placing these POIs in a diametrically opposed way, which would generate unreadable visualizations (many data in the center). We propose an interactive solution to this problem in the following section, but it is obvious that automatic solutions can be found like ordering POIs according to their similarity (see last section). It would also be possible not to preserve a fixed arc length between POIs in order to show the similarities that exist between POIs.

3.2 Interaction

To be really efficient, the visualization of information must be interactive and must make it possible to dynamically refine the display and to answer to the graphic requests of the user. In visualization with POIs, the user can ask for the following requests: what is this data (or this POI), how to enlarge this part of the visualization (zoom without loss of context), how to change POI (to remove some, to add some, to change their order, and possibly to define POIs which are not necessarily some data of the initial database).

When the mouse is moved over a point/data, we indicate what is this point. Then, it is possible to focus on a data by clicking on it. The zoom carries out

the following operations: it centers the data on the middle of the disc; it enlarges the area centered on this data and pushes the other data toward the edges of the visualization. The distortion is calculated using a hyperbolic function. This zoom makes it possible to enlarge the view while preserving the context of the global data display.

As far as the POIs are concerned, the main possible interactions are the following. First of all, it is possible to remove a POI. This is done very simply by dragging a POI inside the disc. This POI takes its place back within the data. The view is dynamically recomputed. A dynamic and progressive transition is performed so that the user can follow the change of representation. He then has the possibility to cancel its action, which causes to put the POI back on the circle. It is also possible to choose a data and to define it as a POI. For this purpose, one drags the data on the circle. If the data is placed on a POI, it replaces this POI, and if it is placed between two POIs, it is inserted between them. The length of the arcs between POIs is kept constant. These functionalities are very significant since they allow the user to redefine at will the representation. However, the initial k POI are important for this method because it is necessary to have at least three POI around the circle. If there are only two POI then all the data will be placed on the line formed by these two POI. If too many POIs are placed around the circle then all data shrink at the center of the visualization.

Lastly, it is possible to generalize POIs so that they are not necessarily some data any more, but more generally any point of the space of representation and even any object for which it is possible to compute a similarity with the data. Thus, one can represent "ideal" data, not really existing, and according to which the user would like to position the real data. We present in section 4 a typical application of this functionality. Also, it would be possible to represent for example a decision rule as a POI, and to place the data according to their matching with this rule. This functionality offers many perspectives by visualizing not only data but also knowledge.

4 Results

4.1 Artificial and Traditional Bases

We have evaluated this method on various artificial and standard data sets. Figure 2 represents an artificial database Art1 made up of 400 data and 4 classes. We illustrate in particular the effects of a zoom. When one has only two classes, the data of the two classes are positioned on the segment ranging from POI1 to POI2. To help the user finding better visualization of the data, one may add another POI or several POIs.

Finally, we have tested our approach on traditional databases from the "Machine Learning Repository" [3]. We thus represent on figure 3 the Iris database (150 data, 3 classes), the Wine database (178 data, 3 classes) and Segment database (2310 data, 7 classes). The expected shapes of these databases are easily found in our visualization (as for Iris and Wine for example).

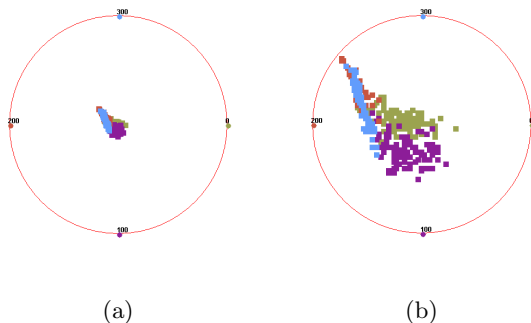


Fig. 2. Visualization of Art1 data without zoom (a) and with zoom (b)

4.2 Real World Application

Some of the activities of Agicom consist in collecting data that result from satisfaction inquiries using questionnaires. These data can be considered as an individual \times variables table where these variables are qualitative (with ordered values). The values of such variables can be for instance ("delighted", "satisfied", "unsatisfied", "disappointed" and "NSP" ("Do not know")).

For a domain expert, in order to have the possibility to exploit these data, it is important to be able to graphically visualize the satisfaction of the customers. The aims of the expert are for instance to detect possible correspondences between individuals, to know the evolution of the customers from one segment to another, but also to visualize the existing relation between a given variable and the other variables. Our goal is thus to design a tool for representing the results of the satisfaction inquiries, with the final aim to understand and improve the user satisfaction.

We have evaluated and tested our method on the Agicom1 database made up of 31 unsupervised data. Figure 4(a) illustrates this first application in which the POIs are not data but typical profiles of variables. A profile thus corresponds

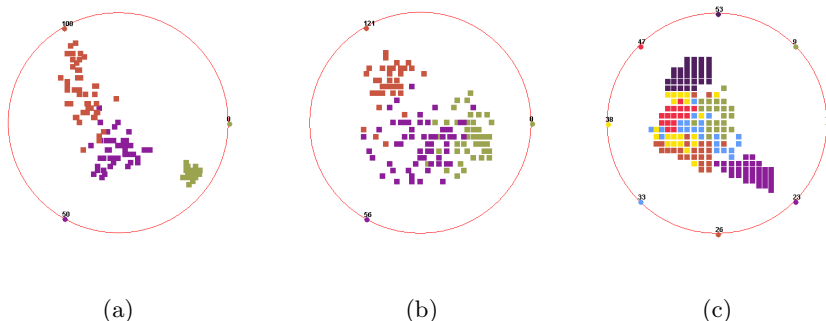


Fig. 3. Visualizations of Iris (a), Wine (b), and Segment (c) databases

to a distribution of values (answers) for this variable. In this manner, the POIs represent various known typologies of variables (like very positive answers, or extremely positive or negative answers, etc).

We present a second example on the Agicom2 database (see figure 4(b)). In this application, we have allowed the Agicom users to interact with the characteristics of POIs and with the zoom (see figure 4(b)). Moreover, in this database we allowed the visualization of the various classes. A validation with real users is currently under study.

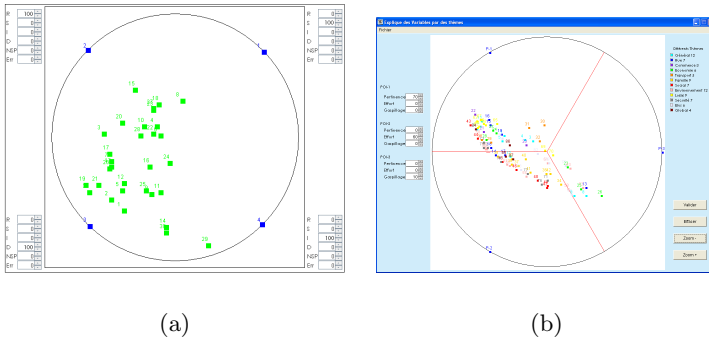


Fig. 4. Visualization of Agicom1(a) and Agicom2(b) database

5 Conclusion and Perspectives

We have described in this paper a new visualization method inspired from the work performed in the context of points of interest. It consists in transforming a initial space represented by a similarity matrix into a visual 2D representation of these similarities. This method has advantages like the speed of display, an intuitive presentation of the data, rather fast user learning, and interactive abilities. We have detailed its behavior on on traditional data and finally within a real application being developed by Agicom.

Several perspectives can emerge. We mentioned the importance of the choice of POIs as well as their location on the circle. A first extension consists in studying the use of an optimization algorithm in order to find the most effective ordering of POIs. This consists in finding some relevant permutations of the k chosen POIs. Another significant perspective consists in extending the visualization so as to remove ambiguities related to the overlapping data. We intend to use a method of graph display based on forces and springs in order to move away the points that are too close to each others on the graph. We also want to make a distinction between the points which are placed at the same location but which have different mean similarity with the POIs. A 3D approach will be tested soon for this purpose.

References

1. Peter Au, Matthew Carey, Shalini Sewraz, Yike Guo, and Stefan M. Ruger. New paradigms in information visualization. In *Research and Development in Information Retrieval*, pages 307–309, 2000.
2. R. A. Becker and W. S. Cleveland. Brushing Scatterplots. *Technometrics*, 29:127–142, 1987. Reprinted in *Dynamic Graphics for Data Analysis*, edited by W. S. Cleveland and M. E. McGill, Chapman and Hall, New York, 1988.
3. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
4. H. Chernoff. Using faces to represent points in k -dimensional space graphically. *Journal of the American Statistical Association*, 68:361–368, 1973.
5. W. S. Cleveland. *Visualizing Data*. Hobart Press, Summit, New Jersey, U.S.A., 1993.
6. Christopher G. Healey, Kellogg S. Booth, and James T. Enns. Harnessing preattentive processes for multivariate data visualization. In *Proceedings of Graphics Interface '93*, pages 107–117, Toronto, ON, Canada, May 1993.
7. Patrick Hoffman, Georges Grinstein, and David Pinkney. Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In *NPIVM '99: Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM international conference on Information and knowledge management*, pages 9–16, New York, NY, USA, 1999. ACM Press.
8. Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.
9. Robert Korfhage. To see, or not to see: Is that the query? In Abraham Bookstein, Yves Chieramella, Gerard Salton, and Vijay V. Raghavan, editors, *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Chicago, Illinois, USA, October 13-16, 1991 (Special Issue of the SIGIR Forum)*, pages 134–141. ACM, 1991.
10. S. McCrickard and C. Kehoe. Visualizing search results using sqwid. In *Proceedings of the Sixth International World Wide Web Conference*, April 1997.
11. Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Visual Languages*, number UMCP-CSD CS-TR-3665, pages 336–343, College Park, Maryland 20742, U.S.A., 1996.
12. Pak Chung Wong and R. Daniel Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization — Overviews, Methodologies and Techniques*, pages 3–33. IEEE Computer Society Press, Los Alamitos, CA, 1997.

Forecasting the Volatility of Stock Price Index

Tae Hyup Roh

Department of Business, Management Information System, Seoul Women's University,
139-774 126 Gongreung-2Dong, Nowon-Gu, Seoul, Korea
rohth@swu.ac.kr

Abstract. Accurate volatility forecasting is the core task in the risk management in which various portfolios' pricing, hedging, and option strategies are exercised. Prior studies on stock market have primarily focused on estimation of stock price index by using financial time series models and data mining techniques. This paper proposes hybrid models with neural network and time series models for forecasting the volatility of stock price index in two view points: deviation and direction. It demonstrates the utility of the hybrid model for volatility forecasting.

1 Introduction

Since the inception of KOSPI(Korea Composite Stock Price Index) 200 option market, July 7, 1997, the trading market volume has steeply increased. Investors and financial institutions became worried about risks caused by increasing volatility of KOSPI 200. Accurate volatility forecasting is an essential part in performing risk management, specifically in allocating assets to various portfolios in order to hedge those portfolios' risk efficiently.

There are various models to forecast time series volatilities. Engle suggested ARCH(p) (Autoregressive Conditional Heteroscedasticity) model that has been used by many financial analysts [4]. GARCH (Generalized ARCH) model is the generalized form of ARCH [2]. RiskMetrics by JP Morgan suggested EWMA(Exponentially Weighted Moving Average) [8] which is basically a non-stationary GARCH(1,1) model. By considering the limitation of GARCH model, leverage effects, EGARCH(Exponential GARCH) model was proposed [12]. These financial time series models can be analyzed with econometrics and be systematically explained based on the market and financial theories. However, due to many noises that are caused by changes in market conditions and environments, financial analysts should consider many market variables. Financial time series models require strict assumptions about distributions of time series, so it is hard to reflect market variables directly in the models. Due to these shortcomings of financial time series models, ANN(Artificial Neural Network) has been applied to various complex financial markets directly. ANN model is a nonparametric method and can forecast future results by learning the pattern of market variables without strict theoretical assumption. Taking advantage of these characteristics, it was revealed that ANN can outperform financial time series models by analyzing S&P 500 future index option volatilities [6].

The possibility of ANN was demonstrated in forecasting the volatilities of financial time series [3].

The objective of this study is that the forecasting power in stock price index domain can be enhanced by integrating financial time series models, such as EWMA, GARCH, EGARCH and ANN model. Prior studies have compared mainly the predicting power between single models. However, this study focuses on two view points: the deviation and direction of stock price index. In addition, most of the prior studies have adjusted the weight of raw volatilities by repetitive trial and error method of learning process and found the optimal coefficient of input variables to produce the best results. This study finds the coefficients of input variables by financial time series process and extracts new variables that greatly influence the results.

Having introduced the research background, the remaining sections of this paper are organized as follows. The section two provides a brief review of the related work. The section three and four present the proposed methodology about the ANN-Financial time series models and experiments. The last two sections state results and the contribution of the research and future consecutive research issues.

2 Related Work

2.1 Statistical Time Series Volatility Forecasting

As a classic statistical time series model, ARCH(p) was proposed to model the characteristics of time series that have volatility clustering and fat tail [4]. Because ARCH(p) causes time lag p to get much larger in forecasting volatility, generalized ARCH as GARCH(p,q) was suggested[2]. GARCH model restricts its parameters to have plus value of conditional variance. These conditions would make the conditional variance process much restrictive over the necessity. The original GARCH model does not consider the negative correlation between future yield and volatility. By the same token, when market is falling against participant's expectation (negative impact), negative effect has bigger influence than same-sized positive effect. This asymmetric shock is generally called leverage effect. The other hand, the general GARCH model results in symmetric shock without regards to the sign of the impact of conditional volatilities because the square of present yield's residuals have influence in future yield's volatility. By considering this leverage effects, EGARCH model was developed [12].

EWMA model gives more weight on the recent data than others included in time series. This method is modeled on "RiskMetrics" by JP Morgan in parametric way. In the case of EWMA, when the objective of volatility forecasting is to catch the short-term movement of it, EWMA is desirable. However, if EWMA places much value only on the recent data, then it reduces the sample size and brings about the result of increasing possibility of measurement error. EWMA has shortcoming in describing characteristics of the financial time series volatility – 'long term memories'.

2.2 Stock Market Applications with Data Mining Techniques

Many studies on stock market prediction using data mining techniques have been performed during the past decade. The early days of these studies focused on estimating

the level of the return on stock price index. One of the earliest studies, Kimoto, Asakawa, Yoda, and Takeoka used several learning algorithms and prediction methods for developing a prediction system for the TSEPI (Tokyo Stock Exchange Prices Index) [11]. They used the modular neural network to learn the relationships among various market factors. They concluded that the correlation coefficient produced by their model is much higher than that produced by multiple-regression.

Some researchers investigated the issue of predicting the stock index futures market. Trippi and DeSieno predicted the daily direction of change in the S&P 500 index futures using ANN [13]. They combined the outputs of individual networks using logical (Boolean) operators to produce a set of composite rules. They suggested that their best composite synthesized rule set system achieved a higher gain than previous research. Recent research tends to hybridize several AI techniques. Nikolopoulos and Fellrath developed a hybrid expert system for investment advising [16]. In their study, genetic algorithms were used to train and configure the architecture of investor's neural network component. A more recent study of Lee and Jo developed an expert system, which uses knowledge in a candlestick chart analysis [17]. The expert system had patterns and rules, which could predict future stock price movements. The experimental results revealed that a developed knowledge-base could provide excellent indicators. In addition, Tsaih, Hsu and Lai integrated a rule-based technique and ANN to predict the direction of change of the S&P 500 stock index futures on a daily basis [14]. In addition, some researchers searched the connection weights of ANN using the GA instead of local search algorithms including a gradient descent algorithm. They suggested that global search techniques including the GA might prevent ANN from falling into a local optimum [9, 10, 18, 19].

Past studies proved that ANN model can enhance the predictive power based on real market data and suggested that it is important to integrate ANN model and financial time series models for future studies [6]. Especially, Hu proposed that predictive power can be improved by ANN in which the forecasted results are relearned in ANN learning process, and he suggested that forecasted results should be studied if they could contribute the ANN forecasting process [7]. Various studies using ANN have been developed in the fields of forecasting stock index [1, 5, 9, 15].

These various studies have proposed the basis in which ANN could be applied to financial engineering and nowadays active researches are developed in the fields of finance and forecasting in which parametric models cannot fully explain the characteristics of their market behaviors.

3 Hybrid ANN-Time Series Model

A common shortcoming of ANN in forecasting volatility is that it is not proved econometrically. Also, users get curiosities about the ANN results because input variables which are the most important factors to influence the forecasted results are determined in the learning process of repetitive trials and errors. In this study, the hybrid ANN-time series model is proposed to solve the difficulties in the ANN learning process and to enhance the predictive power in forecasting volatility.

The hybrid model achieves the efficiency in selecting input variables because they are selected and newly created by the financial time series models. Repetitive trial

error process could be eliminated to one time financial time series process. Input variables that are most weighted on the forecasting are usually contracted to one or two variables. The ANN models have spent most of time to find input variables by repetitive trial and errors. This study will prove that hybrid model can improve the predictive power in the framework of both direction and deviation.

Theoretically, ANN can approximate any function and financial time series models can be fairly approximated by this flexibility. Therefore, the characteristic of conditional volatility by GARCH(1,1) can be approximated through ANN and the methodology of approximation is achieved by inputting variables obtained through GARCH(1,1) process and learning the conditional volatility pattern through ANN process. ANN can adjust these results by using other market variables realistically to reflect real market behaviors. Input variables can be extracted efficiently by financial time series models and ANN can improve the predictive power by using these variables and other market variables from the perspective of deviation and direction of stock price index.

3.1 NN-EWMA Model

NN-EWMA model is to give more weight on recent data and catch the short term volatility behaviors by extracting variables through EWMA model. The equation of EWMA is following.

$$\sigma_t^2 = \lambda \sigma_{t-1}^2 + (1 - \lambda) \varepsilon_{t-1}^2 \tag{1}$$

EWMA gives more weight on recent data and reduce shadow effect in which past data severely influences the prediction. NN-EWMA models can be expressed with two variables.

- σ_{t-1}^2 : square of volatility at t-1
- ε_{t-1}^2 : square of residuals at t-1

Two newly created input variables can be extracted based on the above variables and each variable is adjusted by smoothing factor (decay factor) $\lambda, (1-\lambda)$ and then, included in the ANN input variables. Newly created variables are followings.

- $\sigma_{t-1}' = \lambda \sigma_{t-1}^2$
- $\varepsilon_{t-1}' = (1 - \lambda) \varepsilon_{t-1}^2$

σ_{t-1}' and ε_{t-1}' that are extracted through EWMA model, are similar with variables that are extracted in GARCH(1,1) model, but two model started from totally different conditional variance concept. Namely, the coefficients of GARCH(1,1) model are extracted through financial time series statistically, but those of EWMA are determined by user's discretion based on the experience.

3.2 NN-GARCH Model

ANN-time series models are to extract predictive or adjusted input variables by financial time series models. Most of financial time series model are known to be easily modeled by GARCH(1,1), so in the first place, this paper will try to extract input variables using GARCH(1,1) model. GARCH(1,1) process from time series data is followings.

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{2}$$

GARCH(1,1) model brings about similar effect like using long time lag ARCH model even if it uses small number of parameters. Therefore, it is desirable to use GARCH(1,1) model to make time series which have the characteristics of volatility clustering and fat tail from the perspective of conditional variances. σ_t^2 is the one-period ahead forecast conditional variance. This conditional variance equation is expressed with following three variables.

- α_0 : Nonconditional volatility coefficient
- ε_{t-1}^2 : residual at t-1
- σ_{t-1}^2 : square of volatility at t-1

Consequently, ε_{t-1}^2 and σ_{t-1}^2 that have conditional relations with each other can be extracted and the coefficients of these variables are adjusted to α_1 and β_1 . These are included in the input variables for ANN learning process. Newly extracted variables are followings.

- $\sigma_{t-1}'^2 = \beta_1 \sigma_{t-1}^2$
- $\varepsilon_{t-1}'^2 = \alpha_1 \varepsilon_{t-1}^2$

3.3 NN-EGARCH Model

NN-EGARCH model is to improve the restrictive conditional variance process of GARCH(1,1) model, which is like following equation.

$$\ln \sigma_t^2 = \alpha + \beta \ln \sigma_{t-1}^2 + \gamma \left(\left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} - \sqrt{\frac{2}{\pi}} \right| \right) + \omega \frac{\varepsilon_{t-1}}{\sigma_{t-1}} \tag{3}$$

EGARCH model can explain the asymmetric shock that means large falling in yield can make the next period volatility greater-leverage effect. This $\frac{\varepsilon_{t-1}}{\sigma_{t-1}}$ term in the equation (3) explains the leverage effect by the market shock. EGARCH model can statistically explain asymmetric shock which cannot be explained by conditional variance model-GARCH. EGARCH model is expressed with following 4 variables.

- α : unconditional variance coefficient
- $\ln \sigma_{t-1}^2$: log value of variance at t-1
- $\left(\left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} - \sqrt{\frac{2}{\pi}} \right| \right)$: Asymmetric shock by leverage effect.
- $\frac{\varepsilon_{t-1}}{\sigma_{t-1}}$: leverage effect

New input variables can be extracted based on the above variables and each variable is adjusted by β, γ, ω and then gets included in ANN model.

- $\ln \sigma_{t-1}^2 = \beta \ln \sigma_{t-1}^2$
- $LE(\text{leverage} \cdot \text{effect}) = \gamma \left(\left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} - \sqrt{\frac{2}{\pi}} \right| \right)$
- $L(\text{leverage}) = \omega \frac{\varepsilon_{t-1}}{\sigma_{t-1}}$

Existing studies use observable market variables and various ones that can be created by financial theories but usually not use newly created variables for ANN. It is the newly created leverage and leverage effect variables that differ from existing ANN models in NN-EGARCH model. To specify the efficiency and predictive power by newly created variables, NN-EGARCH model would make $\ln \sigma_{t-1}^2, LE, L$ input variable with other market variables and analyze the predictive power during NN-EGARCH learning process.

4 Experiment

4.1 Dataset and Experimental Setup

This study conducted experiments to evaluate the proposed model. For experiments, we used KOSPI 200 time series data and option daily trading materials, provided by KSE KOSPI 200 index database and daily option data. The dataset consists of 930 trading days' indexes for sample period and 160 indexes for prediction period.

To verify the appropriateness of financial time series models, this study performs ADF(Augmented Dickey-Fuller) test and ARCH LM(Lagrange Multiplier) test. Fig.1 shows the daily KOSPI 200 index for experiment dataset.

ADF test

ADF test is used for verifying the stability. Because there is a unit root in sample data by ADF test, this time series is nonstationary time series that needs to be differencing. Therefore, this time series are changed to KOSPI 200 logarithmic yields time series that are stationary. The logarithmic value of index returns is calculated. As a result of ADF test (Table1), this transformed time series is stationary that is oscillating around mean value 0.

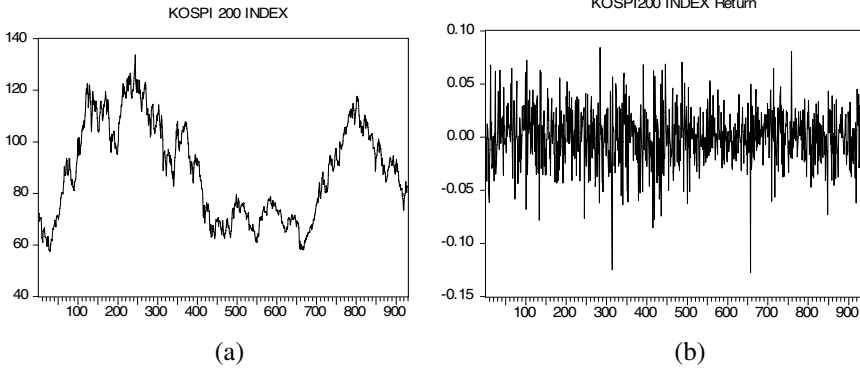


Fig. 1. a)KOSPI 200 time series; b)Logarithmic time series of KOSPI 200 returns

Table 1. ADF test of KOSPI 200 index time series datasets

ADF Test Statistic	-1.929573	1% Critical Value*	-3.4401
		5% Critical Value	-2.8651
		10% Critical Value	-2.5687

ARCH LM Test

ARCH LM test is needed to verify the Heteroscedasticity in time series. The null hypothesis that there is no heteroscedasticity to 5 time lags is rejected by 5% significance level, so there is heteroscedasticity within 5 time lags. At the same time, joint significance of lagged squared residual to 5 time lags is accepted by F-statistic. Therefore, the ARCH model is appropriate in modeling the volatility of KOSPI 200 return, and GARCH(1,1) model is recommended because GARCH(1,1) model can be changed to ARCH(∞) by the repetitive input process. (See Table 2).

Table 2. Results of 5-lag ARCH-LM test

ARCH Test: 5-lag			
F-statistic	2.345315	Probability	0.039580
Obs*R-squared	11.65435	Probability	0.039844

To verify the forecastability of hybrid ANN-time series model, this paper inspects the reasonableness of financial time series model by analyzing KOSPI 200 time series. Then, NN-GARCH(1,1) is analyzed to prove that integration of ANN and GARCH can enhance the predictive power. After analyzing NN-GARCH model, this part will suggest which model(GARCH, EGARCH, EWMA) has most predictive power in forecasting volatility of KOSPI 200 index.

The process of verification can be classified into two ways. ANN, GARCH(1,1) and NN-GARCH(1,1) model are compared with one another from the points of deviation(MAE) and direction(Hit ratio). To assess the volatility, this study uses the realized volatility defined as true volatility in the market and used to compare predictive power between each models. Therefore, in this study, realized volatility is defined as 22-day true standard deviation of the logarithmic return of KOSPI 200 index. Realized volatility of yields is following.

$$TRV_{\tau} = \sqrt{\frac{1}{\tau} \sum_{k=1}^{\tau} (r_{t,k} - \bar{r}_t)^2}$$

TRV : True Realized Volatility

τ_t : 22 days after t

$$\bar{r}_t = \frac{1}{\tau_t} \sum_{k=1}^{\tau_t} r_{t,k} \quad \text{: mean of realized return during 22 days after t.}$$

$r_{t,k}$: return of KOSPI 200 index at month t, day k.

Computed realized volatility is expressed as 22 days volatility, $\sigma_{22 \text{ days}} = \sigma_{1 \text{ day}} \times \sqrt{22}$, to be compared with forecasted 22 days volatility of KOSPI 200 index.

Table 3. Expected learning effects of extracted input variables

Models	Extracted variables and expected learning effects
NN	• σ_{t-1}^2 : Simple volatility learning at t-1 without repetitive trial and error.
NN-GARCH	• $\sigma_{t-1}^2 = 0.83661 \ln \sigma_{t-1}^2$: Learning the pattern of conditional volatility between one period time intervals
	• $\varepsilon_{t-1}^2 = 0.064785 \varepsilon_{t-1}^2$: Learning the residual effects to learn the conditional volatility
NN-EGARCH	• $\ln \sigma_{t-1}^2 = 0.900850 \ln \sigma_{t-1}^2$: Learning the pattern of conditional volatility between one period time intervals
	• $L(\text{leverage}) = -0.062328 \frac{\varepsilon_{t-1}}{\sigma_{t-1}}$: Learning the leverage effects
	• $LE (\text{leverage} - \text{effect}) = 0.138036 \left(\left \frac{\varepsilon_{t-1}}{\sigma_{t-1}} - \sqrt{\frac{2}{\pi}} \right \right)$: Learning the asymmetric shock by leverage effects
NN-EWMA	• $\sigma_{t-1}^2 = 0.97 \sigma_{t-1}^2$: Learning the pattern of volatility adjusted by decay factor between one period time intervals
	• $\varepsilon_{t-1}^2 = 0.03 \varepsilon_{t-1}^2$: Learning the residual effects adjusted by decay factor between one period time intervals

4.2 Variable Coefficients Estimation

Table 3 summarizes the expected learning effects of extracted input variables by each model. NN model is added to compare the results and to analyze the learning effects by integration. NN model is analyzed for the purpose of comparison among the proposed models. Table 4 exhibits the relative contribution factors produced by ANN learning process. The contribution factors of input variables extracted by time series models are adjusted as if it were the coefficients of financial time series models. The statistical meanings of extracted input variables are expressed by relative contribution allocation. These results can be used as a statistical and neural network's basis to propose desirable models.

Table 4. Input variables and relative contribution factors

Input Variables	NN	NN- EWMA	NN- GARCH	NN- EGARCH
KOSPI200 yield square	0.0518	0.0532	0.0567	0.0393
Promised volume	0.0545	0.0452	0.0421	0.0422
KOSPI200 at t-1	0.0568	0.0489	0.0504	0.0516
KOSPI200 yield	0.0583	0.0626	0.0602	0.0534
3-month government bond price	0.0596	0.0555	0.0567	0.0614
1-year government bond yield	0.0605	0.0519	0.0561	0.0489
Open interest volume	0.0633	0.0555	0.0564	0.0528
Premium average	0.0654	0.0623	0.0693	0.0606
Contract volume	0.0667	0.0591	0.0593	0.0566
1-year government bond price	0.0674	0.0593	0.0623	0.0576
3-month government bond yield	0.0685	0.0556	0.0549	0.0558
KOSPI 200 at t	0.0731	0.0722	0.0648	0.0680
σ^2_{t-1}	0.2542			
$\sigma^2_{t-1}' = 0.97 \sigma^2_{t-1}$		0.2244		
$\varepsilon^2_{t-1}' = 0.03 \varepsilon^2_{t-1}$		0.0946		
$\sigma^2_{t-1}' = 0.836611 \sigma^2_{t-1}$			0.2144	
$\varepsilon^2_{t-1}' = 0.064785 \varepsilon^2_{t-1}$			0.0963	
$\ln \sigma^2_{t-1}' = 0.900850 \ln \sigma^2_{t-1}$				0.2176
LE (leverage - effect)				0.0778
L(leverage)				0.0565
Total	1.0000	1.0000	1.0000	1.0000

5 Results

Deviation comparison is done through MAE. The sequence of smallest MAE is NN-EWMA > NN > NN-GARCH > NN-EGARCH. The MAE of NN-EWMA is greater

than NN, so this result can specify that the adjustment by the decay factor λ in NN-EWMA model is not desirable in the same manner in the pure EWMA models. Therefore, this step can confirm that financial time series model that is not successful in volatility forecasting is also not successful in hybrid models, for example NN-EWMA. This outcome can be confirmed by the results of NN-EGARCH model that shows smallest MAE, because EGARCH model is very successful in forecasting volatilities. NN-EGARCH model can produce new input variables, such as leverage and leverage effect. These newly created variables improve prediction of volatilities with the help of logarithmic conditional variance equations. Table 5 shows this result more clearly by the comparison of the forecastability rising ratios.

NN-GARCH and NN-EGARCH model are inferred to enhance the predictive power in the point of MAE compared with NN model. Especially, NN-EGARCH model(29.43% rising) is much better model than NN-GARCH(7.67% rising) in MAE. To guarantee this result statistically, one-way ANOVA is implemented in Table 6.

Table 5. Rising of precision accuracy of hybrid models compared with NN model

Accumulated forecasting days	NN-EWMA	NN-GARCH	NN-EGARCH
20	-52.53%	-15.87%	36.94%
40	-23.61%	-5.15%	8.70%
60	-16.00%	-0.19%	9.99%
80	-7.26%	5.45%	9.42%
100	-0.96%	13.03%	12.57%
120	-0.51%	16.91%	13.16%
140	-4.16%	14.86%	18.21%
160	-12.42%	7.67%	29.43%
	NN-EWMA < NN-GARCH29 < NN-EGARCH15		
formula	$\frac{(MAE_{NN} - MAE_{comparison_model})}{MAE_{NN}}$		

Table 6. One way ANOVA for comparing MAE

	NN	NN-EWMA	NN-GARCH	NN-EGARCH
NN		0.5638	0.8543	***0.0098
NN-EWMA	0.5638		0.1492	***0.0000
NN-GARCH	0.8543	0.1492		*0.1000
NN-EGARCH	***0.0098	***0.0000	*0.1000	

*** 1% significance level ** 5% significance level * 10% significance level

Direction comparison is done through hit ratio analysis, and hit ratio of NN model can be increased by the NN-GARCH and NN-EGARCH model. Table 7 shows that NN-GARCH(60.00%) and NN-EGARCH(60.63%) increase hit ratio compared with

NN (43.75%). Especially, Hit ratios of hybrid models are great in the early stage of forecasted interval. Therefore, short term (under 30 days) volatility forecast by hybrid model is more excellent.

By the analysis of MAE and hit ratio, NN-GARCH and NN-EGARCH model show good performance compared with NN model. By the same token like this analysis, the extraction of new input variables like leverage effect by financial time series model can enhance the predictive power in the overall perspectives.

Table 7. Result of hit ratio by each model

Accumulated forecasting days	NN	NN- EWMA	NN- GARCH29	NN- EGARCH15
10	30.00%	30.00%	100.00%	100.00%
20	40.00%	35.00%	80.00%	85.00%
40	32.50%	32.50%	62.50%	57.50%
60	41.67%	43.33%	61.67%	58.33%
80	47.50%	45.00%	58.75%	58.75%
100	46.00%	45.00%	60.00%	58.00%
120	44.17%	43.33%	57.50%	58.33%
140	42.86%	42.14%	58.57%	59.29%
160	43.75%	43.13%	60.00%	60.63%

6 Conclusions

This study proposed the hybrid model between ANN and financial time series model to forecast volatilities of stock price index. It specified that ANN-time series models can enhance the predictive power for the perspective of deviation and direction accuracy. Most of prior studies have adjusted the weight of raw volatilities by repetitive trial and error of learning process and found the optimal coefficient of input variables to produce the best results. This study found the coefficients of input variables by financial time series process and extracted new variables that greatly influence the results through analyzing stock market domain.

Experimental results showed that the proposed hybrid NN-EGARCH model could be improved in forecasting volatilities of stock price index time series. Of course, there are still many tasks to be done for the hybrid ANN-time series model. These NN-time series models should be further tested for robustness by applying them to other problem domains.

Acknowledgements

This study was supported by research grant of Social Science Research Center, Seoul Women's University in the program year of 2006.

References

1. Armano, G., Marchesi, M., Murru, A.: A Hybrid Genetic-neural Architecture for Stock Indexes Forecasting. *Information Sciences*. 170 (2005) 3-33
2. Bollerslev, T.: Generalized Autoregressive Conditional Heteroscedasticity. *Journal of Econometrics*. 31 (1986) 307-327
3. Brooks, C.: Predicting Stock Index Volatility: Can Market Volume Help? *Journal of Forecasting*. 17 (1998) 59-80
4. Engle, R. F.: Autoregressive Conditional Heteroscedasticity with Estimator of the Variance of United Kingdom Inflation. *Econometrica*. 50(4) (1982) 987-1008
5. Gavrishchaka, V., Ganguli, S.B.: Volatility Forecasting from Multi-scale and High Dimensional Market Data. *Neurocomputing*. 55 (2003) 285-305
6. Hamid, S. A., Zahid, I.: Using Neural Networks for Forecasting Volatility of S&P 500 Index Futures Prices. *Journal of Business Research*. 5881 (2002) 1-10
7. Hu, M. Y., Tsoukalas, C.: Combining Conditional Volatility Forecasts Using Neural Networks: An Application to the EMS Exchange Rates. *Journal of International Financial Markets, Institution and Money*. 9 (1999) 407-422
8. JP Morgan and Reuters.: *RiskMetrics - Technical Document, Fourth Edition*, New York. (1996)
9. Kim, K.: Financial Time Series Forecasting Using Support Vector Machines. *Neurocomputing*. 55 (2003) 307-319
10. Kim, K. Han, I.: Genetic Algorithms Approach to Feature Discretization in Artificial Neural Networks for the Prediction of Stock Price Index. *Expert Systems with Applications*. 19(2) (2000) 125-132
11. Kimoto, T., Asakawa, K., Yoda, M., Takeoka, M.: Stock Market Prediction System with Modular Neural Network. *Proceedings of the International Joint Conference on Neural Networks*, San Diego, California. (1990) 1-6
12. Nelson, D. B.: Conditional Heterosdasticity in Asset Returns: A New approach. *Econometrica*. 59(2) (1991) 347-370
13. Trippi, R.R. DeSieno, D.: Trading Equity Index Futures with a Neural Network. *The Journal of Portfolio Management*. 19 (1992) 27-33
14. Tsaih, R., Hsu, Y., Lai, C.C.: Forecasting S&P 500 Stock Index Futures with a Hybrid AI system. *Decision Support Systems*. 23(2) (1998) 161-174
15. Yao, J. T., Li, Y., Tan, C. L.: Option Price Forecasting Using Neural Networks. *Omega*, 28 (2002) 455-466
16. Nikolopoulos, C., Fellrath, P.: A Hybrid Expert System for Investment Advising. *Expert Systems*. 11(4) (1994) 245-250
17. Lee, K.H., Jo, G.S.: Expert Systems for Predicting Stock Market Timing Using a Candlestick Chart. *Expert Systems with Applications*. 16(4) (1999) 357-364
18. Gupta, J.N.D., Sexton, R.S.: Comparing Backpropagation with a Genetic Algorithm for Neural Network Training. *Omega* 27(6) (1999) 679-684
19. Sexton, R.S., Dorsey, R.E., Johnson, J.D.: Toward Global Optimization of Neural Networks: A Comparison of the Genetic Algorithm and Backpropagation, *Decision Support Systems*. 22(2) (1998) 171-185

ExMiner: An Efficient Algorithm for Mining Top-K Frequent Patterns

Tran Minh Quang, Shigeru Oyanagi, and Katsuhiko Yamazaki

Graduate school of Science and Engineering Ritsumeikan University, Kusatsu city Japan
{quang@cpsy.cs, oyanagi@cs, yamazaki@cs}.ritsumei.ac.jp

Abstract. Conventional frequent pattern mining algorithms require users to specify some minimum support threshold. If that specified-value is large, users may lose interesting information. In contrast, a small minimum support threshold results in a huge set of frequent patterns that users may not be able to screen for useful knowledge. To solve this problem and make algorithms more user-friendly, an idea of mining the k -most interesting frequent patterns has been proposed. This idea is based upon an algorithm for mining frequent patterns without a minimum support threshold, but with a k number of highest frequency patterns. In this paper, we propose an explorative mining algorithm, called *ExMiner*, to mine k -most interesting (i.e. *top-k*) frequent patterns from large scale datasets effectively and efficiently. The *ExMiner* is then combined with the idea of “build once mine anytime” to mine *top-k* frequent patterns sequentially. Experiments on both synthetic and real data show that our proposed methods are more efficient compared to the existing ones.

1 Introduction

Frequent pattern mining is a fundamental problem in data mining and knowledge discovery. The discovered frequent patterns can be used as the input for analyzing association rules, mining sequential patterns, recognizing clusters, and so on. However, discovering frequent patterns in large scale datasets is an extremely time consuming task. Various efficient algorithms have been proposed and published on this problem in the last decade. These algorithms can be classified into two categories: the “candidate-generation-and-test” approach and the “pattern-growth” approach.

Apriori algorithm [1] is the representative of the “candidate-generation-and-test” approach. This algorithm applies the monotonicity property of frequent patterns (*every subset of a frequent pattern is frequent*) to create candidates for k -itemset frequent patterns from a set of $k-1$ -itemset frequent patterns. The candidates will be verified whether satisfy the minimum support threshold by scanning over the dataset. Follow this approach, an extremely huge number of candidates are generated, and the dataset is scanned many times slowing down the response time of the algorithm.

The representative of the “pattern-growth” approach is the *FP-growth* algorithm [2] which scans dataset only twice to compact data into a special data structure (*the FP-tree*) making it easier to mine frequent patterns. *FP-growth* algorithm recognizes the shortest frequent patterns (frequent pattern 1-itemsets) and then “grows” them to

longer ones instead of generating and testing candidates. Owing to this graceful mining approach, *FP-growth* algorithm reduces the mining time significantly.

Conventional frequent patterns mining algorithms require users to provide a support threshold, which is very difficult to identify without knowledge of the dataset in advance. A large minimum support threshold results in a small set of frequent patterns which users may not discover any useful information. On the other hand, if the support threshold is small, users may not be able to screen the actual useful information from a huge resulted frequent pattern set.

Recent years, various researches have been dedicated to mine frequent patterns based upon user-friendly concepts such as maximal pattern mining [3][4], closed pattern mining [5][6], and mining the most interesting frequent patterns (*top-k mining*) [7][8][9][10][11]. With regard to usability and user-friendliness, *top-k* mining permits users to mine the *k-most* interesting frequent patterns without providing a support threshold. In real applications, users may need only the *k-most* interesting frequent patterns to examine for the useful information. If they fail to discover useful knowledge they can continue to mine the next *top-k* frequent patterns and so on.

The difficulty in mining *top-k* frequent patterns is that the minimum support threshold is not given in advance. The support threshold is initially set to 0. The algorithms have to gradually find out the support threshold to prune the search space. A good algorithm is the one that can raise the support value (i.e. from 0) to the actual value effectively and efficiently.

As our best knowledge, the *top-k* mining was introduced first in [9] which extended the *Apriori* approach to find out the *k-most* interesting frequent patterns. Fu, A.W., et al., introduced the *Itemset-Loop/Itemset-iLoop* algorithms [7] to mine every *n-most* interesting patterns in each set of *k-itemsets*. Even these methods apply some optimizations they suffer the same disadvantage as that in *Apriori* approach. *Top-k FP-growth* [11] is an *FP-growth* extension method to mine *top-k* frequent patterns with the adoption of a reduction array to raise the support value. However, *Top-k FP-growth* algorithm is an exhaustive approach and it raises the support value slowly. Beside that, this method has a problem of the effectiveness. It presents exactly *k* (for *top-k*) frequent patterns where *k* is a user specified number. However, with a user specified number, say *k*, a dataset may contain more than *k-most* interesting frequent patterns since some patterns may have the same support.

This research proposes a new algorithm, called *ExMiner*, to mine *top-k* frequent patterns effectively and efficiently. The mining task in this algorithm is divided into two phases – the “*explorative mining*” and the “*actual mining*” phases. The explorative mining phase is performed first to recognize the optimal internal support threshold according to a given number of *top-k*. This value is provided as a support threshold parameter for the actual mining phase. With an optimal internal support threshold, the actual mining phase can mine *top-k* frequent patterns efficiently. The *ExMiner* is then combined with the idea of “build once mine anytime” to mine *top-k* frequent patterns sequentially. This approach is valuable in real applications.

The rest of the paper is organized as follows. Section 2 is the preliminary definitions. The explorative mining algorithm, *ExMiner*, is described in section 3. The idea of mining *top-k* frequent patterns sequentially is explained in section 4. Section 5 is the experimental evaluation and we conclude the paper in section 6.

2 Preliminary Definitions

Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of items. An itemset, X , is a non-empty subset of I , called a pattern. A set of k items is called a k -itemset. A transaction T is a tuple $\langle T_{id}, X \rangle$, where X is an itemset and T_{id} is the identifier. A transactional database D is a set of transactions.

Definition 1. The support of an itemset X , denoted as $Sup(X)$ is the number¹ of transactions in D that contain X .

Definition 2. Let θ be the minimum support threshold. An itemset X is called a frequent itemset or a frequent pattern if $Sup(X)$ is greater than or equal to θ .

Definition 3. Let α be the support of the k^{th} pattern in a set of frequent patterns which are sorted by the descending order of their supports. The k -most interesting frequent patterns is a set of patterns whose support value is not smaller than α .

Observation 1. A set of k -most interesting frequent patterns may contain more or less than k patterns because some patterns may have the same support value or the dataset is too small to generate enough k patterns.

3 ExMiner Algorithm

In the traditional frequent pattern mining, the key parameter is the minimum support threshold. With a given minimum support threshold, algorithms can prune the search space efficiently to reduce the computation time. In contrast, *top-k* mining algorithms are not provided such a useful parameter but have to identify that value automatically according a given number *top-k*. The better *top-k* mining algorithms are those that can identify support threshold not only faster but also in a more effective and efficient way than others do.

Our purpose is to propose an algorithm that has the ability to recognize the final internal support threshold which is used directly to generate only the actual *top-k* frequent patterns (without candidate generation). This ability avoids the additional computation time in the mining phase significantly. Owing to that idea an algorithm called *ExMiner* is proposed.

ExMiner algorithm proceeds from the observation of mineral mining activities in the real world in which some explorative mining activities should be performed before the actual mining. The term *ExMiner* stands for the term “explorative miner”. *ExMiner* extends the *FP-growth* to mine *top-k* frequent patterns effectively and efficiently with following 2 points: a) setting the internal threshold *border_sup*, b) taking an explorative mining to recognize an effective “*final internal support threshold*” which is used in the actual mining phase for recognizing frequent patterns.

Setting *border_sup*: The *ExMiner* scans the dataset once to count the supports of all items and sort them by the descending order of their support. The *border_sup* is set to the support value of the k^{th} item in the sorted list. The list of *top-k* frequent items,

¹ The *support* can be defined as a relative value, that is the fraction of the occurrence frequency per the total number of transactions in the considering dataset.

denoted as *F-list*, is a list of items whose support is not smaller than *border_sup*. *ExMiner* algorithm also takes the second scan on the dataset to construct an *FP-tree* according to the *F-list*.

Explorative mining: Instead of mining *top-k* frequent patterns immediately after the *FP-tree* has been built, *ExMiner* performs a virtually explorative mining first. The purpose of this virtual mining routine, called *VirtualGrowth*, is to identify the *final internal support threshold*.

The pseudo code of *ExMiner* algorithm is described in Figure 1.

<p>Input: Dataset <i>D</i>, number of patterns <i>k</i></p> <p>Output: <i>top-k</i> frequent patterns</p> <p>Method:</p> <ol style="list-style-type: none"> 1. Scan <i>D</i> to count support of all 1-itemsets 2. According to <i>k</i>, set <i>border_sup</i> and generate <i>F-list</i>. Insert the support values of the first <i>k</i> items in <i>F-list</i> into a queue, say <i>supQueue</i> 3. Construct an <i>FP-tree</i> according to <i>F-list</i> 4. Call <i>VirtualGrowth</i>(<i>multiset</i><<i>int</i>>* <i>supQueue</i>, <i>FP-tree</i>, <i>null</i>) to explore the <i>FP-tree</i> and set the final internal support threshold θ to the smallest element, min_q, of <i>supQueue</i> 5. Mine the <i>FP-tree</i> with support threshold θ to output <i>top-k</i> frequent patterns
--

Fig. 1. Pseudo code of the *ExMiner* algorithm

The pseudo code of the *VirtualGrowth* routine, in step 4 of the *ExMiner* algorithm, is depicted in Figure 2 and described as bellow.

Initially *supQueue* contains the first *k* items in the *F-list* (1-itemsets) and sorted by the descending order of their values. If the tree is a single-path tree, the routine examines each node in the tree in the top-down direction (line 1). While the support of considering nodes are greater than the last element in *supQueue*, say min_q (line 2), it figures out the number of potential patterns (accompanied with their support values) which can be generated from a considering node (lines 3 and 4). The condition $\alpha \neq null$ is checked to avoid duplicating the support values of 1-itemsets into *supQueue*. Note that, for a node, say *d*, at a position *i* ($i = 1$ for the first “real” node under Root) there are 2^{i-1} combinations of *d* with other nodes in the path. If α is not *null*, it serves as a real prefix for each of the above combination making all 2^{i-1} patterns to be long ones (i.e. contains more than 1 item). Lines 5 and 6 try to update the *c* recognized supports (i.e. δ) to the last *c* elements in *supQueue* (smallest *c* values). If the tree is not a single-path tree (line 7), the routine traverses the tree from the top of the header table. Let a_i be the considering item. While $sup(a_i)$ is greater than min_q (line 8), the supports of potential patterns generated by “growing” this item will be satisfied to replace min_q . Line 9 is to replace min_q by the support of a potential long pattern. A new prefix β and its conditional *FP-tree*, $Tree_\beta$, are created (lines 10 and 11). If this conditional *FP-tree* is not empty, the routine is recalled recursively to deal

with the new conditional FP-tree (line 12). When the routine terminates, *supQueue* is converged and the last element, min_q , serves as the *final internal support threshold* for mining *top-k* frequent patterns.

```

Procedure VirtualGrowth(supQueue, Tree,  $\alpha$ ) {
(1) If Tree contains a single path P {
     $i = 1$ ;  $\delta = \text{sup}(n_i)$ ;  $min_q \leftarrow \text{supQueue.last}$ ;
    //  $n_1$  represents for the 1st "real" node under Root
(2) While ( $\delta > min_q$ ) {
(3)     If ( $\alpha \neq \text{null}$ )  $c = 2^{i-1}$ ;
(4)     Else  $c = 2^{i-1} - 1$ ;
(5)     For each of c value of  $\delta$ 
(6)         If ( $\delta > min_q$ ) replace  $min_q$  in supQueue by  $\delta$ ;
            // After replacing  $min_q$  is updated
             $i++$ ;
            If ( $n_i \neq \text{null}$ )  $\delta = \text{sup}(n_i)$  Else  $\delta = 0$ ;
        }
(7) Else
(8)     While ( $\text{sup}(a_i) > min_q$ ) { //  $a_i$  in the header table
(9)         If ( $\alpha \neq \text{null}$ ) replace  $min_q$  by  $\text{sup}(a_i)$ ;
(10)         $\beta = a_i \cup \alpha$ ;  $a_i \leftarrow$  next item in header the table;
(11)        Construct  $\beta$ 's conditional FP-tree, Tree $\beta$ ;
            // based on its conditional pattern base
(12)        If (Tree $\beta$   $\neq$  empty)
            call VirtualGrowth(supQueue, Tree $\beta$ ,  $\beta$ );
        }
    }
}

```

Fig. 2. Pseudo code of the *VirtualGrowth* routine

An example of using *ExMiner* to mine *top-7* frequent patterns from a dataset in Table1 is illustrated in Figure 3 and described as follows.

Table 1. A sample transaction dataset

T _{id}	Items	Sorted frequent items
100	f, a, c, g, m, p	f, c, a, m, p
200	a, b, c, f, l, m, o	f, c, a, b, m
300	b, f, j, o	f, b
400	b, c, k, p	c, b, p
500	a, f, c, l, p, m, n	f, c, a, m, p

After the first scan on the dataset, the list of all *I*-itemsets sorted by the descending order of their support is $\langle f:4, c:4, a:3, b:3, m:3, p:3, l:2, o:2, g:1, j:1, k:1 \rangle$. The *border_{sup}* is set to 2, (i.e. the support value of the 7th element, $\langle l:2 \rangle$), and the *F*-list is $\langle f:4, c:4, a:3, b:3, m:3, p:3, l:2, o:2 \rangle$ (*|F-list|* can be greater than 7). An *FP-tree* is

built containing all items in this list. A *supQueue* with 7 elements is initially filled by the support values of the first 7 elements in *F-list*: *supQueue* is $\langle 4, 4, 3, 3, 3, 3, 2 \rangle$ ($min_q=2$). The top item in the header table, *f*, is achieved by the *VirtualGrowth* routine. Since $sup(f) = 4$, greater than min_q , the algorithm traverses the tree to recognize the conditional pattern based of *f*. Unfortunately, this conditional pattern base is empty, no other potential patterns will be generated except the 1-itemset pattern $\langle f:4 \rangle$. Therefore *supQueue* is not changed and min_q is remained to 2. The next item, *c*, is reached resulting in two potential patterns $\langle c:4 \rangle$ and $\langle c.f:3 \rangle$. Since the second pattern is longer than 1 and its support is 3, greater than min_q , *supQueue* is updated to $\langle 4, 4, 3, 3, 3, 3, 3 \rangle$ ($min_q=3$). Next, item *a* is reached, but its support value is 3, not greater than min_q , the algorithm terminates. The current min_q (i.e. 3) is the *final internal support threshold* used to mine *top-7* patterns in the actual mining phase, where the *FP-growth* algorithm [2] is invoked.

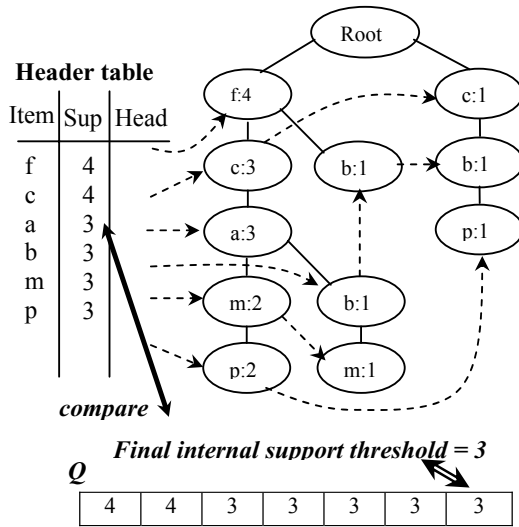


Fig. 3. Example of a VirtualGrowth routine

4 Mining K-Most Interesting Frequent Patterns Sequentially

To mine *top-k* frequent patterns correctly, *ExMiner* requires an *FP-tree* containing at least *k* single items. However, a set of *k* items can produce $2^k - 1$ patterns, from which only the *k*-most interesting patterns (*k* highest frequency patterns) are chosen. Obviously, it is not efficient to construct such a large tree, examine it for just a moderate number of *top-k* frequent patterns. In practice, especially in dense datasets which contain long patterns, just a very small number of items (compared to *k*) are adequate to generate *top-k* frequent patterns. For example, in a real dataset, named *Connect-4*², only the first 15 items can generate *top-1000* frequent patterns and only

² Available at UCI Machine Learning Repository (<http://www.ics.uci.edu/~mlern/MLRepository.html>).

the first 19 items can generate *top-10,000* frequent patterns. The smaller number of items an algorithm examines, the more efficient the algorithm is. However, the problem is how to ensure such a small number of items adequate for mining *top-k* frequent patterns correctly. In fact, it is very difficult to prove theoretically the correctness of mining *top-k* frequent patterns by considering less than k items since the problem varies from datasets to datasets. The idea of mining *k-most* interesting frequent patterns sequentially, or *seq-Miner* for short, can solve this problem soundly.

The idea of mining *k-most* interesting frequent patterns sequentially consists of two major advantages. 1) It releases users from declaring either a conventional minimum support threshold or a number of *top-k*. The program proposes users the highest n_c ³ frequency patterns first. If users are not satisfied with the result, next *top- n_c* frequent patterns (i.e. *top- n_c+1* to *2 n_c*) are proposed. Users can stop the program when the interesting information is found or whenever they want. 2) After finding the first *top- n_c* frequent patterns, algorithm can identify the number of items that adequate for mining the next *top- n_c* frequent patterns correctly and efficiently. For example, if the algorithm starts by mining *top-1000* frequent patterns, the algorithm requires at least 1000 items at the first time. However, after mining the first *top-1000* frequent patterns it recognized that only 15 items, for example, were examined. From this point, the algorithm can successfully consider only 1015 items ($15+1000$) for mining *top-1001* to 2000 frequent patterns and so on. In contrast, conventional algorithms have to consider exactly k items to mine *top-k* frequent patterns at any iteration. Since the number of single items considered at any iteration in *seq-Miner* is smaller than the number of desired *top- n_c* patterns, the performance is improved significantly. The pseudo code of the *seq-Miner* is depicted in Figure 4.

<p>Input: Dataset D, interval n_c Output: <i>top-k</i> frequent patterns sequentially Method: 1. Set $k = n_c$ (1000 by default); $considerItems = n_c$ 2. call $ExMiner(D, k, int\ considerItems, int*\ examinedItems, int*\ minSup)$ 3. $considerItems = n_c + examinedItem$; $k+ = n_c$; 4. If continue, goto 2; else stop</p>

Fig. 4. Pseudo code of the *seq-Miner* algorithm

Note that, the *ExMiner* (previously described in Figure 1) has been modified a little in which three more parameters, *considerItems*, *examinedItems*, and *minSup*, are provided. *ConsiderItems* holds the number of items that the algorithm has to consider before examining. *ExaminedItems* holds the number of items that have really been examined. This value is used to update the number of items that need to be considered at the next iteration (at step 3). *MinSup* holds the support of the last pattern in the current set of *top-k* frequent patterns. This value helps to propose only *top- $k+1$* to *$k+n_c$* , instead of proposing *top- $k+n_c$* (from 1^{st} to $k+n_c^{th}$) patterns, at the next iteration.

³ n_c is a chunk size which can be set by users or set to 1000 automatically by the algorithm.

Using *ExMiner*, or *seq-Miner* algorithm to mine *top-k* frequent patterns, a new *FP-tree* has to be rebuilt from the scratch whenever a given *top-k* changed. If a “large” *FP-tree* is built in advance to be re-used for mining *top-k* frequent patterns with any different value of *top-k*, the performance can be improved significantly. The idea of “build once mine anytime” (*BOMA*) allows us to do that. Moreover, when the *seq-Miner* algorithm is combined with the *BOMA* idea, called *seq-BOMA* approach, the performance is increased surprisingly. This approach is described as following.

A “large” *FP-tree* is built and its information is saved into the hard disk. This information will be read to reconstruct the original “large” *FP-tree* (in the memory) when a *top-k* mining is required. After handling the original *FP-tree*, the *seq-Miner* algorithm can be applied to mine for any *top-k* frequent patterns sequentially.

In the *seq-BOMA* approach, the major computation time is the time to build the “large” *FP-tree*. Nevertheless, this tree can be built at the computer-free time that makes the approach more practical in real applications. There is a subtle difference between this idea and the opinion of mining all patterns “once” in advance and then extracting the best *k* ones according to a user requirement. The latter one is trivial and impossible for medium and large dense datasets. On the other hand, the former one is suitable for the human feeling. One may feel that one can analyze about a thousand patterns each time and be able to keep this work in about ten times, for example. In this case a *large* *FP-tree* with 10,000 single items will be built according to the *seq-BOMA* approach. Therefore *seq-BOMA* is useful in practical applications.

Another advantage of this approach is that an original “large” *FP-tree* of *p* items may contain much more than *top-p* frequent patterns and *seq-BOMA* is able to extract all of them. This means *seq-BOMA* can mine for many different *top-k* frequent patterns sequentially where *k* is much greater than *p*. This characteristic can not be found in any traditional *top-k* frequent patterns mining approach. For example, with an *FP-tree* containing 10,000 items created from the dataset *D: T10I4D2000kN1000k*, *seq-BOMA* can mine for *top-50,000* frequent patterns. On the other hand, *ExMiner* can not execute even to mine *top-15,000* frequent patterns in our computer because of memory overflow when an *FP-tree* of 15,000 items is built.

5 Experimental Evaluations

In this section, the experiments for the proposed algorithms *ExMiner*, and *seq-BOMA* on their efficiency, effectiveness and scalability are presented. The IBM quest synthetic data generation code [12] is used to create synthetic datasets. The experiments also been taken place on a real dataset named *connect-4*. All the experiments were performed on a 3.2GHz Pentium 4 PC with 1 GB main memory running on Window XP platform and MS. Visual C++ 6.0 environment.

a. Efficiency Evaluation

In order to evaluate the efficiency of *ExMiner* and *seq-BOMA*, their overhead is compared to the computation time of the “optimal *FP-growth*” and “optimal *Apriori*” algorithms. The “optimal *FP-growth*” is the *FP-growth* algorithm with the *best tuned* minimum support threshold for a desired number of *top-k* patterns (“optimal *Apriori*” is defined as the same way). The *best tuned* (or *optimal*) support threshold can be obtained when we mine for a desired *top-k* frequent pattern using *ExMiner* algorithm.

The *Top-k FP-growth* [11] method has been implemented on our machine to compare with our methods.

Figure 5 describes the comparison between algorithms running on dataset D_1 : **T10I4D1000kN1000k**, created by IBM data generation code [12]. In this experiment we tried to mine *top-k* frequent patterns where k varied from 1000 to 7000 with an interval of 1000. The figure shows that *ExMiner* is superior to the *TopK-FPGrow* and faster than *optimal Apriori* with a factor of 2. Another interesting thing is that *seq-BOMA* is even better than the *optimal FP-growth*, the algorithm considered the ideal one. To execute the *seq-BOMA* algorithm, a “large” *FP-tree* containing 7000 items (can be used to mine *top-7000* safely) is built in advance at the computer-free time. Since the time to build this *FP-tree* is only 282(s), even that time is taken into account *seq-BOMA* still remains its efficiency and its strong power in general.

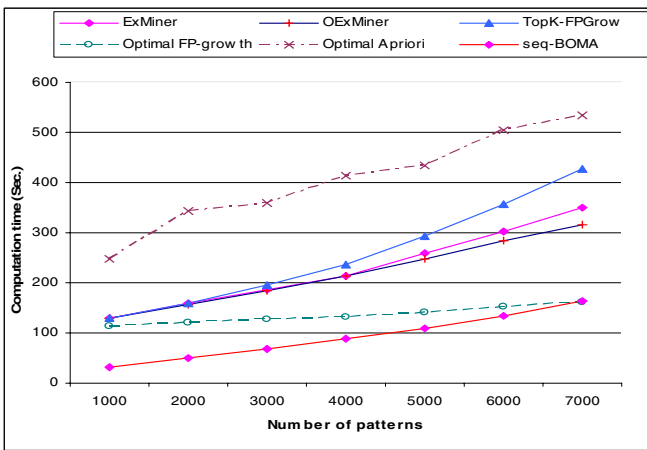


Fig. 5. Time comparison on D_1

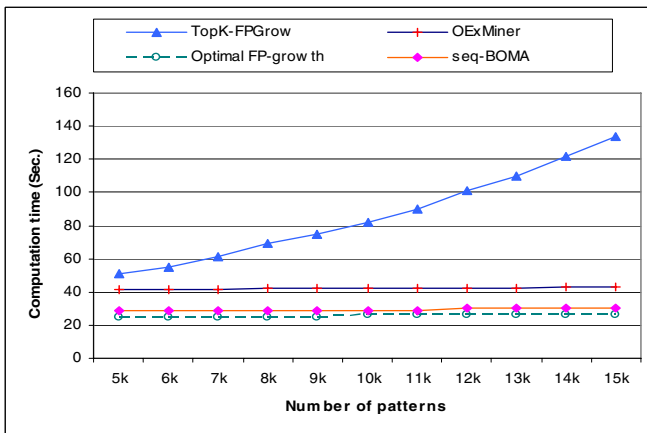


Fig. 6. Time comparison on connect-4 of the algorithms excluding Optimal Apriori

The experiment on a real dataset named *connect-4* (67,557 transactions with 43 items in each one) is shown in Figure 6. Since *connect-4* is a dense dataset containing very long patterns, *Apriori* becomes an extremely costly algorithm that is incomparable to the remaining algorithms. Figure 6 excludes the *optimal Apriori*. The figure shows that *ExMiner* is about 3 times faster than *TopK-FPGrow*; *Seq-BOMA* is about 7 times faster than *TopK-FPGrow* and almost equal to the *Optimal FP-growth*.

b. Effectiveness Evaluation

Assume that θ is the optimal minimum support threshold used to mine *top-k* frequent patterns. If users do not handle any *top-k* frequent pattern mining mechanism, they have to use traditional algorithms such as *Apriori* or *FP-growth* with guessing minimum support thresholds. Let $mins_i = i * \theta$ be the user guessing threshold, where $i = 0.90, 0.95, 0.97$, etc... For example, if $\theta = 0.3\%$ then $mins_{0.97} = 0.97 * 0.3\% = 0.291\%$.

Table 2 shows the total number of patterns generated by *FP-growth* algorithm executed on D_1 : **T10I4D1000kN1000k** with guessing minimum support thresholds. The table shows that even the guessing support is very close to the optimal one, the number of obtained patterns is quite different to the desired top-k frequent patterns. For example, *ExMiner* presents **5007** patterns when users mine for *top-5000* frequent patterns while the optimal *FP-growth* with $mins_{0.95}$ reveals **6061** patterns. However, guessing for such a close number (about 95%) to the optimal one is not an easy job.

Table 2. Number of patterns with guessed thresholds from D_1

Top-k	ExMiner	FP-growth		
		Mins _{0.9}	Mins _{0.95}	Mins _{0.97}
1000	1002	1896	1408	1190
2000	2004	3866	2636	2035
3000	3001	5139	4162	3829
4000	4005	5980	4874	4495
5000	5007	7231	6061	5656

c. Scalability Evaluation

To evaluate the scalability of the proposed methods we use another synthetic dataset, named D_2 : **T10I4D2000kN1000k**, which contains 2 million transactions. As shown in Figure 7, *ExMiner* is still better than *TopK-FPGrow*, and the *seq-BOMA* is clearly faster than *optimal FP-growth*. This result reveals that our proposed methods scale well in large datasets.

In real applications, many datasets such as customer transactional datasets of super markets or electronic shops contain many duplicating transactions (identical transactions). To experiment how well our methods execute on those kinds of dataset we duplicate the dataset *connect-4* with some scale up factors and test on those scaled up datasets.

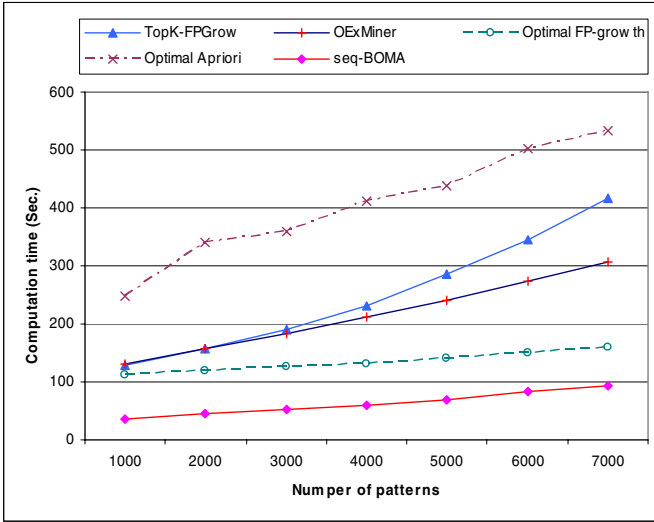


Fig. 7. Time comparison on D_2

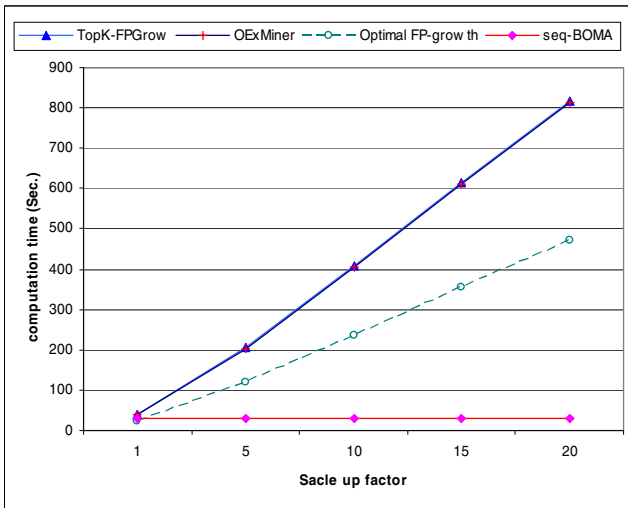


Fig. 8. Time comparison on scaled up connect-4

Figure 8 shows the experimental results where number of desired *top-k* patterns is fixed to 10,000. According to the figure, the computation time of *seq-BOMA* is constant at the value of 29 or 30 seconds while the computation time of three remaining algorithms increases very fast. Another interesting feature is that when the scale up factor is 20, *seq-BOMA* is faster than *optimal FP-growth* 17 times. The reason that the computation time in *seq-BOMA* keeps in a constant is that when the dataset is duplicated, the structure and the size (the number of nodes) of the “large” *FP-tree* which

was previously built does not change. Therefore the time of reading, reconstructing and mining the tree for *top-k* frequent patterns in *seq-BOMA* does not change as well. The result of this experiment reveals that *seq-BOMA* can be worthily applied to mine *top-k* frequent patterns in such datasets that contain many duplicating transactions.

6 Conclusions and Future Work

This research proposed a new algorithm for mining *top-k* frequent patterns efficiently and effectively, called *ExMiner* algorithm. The *seq-BOMA* approach, the combination between *seq-ExMiner* algorithm (to mine *top-k* frequent patterns sequentially) and the idea of “build once mine anytime” proposes many beneficial features for the real applications. It allows users to mine *top-k* frequent patterns sequentially without worry about giving a minimum support threshold or a value of *top-k*. The experimental evaluations on both synthetic and real datasets revealed that our proposed methods are superior to the existing ones and scaled well on large datasets.

The proposed methods, especially *seq-BOMA*, propose a great possibility of being applied into real applications. We are planning to investigate more for applying these approaches to the real works.

References

1. Agrawal, R., and Srikant, R.: Fast algorithm for mining association rules. In proc. of VLDB '94, Santiago, Chile (1994) 487-499
2. Han, J., Pei, J., and Yin, Y.: Mining frequent patterns without candidate generation. In proc. of ACM SIGMOD Conference on Management of Data (2000) 1-12
3. Bayard, R.J.: Efficiently mining long patterns from databases. In proc. of ACM SIGMOD Conference on Management of Data (1998) pp. 85-93
4. Grahne, G., and Zhu, J.: High performance mining of maxima frequent itemsets. In proc. of SIAM'03 workshop on High Performance Data Mining (2003)
5. Grahne, G., and Zhu, J.: Efficiently using prefix-tree in mining frequent itemsets. In proc. of IEEE ICDM workshop on Frequent Itemsets Mining Implementations (2003)
6. Pei, J., Han, J., and Mao, R.: CLOSET: An efficient algorithm from mining frequent closed itemsets. In proc. of DMKD'00 (2000)
7. Fu, A.W., Kwong, R.W., Tang, J.: Mining N most interesting itemsets. In proc. of ISMIS'00 (2000)
8. Han, J., Wang, J., Lu, Y. and Tzvetkov, P.: Mining top-k frequent closed patterns without minimum support. In proc. of IEEE ICDM Conference on Data Mining (2002)
9. Ly, S., Hong, S., Paul, P., and Rodney, T.: Finding the N largest itemsets. In Proc. Int. Conf. on Data Mining, Rio de Janeiro, Brazil (1998) 211-222
10. Wang, J., Han, J., Lu, Y. and Tzvetkov, P.: TFP: An efficient algorithm for mining top-k frequent closed itemsets. In proc. of IEEE Knowledge and Data Engineering, vol 17, no.5 (2005) 652-663
11. Hirate, Y., Iwahashi, E., and Yamana, H.: TF2P-growth: An efficient algorithm for mining frequent patterns without any thresholds. In proc. of ICDM (2004)
12. IBM Quest Data Mining Project. Quest synthetic data generation <http://almaden.ibm.com/software/quest/Resources/index.shtml>

Learning Bayesian Networks Structure with Continuous Variables

Shuang-Cheng Wang^{1,2}, Xiao-Lin Li³, and Hai-Yan Tang²

¹Department of Information Science
Shanghai Lixin University of Commerce, Shanghai 201620, China

²China Lixin Risk Management Research Institute
Shanghai Lixin University of Commerce, Shanghai 201620, China
{wangsc, tlx}@lixin.edu.cn

³National Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
lixl@lamda.nju.edu.cn

Abstract. In this paper, a new method for learning Bayesian networks structure with continuous variables is proposed. The continuous variables are discretized based on hybrid data clustering. The discrete values of a continuous variable are obtained by using father node structure and Gibbs sampling. Optimal dimension of discretized continuous variable is found by MDL principle to the Markov blanket. Dependent relationship is refined by optimization regulation to Bayesian network structure in iteration learning.

1 Introduction

Bayesian networks are graphical representations of dependency relationships between variables. They are intuitive representations of knowledge and are akin to human reasoning paradigms. They are powerful tools to deal with uncertainties, and have been extensively used to uncertainty knowledge representation, inference and reasoning. In the past decades, they have been successfully applied in medical diagnose, software intelligence, finance risk analysis, DNA functional analysis, web mining and so on, and have become a rapidly growing field of research and have seen a great deal of activity.

A Bayesian network consists of two components: structure and parameters. They can respectively be used in qualitative and quantitative causal analysis. Bayesian network learning includes two parts, one for structure learning and one for parameters learning. It is a trivial process to learn parameters given structure and complete data, so the key attention for learning Bayesian networks has been focused on structure learning. A lot of researches have been given in this line [1][2][3][4][5], however, most of them are focused on discrete variables. Researches on learning Bayesian network with continuous variables are rare. In practice, most variables are continuous ones, and finding an efficient method to deal with continuous variables is a key issue to put Bayesian networks to a more extensive application. There are two ways to deal with continuous variables. One is taking some distribution for the continuous variable [6][7][8][9] (Normal distribution in general), and the scoring is approximated in the

search process. As the distribution assumption for continuous variables are somewhat strong, the applications of this method are limited. Another method is to discretize the continuous variables [10][11][12], and convert the learning process from one based on hybrid data to one based on discrete data. The latter method makes no assumption about the variable distribution space, and can be applied to more general cases efficiently. As structure is a quantitative representation of dependency relationship, it is enough to retain only the information relating the dependent relations between variables and neglect some details.

At present, discretization methods for learning Bayesian networks with continuous variable are mainly based on extended entropy discretization, a generalization of method proposed by Fayyad and Irani [10]. The process is an iteration one, in which the conditional entropy conditioned on the parent nodes are used to score, greedy or random searches are conducted on every possible discretization point, MDL [2] (minimal description length) principle or conditional entropy are used to determine the number of discretization point, and the discretized data are used for the next iteration. As the discretization point space increase is in exponential to the data size, few scoring-search methods are tracable for large dataset. In addition, as learning Bayesian networks from data is NP-hard [13], and both the computational complexity of the scoring function and the size of searching space increase exponentially with the additional variables, it is a general case to require a node ordering before the learning process. So the efficiency of the structure relearning process is low, and the process is apt to be trapped in the local maxima.

In this paper, a novel method for learning Bayesian network with continuous variables is proposed. In the method, the continuous variables are discretized by hybrid data clustering that makes full use of the dependent information between variables; the joint distribution is decomposed according to the Bayesian network structure, and the exponential sampling complexity of standard Gibbs sampling [14][15] in the variables number is settled by this method; the Bayesian network structure is improved by dependent optimization regulation continuously to convergence. Problems coming along with the extended entropy discretization can be avoided.

Throughout the paper, Let X_1, \dots, X_n denote discrete or continuous random variables, x_1, \dots, x_n be their instantiation. D represents a hybrid dataset with N cases, all data being generated from some probability distribution P randomly.

2 Initial Discretization of Continuous Variables and Initialization of Bayesian Network Structure

Initial discretization of continuous variables is based on dichotomy. The dataset after discretization is the initial dataset, denoted by $D^{(0)}$. As the dependency relations underlying $D^{(0)}$ may be perplexing, learning Bayesian networks directly from $D^{(0)}$ may lead to many lost or redundant edges, which will severely affect the convergence speed. In our method, we propose to use maximal likelihood tree to retain the important dependency relationship between variables. The maximal likelihood tree can be easily constructed, but it doesn't encode causal information. We use causal semantics

to determine edge direction for the tree, and the resulting directed acyclic graph is viewed as the initial Bayesian network structure.

At the beginning, a maximal likelihood tree T is induced from dataset $D^{(0)}$ [3]. Some edge directions are determined according to Pearl’s Poly-tree method [3], and a chain graph [16] is obtained. Let e_1, \dots, e_s be the edges in the chain graph with unsettled direction. To determine the edge directions for those edges, let G_C^{i+} and G_C^{i-} be the chain graphs for edge e_i with different edge direction, where direction for edges e_1, \dots, e_{i-1} is settled and direction for edges e_{i+1}, \dots, e_s is unsettled. MDL scoring for each chain graph can be calculated according to the joint probability decomposition given by Buntine[16]. Direction for edge e_i then can be determined by comparing $MDL(G_C^{i+} | D)$ and $MDL(G_C^{i-} | D)$. After direction for each edge is determined, a directed acyclic graph $G^{(0)}$ is obtained. Let $G^{(0)}$ be the initial Bayesian network. And the node ordering is determined by $G^{(0)}$ is $X_1^{(0)}, \dots, X_n^{(0)}$.

3 Discretizing Continuous Variables and Regulating Bayesian Networks Structure Iteratively

The learning process is an iteration in essence, in which two sequences are built, one for discretized dataset $\{D^{(k)}\}$ and another for structure $\{G^{(k)}\}$. Each iteration consists two parts. The first part is to get a new discretized dataset $D^{(k)}$ by discretizing the original dataset based on the resulted Bayesian networks structure $G^{(k-1)}$ of the previous iteration. The second part is to regulate the Bayesian network structure based on $D^{(k)}$ and obtain $G^{(k)}$. The iteration process continued until stopping criteria are satisfied. The process is illustrated in Fig.1.

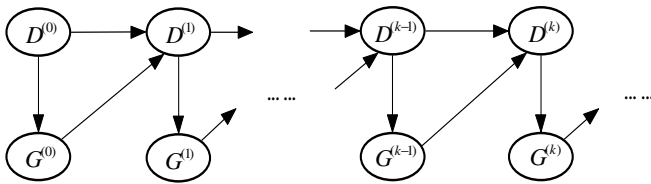


Fig. 1. Iteration process for learning Bayesian network from hybrid data

3.1 Discretizing Continuous Variables

In every iteration, every continuous variable is discretized sequentially according to the order determined by the Bayesian networks structure from the previous iteration. The method of hybrid data clustering is used to discretize a continuous variable, including how to determine the cluster number and to assign values to the corresponding discrete variable.

In the discretizing process, the Bayesian networks structure is fixed, but the dataset are updated. A discretized dataset subsequence $D_0^{(k)} = D^{(k)}, D_1^{(k)}, \dots, D_l^{(k)} = D^{(k+1)}$ is produced, where $D_j^{(k)}$ is a dataset obtained by discretizing variable X_j in dataset $D_{j-1}^{(k)}$.

(1) Discretizing Continuous Variable based on Clustering

To discretize a continuous variable, the influence of the node parents should be taken into consideration so as to retain the dependent relationship between variables after discretizing. Let $X_1^{(k)}, \dots, X_n^{(k)}$ be the node ordering determined by $G^{(k)}$.

Let X_i replace $X_i^{(k)}$ and the derived variable after discretizing X_i is still denoted by $X_i^{(k)}$. The detail of the discretization has been given below.

According to Bayes theorem:

$$p(x_i^{(k)} | x_1^{(k)}, \dots, x_{i-1}^{(k)}, x_i, G^{(k)}) = \alpha p(x_i^{(k)} | \pi_{x_i^{(k)}}(x_i^{(k)}), G^{(k)}) = \beta p(x_i | \pi_{x_i^{(k)}}(x_i^{(k)}), G^{(k)}) p(x_i^{(k)} | \pi_{x_i^{(k)}}(x_i^{(k)}), G^{(k)}) \quad (1)$$

where $\pi_{x_i^{(k)}}$ is the configuration of the node parents set $X_i^{(k)}$ in $G^{(k)}$, α and β are numbers irrelevant to $x_i^{(k)}$.

Discretize X_i according to $G^{(k)}$ and Gibbs sampling, replace value for $X_i^{(k)}$ in $D_i^{(k)}$ with the discretized value, and obtain dataset $D_{i+1}^{(k)}$. In Equation 1, continuous variable takes conditional normal density function. Of course, other density function such as multinomial or kernel density functions can also be used.

$$p(x_i | \pi_{x_i^{(k)}}(x_i^{(k)}), x_i^{(k)}, G^{(k)}) = g(x_i; \mu_i(\pi_{x_i^{(k)}}(x_i^{(k)}), x_i^{(k)}); \sigma_i(\pi_{x_i^{(k)}}(x_i^{(k)}), x_i^{(k)}) | G^{(k)}) = \frac{1}{\sqrt{2\pi\sigma_i(\pi_{x_i^{(k)}}(x_i^{(k)}), x_i^{(k)})}} e^{-\frac{(x_i - \mu_i(\pi_{x_i^{(k)}}(x_i^{(k)}), x_i^{(k)}))^2}{2\sigma_i^2(\pi_{x_i^{(k)}}(x_i^{(k)}), x_i^{(k)})}} \quad (2)$$

where $\mu_i(\pi_{x_i^{(k)}}(x_i^{(k)}), x_i^{(k)})$ and $\sigma_i(\pi_{x_i^{(k)}}(x_i^{(k)}), x_i^{(k)})$ are conditional mean and standard deviation of X_i respectively.

If $p(x_i^{(k)} | \pi_{x_i^{(k)}}(x_i^{(k)}), G^{(k)}) = 0$, then $p(x_i^{(k)} | \pi_{x_i^{(k)}}(x_i^{(k)}), G^{(k)})$ should be Laplace-corrected [17], $p(x_i^{(k)} | \pi_{x_i^{(k)}}(x_i^{(k)}), G^{(k)}) = (1/N) \left(N(\pi_{x_i^{(k)}}(x_i^{(k)})) + N(x_i^{(k)}) (1/N) \right)$, where $N(\pi_{x_i^{(k)}}(x_i^{(k)}))$ is the case number of the $X_i^{(k)}$'s parent node set $\Pi_{X_i^{(k)}}$ with configuration of $\pi_{x_i^{(k)}}(x_i^{(k)})$, $N(x_i^{(k)})$ is the case number for $X_i^{(k)} = x_i^{(k)}$.

For the chosen dimension $l(2 \leq l \leq M_i^{(k)})$, initialize $X_i^{(k)}$ randomly, and then revised the value for $X_i^{(k)}$ in $D_i^{(k)}$ by sampling until convergence.

Let $M_i^{(k)}$ be the maximal dimension of variable X_i after discretizing (normally it is set to 5 or 6), revised value for $X_i^{(k)}$ according to the record sequence in database.

Let $x_{im}^{(k)}$ be the m th value to be revised for $X_i^{(k)}$, $\hat{x}_{im}^{(k)}$ be value after revision, x_i^1, \dots, x_i^l be the possible values for $X_i^{(k)}$. After normalizing the sampling equation,

denote $w(h) = \frac{p(x_i | \pi_{x_i}, x_i^h, G^{(k)})p(x_i^h | \pi_{x_i}, G^{(k)})}{\sum_{j=1}^l p(x_i | \pi_{x_i}, x_i^j, G^{(k)})p(x_i^j | \pi_{x_i}, G^{(k)})}$, $h \in \{1, \dots, l\}$

For a random number λ , value for $X_i^{(k)}$ is:

$$\hat{x}_{im}^{(k)} = \begin{cases} x_i^1, & 0 < \lambda \leq w(1) \\ \dots\dots\dots \\ x_i^h, & \sum_{j=1}^{h-1} w(j) < \lambda \leq \sum_{j=1}^h w(j) \\ \dots\dots\dots \\ x_i^l, & \lambda > \sum_{j=1}^{l-1} w(j) \end{cases} \tag{3}$$

(2) Stopping Criterion for Discretization Iteration

Let $x_{i1}^{(k)}, x_{i2}^{(k)}, \dots, x_{iN}^{(k)}$ and $x_{i1}^{(k+1)}, x_{i2}^{(k+1)}, \dots, x_{iN}^{(k+1)}$ be value sequences for two consecu-

tive discretization, denote $sig(x_{ij}^{(k)}, x_{ij}^{(k+1)}) = \begin{cases} 0, & x_{ij}^{(k)} = x_{ij}^{(k+1)} \\ 1, & x_{ij}^{(k)} \neq x_{ij}^{(k+1)} \end{cases}, 1 \leq j \leq N$. If the consis-

tent degree of the two sequences $\frac{1}{N} \sum_{j=1}^N sig(x_{ij}^{(k)}, x_{ij}^{(k+1)}) \leq \eta_0$ are for the given thresh-
old $\eta_0 > 0$, then the iteration process is stopped.

(3) Determine the Optimal Discretization Strategy

Let $D_{i2}^{(k)}, \dots, D_{iM_i^{(k)}}^{(k)}$ be the discretized dataset sequence for variable X_i using different discretization strategy $\Lambda_h (2 \leq h \leq M_i^{(k)})$, an h corresponding to a different discreti-
zation strategy). Score every discretization strategy according to MDL principle. Let $h_0 = \min_{2 \leq h \leq M_i^{(k)}} \{MDL(\Lambda_h | D_{ih}^{(k)})\}$ be the discretization dimension.

$$MDL(\Lambda_h | D_{ih}^{(k)}) = \frac{\log N}{2} |\Lambda_h| - L(\Lambda_h | D_{ih}^{(k)}) \tag{4}$$

where $|\Lambda_h|$ be the parameters number of the Markov blanket of variable $X_i^{(k)}$ in $G^{(k)}$.

$$\begin{aligned}
L(\Lambda_h | D_{ih}^{(k)}) &= \sum_{i=1}^N \log(P(u_i | M_{X_i^{(k)}}, D_{ih}^{(k)}, \Lambda_h)) \\
&= N \sum_{x_i^{(k)}, \pi_{x_i^{(k)}}} p(x_i^{(k)}, \pi_{x_i^{(k)}} | M_{X_i^{(k)}}, D_{ih}^{(k)}, \Lambda_h) \prod_{x_i^{(k)} \in \pi_{x_j}} p(x_j, \pi_{x_j} | M_{X_i^{(k)}}, D_{ih}^{(k)}, \Lambda_h) \\
&\quad \cdot \log(p(x_i^{(k)} | \pi_{x_i^{(k)}}, M_{X_i^{(k)}}, D_{ih}^{(k)}, \Lambda_h) \prod_{x_i^{(k)} \in \pi_{x_j}} p(x_j | \pi_{x_j}, M_{X_i^{(k)}}, D_{ih}^{(k)}, \Lambda_h)).
\end{aligned}$$

3.2 Bayesian Networks Structure Regulation

To get a better Bayesian network structure, MDL principle and classic K2 algorithm [1] are utilized to regulate the structure. It has been noted for a given dataset and node ordering. If K2 algorithm can find the structure reliably, the optimization of Bayesian network structure is reduced to optimization of node ordering.

Let $X_1^{(k)}, \dots, X_n^{(k)}$ be the node ordering of $G^{(k)}$, $e_1^{(k)}, \dots, e_t^{(k)}$ be the sequence for directed edges, $G^{(k)}(e_j^{(k)})$ be a valid (no circle) Bayesian network corresponding to $G^{(k)}$ with the direction of edge $e_j^{(k)}$ reversed. Find $e_{k_1}^{(k)}$ in $e_1^{(k)}, \dots, e_t^{(k)}$ with the minimal MDL scoring and satisfying condition $MDL(G^{(k)}(e_j^{(k)}) | D^{(k+1)}) < MDL(G^{(k)} | D^{(k+1)})$, reverse $e_j^{(k)}$ in $G^{(k)}(e_{k_1}^{(k)})$ and get a new structure $G^{(k)}(e_{k_1}^{(k)}, e_j^{(k)})$. The regulation continues until there is no edge satisfying the condition stated above. After regulation, a better ordering is obtained, denoting as $X_{k_1}^{(k)}, \dots, X_{k_n}^{(k)}$ and the corresponding network as $G^{(k^*)}$.

According to $X_{k_1}^{(k)}, \dots, X_{k_n}^{(k)}$, K2 algorithm can find a new Bayesian network $G^{(k^{**})}$. If $MDL(G^{(k^{**})} | D^{(k+1)}) < MDL(G^{(k^*)} | D^{(k+1)})$, let $G^{(k+1)} = G^{(k^{**})}$, otherwise let $G^{(k+1)} = G^{(k^*)}$.

The ordering regulation method we have given is a deterministic greedy search. And other non-deterministic heuristic search methods, such as genetic algorithms, simulated annealing, can be used too.

3.3 Stopping Criterion for Structure Learning Iteration

According to a node ordering, let sequence $w^{(k)} = (a_{12}^{(k)}, \dots, a_{1n}^{(k)}, \dots, a_{i(i+1)}^{(k)}, \dots, a_{in}^{(k)}, \dots, a_{(n-1)n}^{(k)})$ represent the structure of $G^{(k)}$, where $a_{ij}^{(k)} = 1 (i < j)$ if there is an edge between X_i and X_j . Otherwise, $a_{ij} = 0$. Given positive threshold n_0 , if $\sum_{i < j} |a_{ij}^{(k+1)} - a_{ij}^{(k)}| < n_0$, iteration ends.

4 Experiments

In the experiments, artificial dataset is generated according to the Alarm network configuration from website <http://www.norsys.com>. The discrete variables X_2, X_4, \dots, X_{36} were covert to continuous by the method provided in reference [12].

Let $\eta_0 = 0.5$ for the first network structure iteration. Fig.2. (a) shows the discretizing iteration convergence for continuous variables X_{13}, X_{35}, X_{27} taking dimensions of 4, 4, 2 respectively. The discretizing iteration processes converge after 9 iterations, demonstrating high discretizing efficiency, while other discretization method needs a high cost. Similar results are observed for the discretization process of other variables. Fig.2. (b) shows Structure iteration processes. Structure sequence quickly converges as well which the arithmetic has high efficiency.

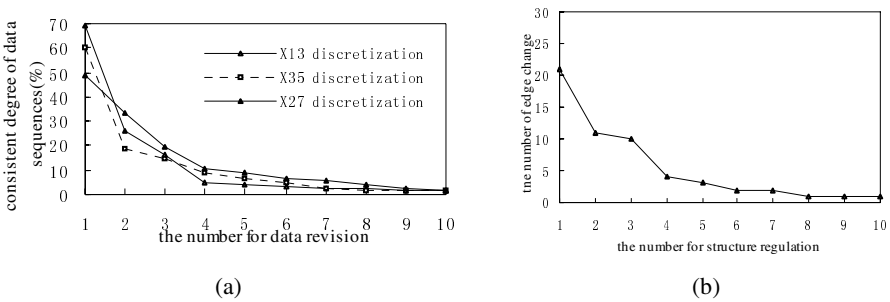


Fig. 2. Convergences of the discretizing and struture iteration process

Let $n_0 = 3$. The reliabilities of the extended entropy discretization method(EE-DM) and the method we proposed based on clustering(CB-DM) are compared and illustrated in Fig.3. As shown in Fig.3, the extended entropy discretization method outperforms the clustering-based discretization method for small dataset. But with the increasing of case number, the gap between the two methods grows narrower and at

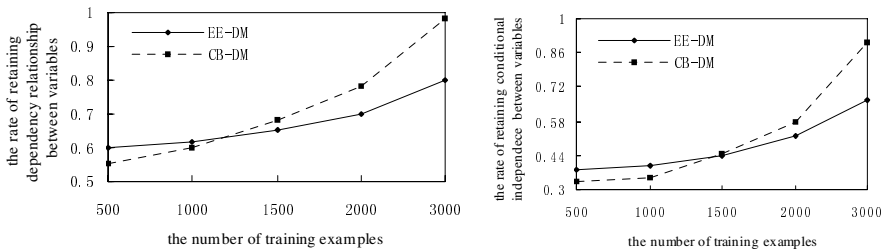


Fig. 3. Experimental compare between extended entropy discretization method and clustering-based discretization method

last, the clustering-based discretization method surpasses the extended entropy discretization method. The main reason that accounts for this phenomenon is: the relationship between variables becomes clear as the cases increase, and is utilized to direct the data discretization and network structure regulation. Furthermore, while in the extended entropy discretization method, the influence of the local maxima of the discretization and structure learning process increase with the increase of the cases.

We also selected three classification data sets from the UCI repository. The data sets used were: iris, liver_disease, and pima_indians_diabetes. Fig.4. show the results of our experiment, where the last variable is class and others are attributes.

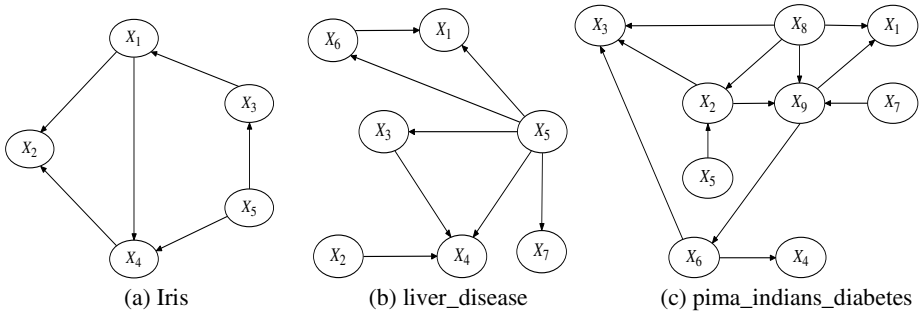


Fig. 4. The learned Bayesian networks structures

5 Conclusion

An efficient method of learning Bayesian network structure with continuous variables is proposed in this paper. The learning process is an iteration. In each iteration, on the one hand, continuous variables are discretized based on clustering according to parent nodes structure and Gibbs sampling, the optimal cluster number is determined by the variable's Markov blanket and MDL scoring, thus retain the dependency relationship between variables; on the other hand, the network structure is improved by structure optimizing regulation until convergence. Causal relationship can be effectively found by using the method and the problems coming along with the extended entropy discretization method can be avoided. Moreover, the method can be generalized to deal with other correlative problems with continuous variables.

Acknowledgments

This research is supported by National Science Foundation of China under Grant No. 60275026, Shanghai Academic Discipline Project under Grant No. P1601 and Key Research Project from Shanghai Municipal Education Commission under Grant No. 05zz66.

References

1. Cooper G F, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 1992, 9 (4): 309-347
2. Lam W, Bacchus F. Learning Bayesian belief networks: an approach based on the MDL principle. *Computational Intelligence*, 1994, 10(4): 269-293
3. Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. San Mateo, California, Morgan Kaufmann, 1988, 383-408
4. Heckerman D, Geiger D, Chickering D M. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 1995, 20(3): 197-243
5. Cheng Jie, Bell D, Liu Wei-ru. Learning Bayesian networks from data: An efficient approach based on information theory. *Artificial Intelligence*, 2002, 137 (1-2): 43-90
6. Geiger D, Heckerman D. Learning Gaussian networks. Technical Report MSR-TR-94-10, Microsoft Research, Redmond, 1994
7. Olesen K G. Causal probabilistic networks with both discrete and continuous variables. *IEEE Trans, on Pattern Analysis and Machine Intelligence*, 1993, 3(15): 275-279
8. Xu L, Jordan M I. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Computation*, 1996, 8 (1): 129-151
9. Monti S, Cooper G F. Learning hybrid Bayesian networks from data. *Learning in Graphical Models*, Kluwer Academic Publishers, 1998
10. Fayyad U, Irani K. Mult-interval discretization of continuous-valued attributes for classification learning. In *Proceedings International Joint Conference on Artificial Intelligence*, Chambéry, France, 1993, 1022-1027
11. Friedman N, Goldszmidt M. Discretization of continuous attributes while learning Bayesian networks. In: *Proceedings 13th International Conference on Machine Learning*, Bari, Italy, 1996, 157-165
12. Wang F, Liu D Y, Xue W X. Discretizing continuous variables of Bayesian networks based on genetic algorithms. *Chinese Journal of Computers*, 2002, 25(8), 794-800.
13. Chickering D M. Learning Bayesian networks is NP-Hard. Technical Report MSR-TR-94-17, Microsoft Research, Redmond, 1994
14. Mao S S, Wang J L, Pu X L. *Advanced mathematical statistics*. 1th ed., Beijing: China Higher Education Press, Berlin: Springer-Verlag, 1998, 401-459.
15. Geman S, Geman D. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1984, 6(6): 721-742
16. Buntine W L. Chain graphs for learning. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*(P. Besnard and S. Hanks, eds.), Morgan Kaufmann, San Francisco. 1995, 46-54
17. Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 1997, 29(2-3): 103-130

A Unified Strategy of Feature Selection

Peng Liu, Naijun Wu, Jiaxian Zhu, Junjie Yin, and Wei Zhang

School of Information Management and Engineering,
Shanghai University of Finance and Economics, Shanghai, 200433, P.R. China
liupeng@mail.shufe.edu.cn

Abstract. In the field of data mining (DM), feature selection is one of the basic strategies handling with high-dimensionality problems. This paper makes a review of current methods of feature selection and proposes a unified strategy of feature selection, which divides overall procedures of feature selection into two stages, first to determine the FIF (Feature Important Factor) of features according to DM tasks, second to select features according to FIF. For classifying problems, we propose a new method for determining FIF based on decision trees and provide practical suggestion for feature selection. Through analysis on experiments conducted on UCI datasets, such a unified strategy of feature selection is proven to be effective and efficient.

1 Introduction

Real world datasets often consist of some redundant features, which are not involved in data mining (DM) model. There are just partial features to be concerned among majority of features in dataset. Moreover, existence of 'curse of dimensionality' also makes DM problem focused on partial feature subsets. 'Curse of dimensionality' refers that, to maintain the accuracy of model, the size of dataset in an n -dimensional space increases exponentially with dimensions. High-dimension has four important characteristics [1]:

1. The size of a data set yielding the same density of data points in an n -dimensional space increases exponentially with dimensions;
2. A larger radius is needed to enclose a fraction of the data points in a high-dimensional space;
3. Almost every point is closer to a boundary than to another sample point in a high-dimensional space;
4. Almost every point is an outlier.

These rules of the 'curse of dimensionality' often have serious consequences when dealing with a finite number of samples in a high-dimensional space. Properties 1 and 2 reveal the difficulty in making estimates for high-dimensional samples. We need more samples to establish the required data density for planned mining activities. Properties 3 and 4 indicate the difficulty of predicting a response at a given point, since any new point will on average be closer to a boundary than to the training examples in the central part. Therefore, as to a high-dimension dataset, the size of dataset is so large that learning might not perform quite well until removing these unwanted features.

Reducing the number of irrelevant/redundant features drastically reduces the running time of a learning algorithm and yields more general knowledge.

There are two basic strategies handling with high-dimensional dataset: Feature Selection and Feature Transforming. The second is simply to transform current features based on certain algorithms. Although it can cope with some high-dimensional problems, new variables may bring about issues on explanation. This paper lays a strong emphasis on the first strategy, Feature Selection. It is reviewed in Sect. 2. Then, a novel methodology, that is, a unified strategy of feature selection is introduced in Sect. 3. Experiments in Sect. 4 prove its efficiency and applicability. Summary and perspective of future work will be given in the end.

2 Methods of Feature Selection

2.1 Concept and Application Framework of Feature Selection

Feature selection is a process choosing feature subset from the original features so that the feature space is optimally reduced according to a certain criterion. It is a data pre-processing technique used often to deal with large datasets in DM. Its aim is to: reduce the dimensionality of feature space; speed up learning algorithm; improve the comprehensibility of the mining results; and increase the performance of mining algorithms (e.g., prediction accuracy).

Theoretically, feature selection methods search through feature subsets to find the best one among the competing 2^N ones according to some evaluation function (N is the number of all features) [2]. Finding the best subset is usually intractable. Therefore, troubleshooting feature selection often adopts heuristic searching in order to reduce computational complexity. Dash and Liu (1997) introduced a framework of typical feature selection method of 4 basic steps [3] (Fig.1)

1. a *generation procedure* to generate the next candidate subset;
2. an *evaluation function* to evaluate the subset under examination;
3. a *stopping criterion* to decide when to stop; and
4. a *validation procedure* to check whether the subset is valid.

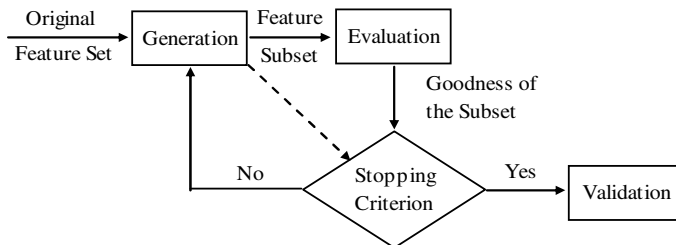


Fig. 1. Feature selection process with validation

2.2 Classification of Feature Selection Methods

According to different application to the datasets, feature selection can be categorized into 'supervised' and 'unsupervised'. The distinction between 'supervised' and 'unsupervised' for DM methods comes from classification problem. If using a training dataset with correct classifications for learning specific predictive patterns, it is called 'supervised'. If we just use the dataset itself and its internal structure without classification information, the method is called 'unsupervised'.

1) Feature selection for supervised learning

The methods of feature selection for supervised learning can be grouped into *filter* and *wrapped* approaches, based on their dependence on the inductive algorithms [4]. Filter methods are independent of the inductive algorithm, whereas wrapper methods use the inductive algorithm as the evaluation function.

2) Feature selection for unsupervised learning

As DM has penetrated to more application fields, feature selection for unsupervised learning is being concerned increasingly. The methods include Principal Components Analysis (PCA), Factor Analysis, Independent Components Analysis (ICA), etc.

3 A Unified Strategy of Feature Selection

This paper proposes a unified strategy of feature selection, which considers the task of feature selection into two stages: first stage is to identify Feature Important Factor (FIF), and second is to select important features according to those FIFs.

3.1 Identification of Feature Important Factor

Feature selection is to choose features which are important for DM tasks. Then, how to describe the importance of features? This paper determines Feature Important Factor (FIF) for every feature to confirm their importance. The bigger FIF is, the more important the feature is. So, if we range all features according to their FIFs by descending order, we can get the order of the importance of features.

FIF can be gained by judgment of experts' experience or by Information-gain (or Information-gain Ratio) when dealing with classifying problems. Ratanamahatana and Gunopulos (2003) took advantage of the features used when constructing decision tree [5]. C4.5 decision tree is generated by 10% of data samples and the features in the first 3 layers of the tree are selected as one group of candidate features. The process is repeated 10 times and 10 groups are combined as final important feature subsets. In spite of efficiency of this method, issue of too much features is also generated. On average, 37.6% of original ones are selected as important features. At the same time, more than 50% of original features in nearly half datasets are selected.

Based on the idea of Ratanamahatana, we propose a method for determining FIF, which is calculated under the structure of decision tree, and compare such method with those based on Information-gain and Information-gain Ratio.

Decision tree is similar to the structure of data flow. Each node on the tree represents a test of each feature. Each branch is the output of testing and each leaf represents the distribution of one of the classes. Node of top layer is root of the tree.

Different decision tree models choose different algorithms to test features, which will result in different structure of trees. But all the tree models select testing features which can do most contribution for classification. It is believed that testing features in the tree play an important role in classification. The features on different branches and layers reflect their importance relatively. Based on some famous decision trees, such as J.R.Quinlan's C4.5 and Breiman's CART, FIF can be gained. This paper adopts C4.5 model and its improved model R-C4.5_s [6] to identify the FIF.

Our method of identifying 'Feature Important Factor' (FIF), which is based on the decision tree structure, includes procedures as follows. Firstly, we define the FIF of the feature at the root as 1. Secondly, FIF of other node feature is defined as FIF of its father-node feature multiplying a weight. Such a weight is the proportion of records number on current node to the records number of its father node. On most cases, a feature may be chosen as a testing feature in different branches, or on different layers. Then, FIF of such a feature is the sum of FIFs in different branches and on different layers. FIF of those features that do not appear in the tree is defined as 0. Thus, FIF is normalized between 0 and 1. 1 means this feature is strongly important while close to 0 means weakly important.

3.2 Selecting Features Based on 'Feature Important Factor'

Since FIF reflects the importance of features, firstly we range all features according to their FIFs by descending order, then choose the important features according to DM tasks. We can choose prior features according to some percentage. We can also select features with FIF or number between some given bound to assure that the number of selected features is within a rational range.

4 Experiment Analysis

4.1 Datasets

This paper selected 9 datasets from the UCI datasets [7], all features of the 9 datasets are nominal. In order to conduct our experiments, some records or features with missing data were deleted in advance. Information of those 9 datasets is displayed in Table 1.

4.2 Design and Process of Experiments

Firstly, we calculate FIF of all features and range all of them according to their FIFs by descending order, thus, order of importance of all features can be gained. Secondly, we choose prior 1, 2, ..., N features (N is the number of all features), use Naïve Bayes classifier (NBC) by Weka [8], a shareware, to record the prediction accuracy of target feature. The features used when prediction accuracy achieves the peak are the optimal feature subset. The reason to choose NBC is that NBC performs best when features are important and less co-related.

Table 1. Introduction of 9 datasets

No.	Datasets	Original (including missing data)		Current (without missing data)		#Classes
		#Records	#Features	#Records	#Features	
1	Audiology	226	70	146	69	22
2	Breast-cancer	286	10	277	10	2
3	Kr vs kp	3196	37	3196	37	2
4	Mushroom	8124	23	8124	22	2
5	Vote	435	17	232	17	2
6	Zoo	101	18	101	18	7
7	Primary-tumor	339	18	270	17	22
8	Soybean	682	36	562	36	15
9	Splice	3190	62	3190	61	3

4.3 Result Analysis of Experiments

1) Strategy's performance on reducing features and raising prediction accuracy

The unified strategy of feature selection advanced in our paper is quite efficient. Limited by the space, we choose the results of datasets 'Audiology' and 'Mushroom', which are displayed in Fig.2 and Fig. 3 respectively. Results of all datasets can be seen in Table 2 and Table 3 (In all tables and figures, FIF-IG means the FIF feature selection method based on Information-gain while FIF-GR refers to Information-gain Ratio, the same is to FIF-C4.5 and FIF-RC4.5s).

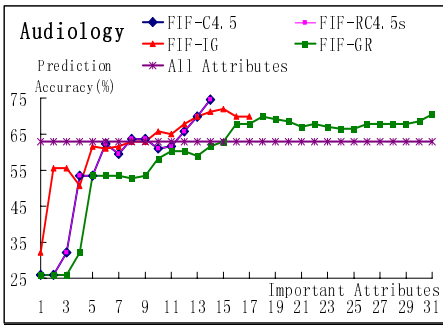


Fig. 2. Results of 'Audiology'

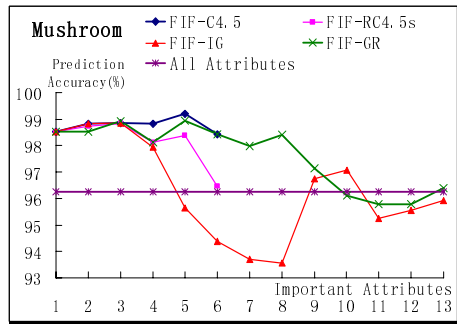


Fig. 3. Results of 'Mushroom'

It can be seen by the results that the unified feature selection strategy based on FIF performs efficiently. The feature subset is much smaller than the original, moreover, we can achieve higher prediction accuracy using the subset than using whole features. In summary, FIF methods based on decision tree outstand those based on Information-gain and Information-gain Ratio. Between the two decision trees, as an improved edition of C4.5, FIF method on R-C4.5s chooses less features (28.79% of the whole on average, see Table 2) and achieves higher accuracy (3.53% over the whole on average, see Table 3), showing significant advantage.

Table 2. The optimal features selected by 4 FIF methods

No.	Datasets	All Featu.	FIF-C4.5		FIF-RC4.5s		FIF-IG		FIF-GR	
			Optimal Featu.	%	Optimal Featu.	%	Optimal Featu.	%	Optimal Featu.	%
1	Audiology	68	14	20.59	14	20.59	15	22.06	31	45.59
2	Breast	9	2	22.22	2	22.22	2	22.22	5	55.56
3	Kr vs kp	36	5	13.89	5	13.89	3	8.33	7	19.44
4	Mushroom	21	5	23.81	2	9.52	3	14.29	5	23.81
5	Vote	16	1	6.25	1	6.25	1	6.25	1	6.25
6	Zoo	17	5	29.41	6	35.29	8	47.06	9	52.94
7	Primary	16	15	93.75	14	87.50	12	75.00	16	100.00
8	Soybean	35	10	28.57	13	37.14	35	100.00	35	100.00
9	Splice	60	14	23.33	16	26.67	17	28.33	20	33.33
Average				29.09± 23.84		28.79± 23.03		35.95± 30.38		48.55± 31.36

Table 3. Percentage of raised prediction accuracy under optimal feature subset

No.	Datasets	FIF-C4.5 (%)	FIF-RC4.5s (%)	FIF-IG (%)	FIF-GR (%)
1	Audiology	11.64	11.64	8.90	7.53
2	Breast-cancer	1.08	1.08	1.08	0.00
3	Kr vs kp	6.45	6.45	2.53	6.45
4	Mushroom	2.97	2.57	2.61	2.70
5	Vote	6.03	6.03	6.03	4.68
6	Zoo	0.99	3.96	1.98	3.96
7	Primary-tumor	0.00	0.00	0.74	0.00
8	Soybean	0.18	0.00	0.00	0.00
9	Splice	0.03	0.00	0.85	0.91
Average		3.26±3.78	3.53±3.73	2.75±2.73	2.91±2.74

2) Precision Ratio and Recall Ratio of 4 FIF feature selection methods

In order to validate the new strategy, we evaluate the 4 FIF feature selection methods by 2 indexes, Precision Ratio and Recall Ratio. Precision is the proportion of important features in feature subset. Recall is the proportion of selected important features in total important features. Among all, important features mean those features which are chosen by the 4 methods jointly.

Table 4. Precision and recall ratio of 4 feature selection methods

No.	Datasets	Precision Ratio (%)				Recall Ratio (%)			
		FIF-C4.5	FIF-RC4.5 _s	FIF-IG	FIF-GR	FIF-C4.5	FIF-RC4.5 _s	FIF-IG	FIF-GR
1	Audiology	97.98	95.67	70.93	71.23	85.23	80.56	66.87	66.87
2	Breast	100.00	100.00	50.00	50.00	100.00	100.00	50.00	50.00
3	Kr vs kp	63.64	63.64	40.91	59.09	100.00	100.00	64.29	92.86
4	Mushroom	60.00	80.00	60.00	80.00	75.00	100.00	75.00	100.00
5	Vote	60.00	60.00	60.00	60.00	100.00	100.00	100.00	100.00
6	Zoo	50.00	50.00	33.33	50.00	100.00	100.00	66.00	100.00
7	Primary	92.86	92.86	92.86	92.86	100.00	100.00	92.31	76.92
8	Soybean	75.00	75.00	65.00	60.00	100.00	100.00	86.67	80.00
9	Splice	47.06	76.47	76.47	76.47	61.54	100.00	100.00	100.00
Average		71.80± 19.34	74.20± 16.12	62.40± 17.20	68.70± 13.67	91.31± 13.51	97.84± 6.11	77.90± 17.66	85.18± 17.01

Table 5. Number of selected features and prediction accuracy (%) under different suggestion

Datasets	All Featu.	a		b		c		d		Gap in Accuracy		
		All*20%		R-C4.5s*2/3		Selected Featu. Subset		Optimal Subset		d-a	d-b	d-c
		Num	Accuracy	Num	Accuracy	Num	Accuracy	Num	Accuracy			
Audiology	68	14	74.66	9	63.70	14	74.66	14	74.66	0.00	10.96	0.00
Breast	9	2	76.53	1	72.92	2	76.53	2	76.53	0.00	3.61	0.00
Kr vs kp	36	7	92.12	13	90.99	13	90.99	5	94.34	2.22	3.35	3.35
Mushroom	21	4	98.12	4	98.12	4	98.12	2	98.82	0.70	0.70	0.70
Vote	16	3	95.17	1	96.98	1	96.98	1	96.98	1.81	0.00	0.00
Zoo	17	3	83.17	4	88.12	4	88.12	6	97.03	13.86	8.91	8.91
Primary	16	3	32.22	9	42.22	9	42.22	16	46.30	14.08	4.08	4.08
Soybean	35	7	81.32	11	89.68	11	89.68	13	91.64	10.32	1.96	1.96
Splice	60	12	88.56	13	88.43	13	88.43	60	95.36	6.80	6.93	6.93
Average										5.53	4.50	2.88
										±5.54	±3.50	±3.07

See Table 4, we find that FIF-RC4.5_s performs best in both Precision and Recall, the followings are FIF-C4.5 and FIF-GR, FIF-IG is the worst. Therefore, we can regard feature subset generated by FIF integrated with R-C4.5s the best choice. What's more, we find that, FIF-RC4.5_s approaches FIF-C4.5 in the number and order of selected features, but differs much from FIF-IG and FIF-GR.

3) Selecting features based on FIF integrated with R-C4.5s

In the end of this section, we will provide some suggestion after analyzing Table 2, 3, 4, that is, determining FIF according to R-C4.5s decision tree model and selecting features. The size of feature subset should be no bigger than subset used by R-C4.5s. Generally speaking, first, we select 20% of all features and 2/3 of the subset used by R-C4.5s as the subsets respectively, then we choose the bigger one as the final feature subset. Results of different suggestions are shown in Table 5. Although several suggestions perform worse than the optimal subset (when accuracy achieves the peak) more or less respectively, the method of choosing the maximal in 20% of all features and 2/3 of subset used by R-C4.5s as the final feature subset can be the best choice, since the average and standard-deviation are both the least in the gap from the peak.

5 Conclusions and Future Work

Feature selection has become one of the focuses in DM fields during recent years. However, proposing a practical and simple feature selection method is still a challenge. We propose a unified strategy of feature selection, which divides overall procedures of feature selection into two stages, first to determine the FIF according to DM tasks, second to select features according to FIF. There are different methods to determine FIF according to different DM tasks, which is our future work. Selecting final feature subsets based on FIF and applying such a strategy to other feature selection problems are also tough tasks to be accomplished in the future.

References

1. Kantardzic, M.: *Data Mining Concepts, Models, Methods, and Algorithms*. A John Wiley & Sons, INC (2003)
2. Tan, P.N., Steinbach, M., Kumar, V.: *Introduction to Data Mining*. Pearson Education, INC (2006)
3. Dash, M.: Feature Selection for Classification. *Intelligent Data Analysis*. 1 (1997) 131–156
4. Das, S.: Filters, Wrappers and A Boosting Based Hybrid for Feature Selection. In: *Proceedings of the Eighteenth International Conference on Machine Learning* (2001) 74–81
5. Ratanamahatana, C.A., Gunopulos, D.: Feature Selection for the Naive Bayesian Classifier Using Decision Trees. *Applied Artificial Intelligence*. 17 (5–6), (2003) 475–487
6. Liu, P.: R-C4.5: A Robust Decision Tree Improved Model. In: *Proceedings of ISICA'05 (The International Symposium on Intelligent Computation and Its Application), Progress in Intelligent Computation and Its Applications, Wuhan, China* (2005) 454–459
7. Merz, C.J., Murphy, P.M.: *UCI Repository of Machine Learning Datasets*. <http://www.ics.uci.edu/~mllearn/MLRepository.html> (1998)
8. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*. Morgan Kaufmann, London (2005)

Experimental Comparison of Feature Subset Selection Using GA and ACO Algorithm

Keunjoon Lee¹, Jinu Joo², Jihoon Yang^{3,*,**}, and Vasant Honavar⁴

¹ Kookmin Bank, Sejong Daewoo B/D
167 Naesu-Dong, Jongno-Ku, Seoul 110-070, Korea
leekjsg@hanmail.net

² Development Laboratory 1, Mobile Handset R&D Center
Mobile Communications Company, LG Electronics Inc.
Gasam-Dong, Gumchon-Ku, Seoul 153-801, Korea
jujoo@lge.com

³ Department of Computer Science, Sogang University
1 Shinsoo-Dong, Mapo-Ku, Seoul 121-742, Korea
yangjh@sogang.ac.kr

⁴ Artificial Intelligence Research Laboratory, Department of Computer Science
Iowa State University Ames, IA 50011 USA
honavar@cs.iastate.edu

Abstract. Practical pattern classification and knowledge discovery problems require selecting a useful subset of features from a much larger set to represent the patterns to be classified. Exhaustive evaluation of possible feature subsets is usually infeasible in practice because of the large amount of computational effort required. Bio-inspired algorithms offer an attractive approach to find near-optimal solutions to such optimization problems. This paper presents an approach to feature subset selection using bio-inspired algorithms. Our experiments with several benchmark real-world pattern classification problems demonstrate the feasibility of this approach to feature subset selection in the automated design of neural networks for pattern classification and knowledge discovery.

1 Introduction

In practical pattern classification and knowledge discovery problems, many input data contain large amount of features or attributes which are mutually redundant and irrelevant with different associated measurements. Among these large number of features, selecting useful subset of features to represent the patterns that are presented to a classifier mainly affect the accuracy, time, the number of examples needed for learning a sufficiently accurate classification function, the cost of performing classification using the learned classification function, and the comprehensibility of the knowledge acquired through learning. Therefore this

* This work was supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation and by the Brain Korea 21 Project in 2006.

** Corresponding author.

presents us with a feature subset selection problem in pattern classification tasks. The feature subset problem is to identify and select a useful subset of features in order to use to represent patterns from a much larger set of features. Many feature subset selection methods have been introduced for automated design for pattern classifiers. We introduce a new feature subset selection approach based on bio-inspired algorithms and selected feature subsets evaluated by a neural network (DistAI). We present our experimental results from various experiments and prove our methods usability with several benchmark classification problems.

2 Feature Selection Using Bio-inspired Algorithms for Neural Network Pattern Classifiers

Among a number of bio-inspired algorithms, we consider the GA and ACO algorithm in this paper.

2.1 Genetic Algorithm

Evolutionary algorithms [1,2,3,4] include a class related randomized, population-based heuristic search techniques which include genetic algorithms [1,2], genetic programming [3], evolutionary programming [4]. They are inspired by processes that are modeled after biological evolution. The individuals represent candidate solutions to the optimization problem being solved. A wide range of genetic representations (e.g. bit vectors, LISP programs, matrices, etc.) can be used to encode the individuals depending on the space of solutions that needs to be searched. In the feature subset selection problem, each individual would represent a feature subset. It is assumed that the quality of each candidate solution (or fitness of the individual in the population) can be evaluated using a fitness function. In the feature subset selection problem, the fitness function would evaluate the selected features with respect to some criteria of interest (e.g. cost of the resulting classifier, classification accuracy of the classifier, etc.).

2.2 ACO Algorithm

The ant algorithm is a heuristic search algorithm using artificial ants known as multi-agents which run parallel when constructing feasible solutions probabilistically based on pheromone information deposited upon each plausible candidate solution or trail. The early version of the ant algorithm introduced was known as ant system (AS) [5] algorithm by Dorigo. Recently variants of ant algorithm were combined in a common frame work called ant colony optimization (ACO) meta-heuristic [6]. In this paper we have adopted the graph based ant system (GAS) [7] which has been mentioned by Gutjahr. GAS is a specific version of ACO meta-heuristic algorithm where candidate solutions can be represented in directed graphs. It is particularly successful in solving combinatorial optimization problems such as constructing paths based on direct graphs with specific starting points. GAS updates pheromone globally: pheromone trail is updated after all ants have traveled in its cycle, and provides a pheromone evaporation

factor to prevent ants converging into local minima. In our ant algorithm elite policy is used for updating pheromone information on each trail. Throughout this paper algorithms that follow the ACO meta-heuristic will be called ACO algorithm.

2.3 DistAl: A Fast Algorithm for Constructing Neural Network Pattern Classifiers

Because feature subset selection method powered by ACO algorithm require numerous cycles of running the ACO algorithm itself and each cycle contains a lot of ants holding candidate solutions to be evaluated by training the neural network, it is not feasible to use computationally expensive iterative weight update algorithms. Consequently DistAl, offering a fast and efficient approach in training neural networks, is used for evaluating the fitness of the candidate solution. DistAl [8] is a simple and relatively fast constructive neural network learning algorithm for pattern classification. The results presented in this paper are based experiments using neural networks constructed by DistAl. The key idea behind DistAl is to add *hyperspherical* hidden neurons one at a time based on a greedy strategy which ensures that the hidden neuron correctly classifies a maximal subset of training patterns belonging to a single class. Correctly classified examples can then be eliminated from further consideration. The process terminates when the pattern set becomes empty (that is, when the network correctly classifies the entire training set). When this happens, the training set becomes linearly separable in the transformed space defined by the hidden neurons. In fact, it is possible to set the weights on the hidden to output neuron connections without going through an iterative process. It is straightforward to show that DistAl is guaranteed to converge to 100% classification accuracy on any finite training set in time that is polynomial in the number of training patterns [8]. Experiments reported in [8] show that DistAl, despite its simplicity, yields classifiers that compare quite favorably with those generated using more sophisticated (and substantially more computationally demanding) learning algorithms. This makes DistAl an attractive choice for experimenting with social intellectual approaches to feature subset selection for neural network pattern classifiers.

3 Implementation Details

In this section we explain our implementation details on GA and ACO algorithms which are utilized in our feature subset selection problem.

3.1 GA Implementation

Our GA algorithm is based on rank-based selection strategy described in Figure 1. The rank based selection strategy gives a non-zero probability of selection of each individual [9]. For more specific implementation details look at [10].

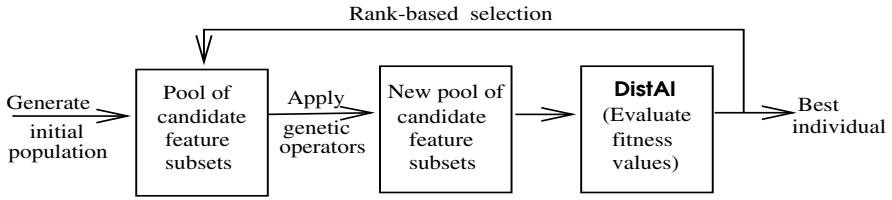


Fig. 1. GADistAl: Feature subset selection using a genetic algorithm with DistAl

3.2 ACO Implementation

Our ACO algorithm is based on Gutjahr’s GAS algorithm [7] with the following adjustments.

– Representation:

Each traversed path by an ant in a cycle represents a candidate solution to the feature subset selection problem. As described in Figure 2, the selected features are represented as combination of arcs where ants have traversed through the graph. Note that every ant must visit every node in the graph no more than once and every ant starts at a specific node (first feature) and ends at a specific node (last feature) visiting every node in between with a given sequence. Every node has two arcs connected to its next visiting node, each representing either selection or exclusion of the feature it is assigned to. Therefore combining traversed arcs together gives a full representation of a candidate solution of feature selection, defined as a path, to classify the given dataset.

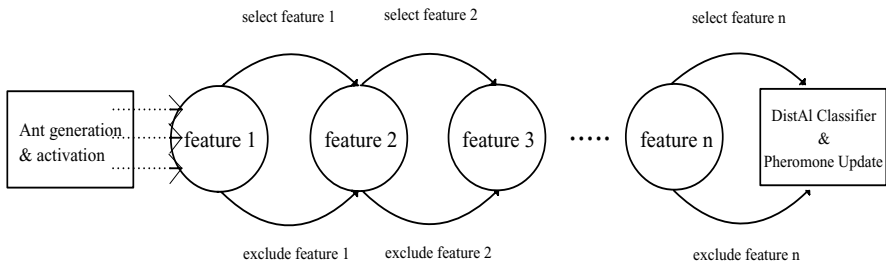


Fig. 2. ACODistAl: Feature subset selection using ACO algorithm with DistAl

– Definition of pheromone update rule with elite policy:

Each node has two choices of arc leading to the next neighboring node. That is, if $\forall i, j \in V$ then $\forall (i, j) \in E$ is $(i, j) = (i, j)^+ \cup (i, j)^-$ where V is the set of nodes in the graph, j the next visiting node from i , E the set of arcs in the graph, and $(i, j)^+, (i, j)^-$ are selection and exclusion arcs from node i to j respectively. Therefore the initial pheromone on each trail

is, $\tau_{ij} = 0.5 = 1/(\text{number of arcs possible to traverse from node } i)$. Unlike GAS algorithm, we introduce an *elite policy* to guide our pheromone update on each path. Pheromone updates occur on paths that outperform the best path in the previous cycle. In other words, paths that perform better than the previous best are the only paths considered to deposit more pheromone on the trail. The partial pheromone deposited on each arc by each ant is,

$$\Delta\tau_{ij}^s = \begin{cases} \mu(p_m^s) & \text{if } \mu_{m-1}^* \leq \mu(p_m^s) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where μ_{m-1}^* is the best performance measurement performed at cycle $m-1$, and p_m^s is the path built by ant s at cycle m . From (1) the total pheromone deposited on each arc when an elite path is found is $\Delta\tau_{ij} = \frac{1}{C} \sum_{s=1}^S \Delta\tau_{ij}^s$,

where C is the normalization factor defined as $C = \sum_{(i,j)} \sum_{s=1}^S \Delta\tau_{ij}^s$.

Therefore the pheromone update rule is $\tau_{ij}(m+1) = (1-\rho)\tau_{ij}(m) + \rho\Delta\tau_{ij}$,

where ρ is the evaporation factor and m is the number of cycles performed so far. On the contrary, if an elite has been found on the m_{th} cycle then the pheromone update rule is $\tau_{ij}(m+1) = \tau_{ij}(m)$.

– **Definition of transition probability:**

From the pheromone update rule introduced above, the transition probability is estimated as,

$$p_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{i,k \in A} (\tau_{ij})^\alpha (\eta_{ij})^\beta} \quad (2)$$

where η is the heuristic value, and α, β are parameters.

– **Setting of user-controlled parameters:**

Iteration of ACODistAl performed : 5; Number of ant : 50; Number of cycle : 20; Evaporation factor ρ : 0.3; Heuristic value η : 1; Transition probability parameter : $\alpha = 1.0$, $\beta = 1.0$.

4 Experiments

4.1 Description of Datasets

The experiments reported here used a wide range of real-world datasets from the machine learning data repository at the University of California at Irvine [11] as well as a carefully constructed artificial dataset (3-bit parity) to explore the feasibility of using bio-inspired algorithms for feature subset selection for neural network classifiers¹. The feature subset selection using DistAl is also applied to document classification problem for journal paper abstracts. For more details on datasets see [10].

¹ [<http://www.ics.uci.edu/mllearn/MLRepository.html>]

Table 1. Datasets used in the experiments. *Size* is the number of patterns in the dataset, *Features* is the number of input features, and *Class* is the number of output classes.

<i>Dataset</i>	<i>Size</i>	<i>Features</i>	<i>Feature Type</i>	<i>Class</i>
3-bit parity problem (3P)	100	13	numeric	2
audiology database (Audiology)	200	69	nominal	24
pittsburgh bridges (Bridges)	105	11	numeric, nominal	6
breast cancer (Cancer)	699	9	numeric	2
credit screening (CRX)	690	15	numeric, nominal	2
flag database (Flag)	194	28	numeric, nominal	8
heart disease (Heart)	270	13	numeric, nominal	2
heart disease [Cleveland](HeartCle)	303	13	numeric, nominal	2
heart disease [Hungarian](HeartHun)	294	13	numeric, nominal	2
heart disease [Long Beach](HeartLB)	200	13	numeric, nominal	2
heart disease [Swiss](HeartSwi)	123	13	numeric, nominal	2
hepatitis domain (Hepatitis)	155	19	numeric, nominal	2
horse colic (Horse)	300	22	numeric, nominal	2
ionosphere structure (Ionosphere)	351	34	numeric	2
pima indians diabetes (Pima)	768	8	numeric	2
DNA sequences (Promoters)	106	57	nominal	2
sonar classification (Sonar)	208	60	numeric	2
large soybean (Soybean)	307	35	nominal	19
vehicle silhouettes (Vehicle)	846	18	numeric	4
house votes (Votes)	435	16	nominal	2
vowel recognition (Vowel)	528	10	numeric	11
wine recognition (Wine)	178	13	numeric	3
zoo database (Zoo)	101	16	numeric, nominal	7
paper abstracts 1 (Abstract1)	100	790	numeric	2
paper abstracts 2 (Abstract2)	100	790	numeric	2

4.2 Experimental Results

The experiment explored the performance of ACODistAl, comparing it with GA-based approaches for feature subset selection. The parameter setting described in Section 3 was chosen for fair comparison of ACODistAl with GADistAl.

Fitness evaluation was obtained by averaging the observed fitness value for 10 different partitions of the data into training and test sets. The final results are estimated by averages over 5 independent runs of the algorithm which are shown in Table 2. The entries in the tables give the means and standard deviations in the form *mean* \pm *standard deviation*. The results of Table 2 show that, in most of the datasets ACODistAl and GADistAl perform better than the original DistAl with full feature sets. Datasets with similar accuracy among the three algorithm show that ACODistAl and GADistAl can perform with high accuracy with almost half the features used to classify the dataset. For example nearly 90 \sim 95% accuracies were yielded in **Cancer**, **HeartSwi**, **Promoters**, **Votes** and **Abstract1** datasets, where ACODistAl and GADistAl classified each dataset with almost half the features used in DistAl. Contrary to the fact that

Table 2. Comparison of neural network pattern classifiers constructed by DistAl using the entire set of features with the best network constructed by GADistAl and ACODistAl using fitness estimates based on 10-fold cross-validation.

<i>Dataset</i>	DistAl		GADistAl		ACODistAl	
	<i>Features</i>	<i>Accuracy</i>	<i>Features</i>	<i>Accuracy</i>	<i>Features</i>	<i>Accuracy</i>
3P	13	79.0±12.2	4.8 ± 0.7	100.0 ± 0.0	10.8 ± 0.4	100 ± 0.0
Audiology	69	66.0±9.7	37.2 ± 1.8	72.6 ± 2.8	31.2 ± 2.5	68.2 ± 2.4
Bridges	11	63.0 ± 7.8	4.9 ± 0.6	56.9 ± 7.6	5.8 ± 1.5	67.6 ± 1.8
Cancer	9	97.8 ± 1.2	6.0 ± 1.1	98.0 ± 0.3	5.4 ± 0.8	97.7 ± 0.1
CRX	15	87.7 ± 3.3	7.4 ± 2.6	87.7 ± 0.4	6.8 ± 1.8	89.6 ± 0.2
Flag	28	65.8 ± 9.5	14.2 ± 2.8	63.9 ± 6.1	14.2 ± 2.3	68.3 ± 0.5
Heart	13	86.7 ± 7.6	7.6 ± 0.8	85.5 ± 0.7	9.4 ± 1.2	87.2 ± 0.4
HeartCle	13	85.3 ± 2.7	8.4 ± 0.8	86.9 ± 0.6	12.6 ± 0.5	84.1 ± 0.5
HeartHun	13	85.9 ± 6.3	7.4 ± 1.4	85.4 ± 1.3	6.8 ± 1.8	88.4 ± 0.2
HeartLB	13	80.0 ± 7.4	7.6 ± 1.0	79.8 ± 1.9	7.8 ± 2.2	82.3 ± 0.2
HeartSwi	13	94.2 ± 3.8	7.4 ± 1.7	95.3 ± 1.1	6.2 ± 2.1	95.8 ± 0.0
Hepatitis	19	84.7 ± 9.5	10.2 ± 1.6	85.2 ± 2.9	17 ± 0.0	84.1 ± 0.8
Horse	22	86.0 ± 3.6	9.6 ± 2.7	83.2 ± 1.6	11.4 ± 2.4	85.5 ± 1.1
Ionosphere	34	94.3 ± 5.0	16.6 ± 3.0	94.5 ± 0.8	17.4 ± 1.6	95.4 ± 0.8
Pima	8	76.3 ± 5.1	4.0 ± 1.7	73.1 ± 3.1	3.8 ± 1.0	77.6 ± 0.0
Promoters	57	88.0 ± 7.5	30.6 ± 2.1	89.8 ± 1.7	30.6 ± 4.7	94.3 ± 0.9
Sonar	60	83.0 ± 7.8	32.2 ± 2.2	84.0 ± 1.6	31 ± 4.1	79.6 ± 1.0
Soybean	35	81.0 ± 5.6	21.0 ± 1.4	83.1 ± 1.1	18.2 ± 2.8	43.4 ± 2.9
Vehicle	18	65.4 ± 3.5	9.4 ± 2.1	50.1 ± 7.9	9.4 ± 1.6	68.8 ± 0.7
Votes	16	96.1 ± 1.5	8.2 ± 1.5	97.0 ± 0.7	8.6 ± 2.1	97.2 ± 0.2
Vowel	10	69.8 ± 6.4	6.8 ± 1.2	70.2 ± 1.6	4.2 ± 1.6	49.0 ± 0.4
Wine	13	97.1 ± 4.0	8.2 ± 1.2	96.7 ± 0.7	5.4 ± 0.5	95.1 ± 0.5
Zoo	16	96.0 ± 4.9	8.8 ± 1.6	96.8 ± 2.0	9.4 ± 0.8	95.6 ± 0.5
Abstract1	790	89.0±9.4	402.2 ± 14.2	89.2 ± 1.0	387.2 ± 10.4	90.0 ± 1.1
Abstract2	790	84.0±12.0	389.8 ± 5.2	84.0 ± 1.1	401.0 ± 9.1	88.4 ± 0.5

most of the datasets yield similar performances between ACODistAl and GADistAl, some datasets like **Heart**, **HeartHun**, and **Promoters** showed specifically higher accuracies in ACODistAl compared to the other methods. However, the performance of ACODistAl is much worse in **Soybean** and **Vowel** datasets. We surmise that our current implementation of ACO is not appropriate for those particular problems.

5 Summary and Discussion

GADistAl and ACODistAl are methods to feature subset selection for neural network pattern classifiers. In this paper a fast inter-pattern distance-based constructive neural network algorithm, DistAl, is employed to evaluate the fitness of candidate feature subsets in the ACO algorithm. The performance of ACODistAl was comparable to the GA based approach (GADistAl), both of which outperformed DistAl significantly. The results presented in this paper indicate that

ACO algorithms offer an attractive approach to solving the feature subset selection problem in inductive learning of pattern classifiers in general, and neural network pattern classifiers in particular.

Some directions for future research include: Extensive experiments on alternative datasets including documents and journals; Extensive experimental (and wherever feasible, theoretical) comparison of the performance of the proposed approach with that of other bio-inspired algorithm-based and conventional methods for feature subset selection; More principled design of multi-objective fitness functions for feature subset selection using domain knowledge as well as mathematically well-founded tools of multi-attribute utility theory [12].

References

1. Goldberg, D.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York (1989)
2. Holland, J.: *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA (1992)
3. Koza, J.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA (1992)
4. Fogel, D.: *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ (1995)
5. Dorigo, M., Maniezzo, V., Colomi, A.: The ant system: An autocatalytic optimizing process (1991)
6. Dorigo, M., Di Caro, G.: The ant colony optimization meta-heuristic. In Corne, D., Dorigo, M., Glover, F., eds.: *New Ideas in Optimization*. McGraw-Hill, London (1999) 11–32
7. Gutjahr, W.J.: A graph-based ant system and its convergence. *Future Gener. Comput. Syst.* **16**(9) (2000) 873–888
8. Yang, J., Parekh, R., Honavar, V.: Distal: An inter-pattern distance-based constructive learning algorithm. In: *Proceedings of the International Joint Conference on Neural Networks*, Anchorage, Alaska (1998) 2208 – 2213
9. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA (1996)
10. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. In Motoda, Liu, eds.: *Feature Extraction, Construction and Selection - A Data Mining Perspective*. Kluwer Academic Publishers (1998) 117–136
11. Murphy, P., Aha, D.: *Uci repository of machine learning databases*. Department of Information and Computer Science, University of California, Irvine, CA (1994)
12. Keeney, R., Raiffa, H.: *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, New York (1976)

OMVD: An Optimization of MVD*

Zhi He, Shengfeng Tian, and Houkuan Huang

School of Computer and Information Technology, Beijing Jiaotong University
Beijing 100044, China
hezhidragon@163.com

Abstract. Most discretization algorithms are univariate and consider only one attribute at a time. Stephen D. Bay presented a multivariate discretization(MVD) method that considers the affects of all the attributes in the procedure of data mining. But as the author mentioned, any test of differences has a limited amount of power. We present OMVD by improving MVD on the power of testing differences with a genetic algorithm. OMVD is more powerful than MVD because the former does not suffer from setting the difference threshold and from seriously depending on the basic intervals. In addition, the former simultaneously searches partitions for multiple attributes. Our experiments with some synthetic and real datasets suggest that OMVD could obtain more interesting discretizations than could MVD.

1 Introduction

On the research of data mining, the usage of discrete values is necessary for some algorithms, such as Apriori rules[1] and C4.5[2]. So far, there are many discretization algorithms, most of which are univariate and consider only one attribute at a time. This kind of approach does not work well in finding association rules and decision tree as it will destroy hidden patterns in data. Stephen D. Bay[3] proposed a multivariate discretization algorithm named MVD. After he had split the range of an attribute into depth-equal or width-equal basic intervals, he combined the adjacent similar intervals determined by STUCCO[4].

But MVD also has limitations in testing support differences:

- The algorithm requires user to specify a threshold that denotes the minimum support difference[4] we can tolerate. In addition, it sets a tendency of merging intervals with small combined support. A simple example of this is to consider the case where we are examining the heights of three groups of people, {5ft, 5ft 4in, 6ft}. There are 30, 10 and 10 instances for them, respectively. It is assumed that we be interested in height difference of 10 inches or more. Since a less combined support is got by merging the second and the third groups than by merging the first and the second groups, MVD results in an average height of 5ft 8in and it has an 8 inches difference from the first group. Then the combined interval will be further emerged with the

* This work is funded by China National Natural Science Foundation grants 60442002 and 60443003.

first group and we finally get **{5ft 3.2in}**. As a contrast, we may consider another sequence where merging firstly the first two groups. It results in an average height of 5ft 1in and it has an 11 inches difference from the third group. Finally, we get the set **{5ft 1in, 6ft}**. This example illustrates that MVD has difficulty in finding some satisfying difference with an improper threshold and the combination tendency.

- MVD discretizes the attributes one at a time instead of simultaneously. During the discretization, the algorithm tests the support difference using the contrast sets[3] that only consist of basic intervals. The similarity of two intervals depends largely on the basic intervals. For example, we are testing the difference of two intervals on attribute *A*: [1 3) and [3 5]. The support difference is caused by the interval on *B*: [2 4). But we can not find this difference if we has basic intervals on *B*, for instance, [1 3) and [3 4).

In this paper, we present an algorithm named OMVD to deal with the problems mentioned above. A new measure based on the support difference is proposed to evaluate the quality of discretization. During the discretization, the measure is optimize with a genetic algorithm. We generate immediately the final partitions between the basic intervals instead of merging them bottom-up with some pre-specified tendency. And we calculate the support differences by regarding the partitions as contrast sets instead of regarding the basic intervals as contrast sets.

The rest of paper is organized as follows. In next section, we briefly discuss the state of the art of discretization approaches. We propose OMVD algorithm in detail in section 3. In section 4, we run OMVD and MVD on synthetic and real datasets and compare the experiment results. At the end of paper, we draw a conclusion in section 5.

2 Related Work

Wojciech and Marek[5] described EDRL-MD, an evolutionary algorithm-based system, for learning decision rules from databases. They simultaneously searched for threshold values for all continuous attributes. Srikant and Agrawal[6] dealt with quantitative attributes by finely partitioning the ranges of attributes and then combining adjacent partitions if necessary. Miller and Yang[7] pointed out that [6] might combine intervals that were not meaningful and thus could result in rules conflicting with our intuitions. They presented an alternative approach based on clustering the data and then building association rules by treating the clusters as frequent itemsets. However, they performed a univariate discretization and their results strongly depended on the clustering algorithm and distance metric used. Monti and Cooper[8] used clustering to perform discretization in the absence of a class variable. Ludl and Widmer[9] also studied the problem of discretizing continuous variables for unsupervised learning algorithms such as association rule miners. They discretized a target attribute by clustering the projections of some source attributes on it.

Stephen D. Bay[3] argued that a major disadvantage of [9] was that they only considered how pairs of variables interact and didn't examine higher-order combinations. Thus, they might have difficulty handling data such as the XOR problem. S. Lallich et al[10] proposed a discretization method that preserved the correlation of attributes. They got the principle components(PC) by PCA and clustered the PCs in eigenspace. The discretization was got by reprojecting eigen cut-points to original dimensions.

3 OMVD: An Optimization of Multivariate Discretization

We propose OMVD in this section. Similarly with MVD, OMVD also need to split attributes into basic intervals as a preparation. The difference between them is that the latter generated discretization immediately by partitioning the partitions instead of merging the basic intervals bottom-up. The discretization in OMVD is evaluated by some measures and optimized with a genetic algorithm. In the next subsection, we propose the measures.

3.1 Maximum Support Difference

Here, we present a new definition called *Maximum Support Difference*(MSD) to evaluate the partition between two adjacent intervals.

Definition 1. For two adjacent intervals I_i and I_{i+1} , *CSET* is the set of all deviations [4] of them. Their MSD_i is defined as:

$$\max_{cset \in CSET} |support(cset, I_i) - support(cset, I_{i+1})| \tag{1}$$

The value of equation 1 is the biggest support difference between two intervals caused by all contrast sets.

We only need to do a small adjustment to STUCCO to get MSD. After we have found a *deviation* using the method in [4], we replace δ in equation 2 in [4] with the value of support difference caused by the *deviation* we have got. Iteratively, we get MSD until there are no *deviation*.

For all adjacent intervals, we get a list of MSDs. We have three other measures to describe the whole situation of a list of MSDs. They are SMSD, MMSD and MNMSD. If we got n partitions for an attribute, these measures are defined in formula 2, 3 and 4.

$$SMSD_n = \sum_{i=1}^n MSD_i \tag{2}$$

$$MMSD_n = \max_{i=1..n} MSD_i \tag{3}$$

$$MNMSD_n = \frac{\sum_{i=1}^n MSD_i}{n} \tag{4}$$

SMSD, MMSD, and MNMSD is the sum, the maximum, and the average of the list of MSDs, respectively.

3.2 OMVD: Optimizing MSD by a Genetic Algorithm

Here we propose our algorithm for optimizing the partitions with a genetic algorithm[11]. As usual, we introduce our genetic algorithm from the following viewpoints: individual coding, population initialization, operators, fitness function and termination criterion.

- **Individual coding:** Every individual represents a discretization scheme. We adopt binary coding, in which every binary bit is set to be 1/0 to represent the presence/absence of a potential partition. Every partition between two adjacent basic intervals is viewed as a potential cut-point. That is, the i^{th} bit represents whether the i^{th} basic interval is partitioned from the $(i + 1)^{th}$ basic interval. For convenience, we call a segment of a chromosome a *block* if it represents all potential cut-points of an attribute and it excludes the potential cut-points of another attribute. Thus, we get m *blocks* given m attributes. The length of a chromosome is $\sum_{i=1}^m nb_i - m$ bits given m attributes, where nb_i denotes the number of basic intervals of the i^{th} attribute. For example, the chromosome '101' with $nb_1=4$ and $m=1$ means there are two partitions and three intervals is formed after discretization.
- **Population initialization:** We randomly initialize every individual of population. As a result, those partitions are generated immediately without merging the basic interval step by step. The benefit we get from it is that there is no need to specify a minimum support difference and to select some candidate intervals to merge. As discussed in section 1, we avoid a difficult job. Besides, our approach enlarges the search space so that some better partitions can be found.
- **Selection operator:** We adopt tournament selection operator where k individuals are randomly selected from the population and play a tournament[11].
- **Crossover operator:** The multi-point crossover operator[11] is employed in OMVD. In fact, one-point crossover operator[11] is used for every *block* of a chromosome.
- **Mutation operator:** Every bit in each chromosome is conversed with a particular probability.
- **Fitness function:** The fitness of a chromosome is got by the sum of independent ranks of all its *blocks* in population. A *block* is ranked by MMSD, SMSD and the number of partitions. The reason we use these three measures is that we desire to find more significant, bigger MSDs with fewer partitions. We do not use MNMSD because it doesn't consider the significant difference. For example, there are two list of MSDs, $\{0.8, 0.1\}$ and $\{0.6\}$. We prefer the first case although a bigger MNMSD is got in the second case. The intervals formed by partitions are viewed as candidate contrast sets to calculate MSDs. Formally, we get the the list of MSDs for the i^{th} individual as $\{d_{11}^i, \dots, d_{1r_1}^i; \dots; d_{j1}^i, \dots, d_{jr_j}^i; \dots; d_{m1}^i, \dots, d_{mr_m}^i\}$, where r_j denotes the number of partitions on the j^{th} attribute. The MSDs of different attributes are separated into different sublists by semicolons. Then, the MMSD and SMSD can be calculated for each sublist. The ranks of them in population are got by sorting the those values by an ascending

order. The rank of the number of intervals is got by sorting the values by a descending order. After that, we sum the three ranks of all *blocks* to get the fitness of a chromosome. For example, we get a population with three chromosomes, each of which consists of two *blocks*. The lists of MSDs are listed in table 1.

Table 1. An example for computing the fitness of a chromosome(the rank of a sublist consists of the rank of MMSD, SMSD and the number of interval, from left to right)

index	sublists of MSD	ranks of sublist 1	ranks of sublist 2	fitness
1	0.5 0.3; 0.09 0.01	1 1 2	2 2 1	9
2	0.5 0.2 0.1; 0.07 0.05	1 1 1	1 3 1	8
3	0.6 0.1 0.1; 0.07	2 1 1	1 1 2	8

- **Termination criterion:** The Hamming distance between the best chromosome of the current generation and that of the last one is calculated once upon the population has finished a generation of evolution. If the distance is smaller than some threshold, evolution terminates.

4 Experiments

In this section, we conduct experiments with both synthetic and real datasets and compare the results of OMVD with those of MVD.

4.1 Experiments on Synthetic Datasets

We ran OMVD and MVD on two variable synthetic dataset to illuminate that OMVD can find better partitions than MVD in terms of MSDs.

Generation of Datasets. The datasets are same with those used in [3]. They are generated from a mixture of two-dimensional multivariate Gaussians. We refer to the two continuous variables as x_1 and x_2 . We used 1000 examples and equiprobable mixture components. The first dataset in Fig.1(a) shows the two multivariate Gaussians with similar covariance matrices but with a slight offset from each other. The second dataset in Fig.1(b) shows two multivariate Gaussians arranged in a cross.

Experiments on DS1. As a preparation of MVD and OMVD, we split every attribute of DS1 into 20 depth-equal basic intervals, respectively. The results of MVD with different thresholds on the DS1 are shown in table 2. With the threshold increasing, the number of partitions is decreasing. However, the MSDs did not increase significantly. We run OMVD for ten times due to its randomness. The population size is fixed to 40 individuals. The probabilities of crossover and mutation are set to be 0.8 and 0.1, respectively. A summary of the experiment results of OMVD is list in table 3. The values in it is averaged among the ten experiments. Having compared the results in table 2 with those in table 3, we know OMVD works better than MVD considering MSDs.

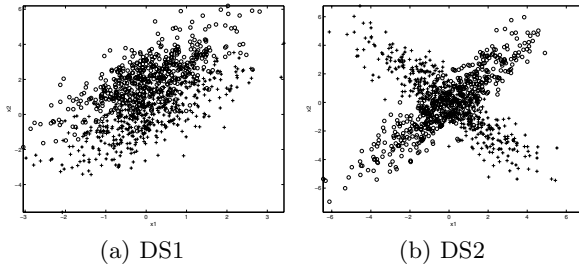


Fig. 1. Two synthetic datasets

Table 2. The results of MVD on DS1

	$\delta=0.01$		$\delta=0.02$		$\delta=0.03$
MSD list of x_1	0.1723	0.0650 0.0550 0.1426	0.1475	0.1475	0.1103
location of partitions x_1	-1.182	-0.0019 0.5599 1.365	-0.7159	0.8678	-0.2131
MSD list of x_2	0.1597	0.0600 0.1305	0.1233	0.0797	0.0978
location of partitions x_2	-1.2384	1.0714 3.2692	-0.4796	1.5268	0.6138

Table 3. The summary of results of OMVD on DS1

attribute	MMSD	MNMSD
x_1	0.2978	0.2318
x_2	0.3999	0.2741

Experiments on DS2. For DS2, we split the basic intervals as we did on DS1. The results are list in table 4. All the basic intervals are merged into one when $\delta=0.03$. OMVD is run for ten times on DS2 and the average values of the ten experiments are listed in table 5. The results are all better than those in table 3 because DS2 shows very significant interaction between x_1 and x_2 . Similarly with the results on DS1, it suggests that OMVD performs better than MVD on DS2.

In table 6, one complete result is randomly selected among the ten experiment results to compare with table 4. We can compare the semantic meanings of the results in terms of the contrast sets. For x_1 , the MMSD found by MVD($\delta = 0.01$) occurs between $[-6.438 -3.439]$ and $[-3.439 -1.971]$. It is brought by $x_2=[-7.688 -3.645]$. Differently from MVD, OMVD finds the MMSD, 0.676, between $[-0.780 2.03]$ and $[2.03 6.89]$. And it is brought by $[1.8771 3.573]$. It suggests that OMVD be more powerful than MVD on finding the affect of x_2 on x_1 . For x_2 , we can get a similar conclusion by comparison.

4.2 Experiments on IPUMS Dataset

In this subsection, MVD and OMVD are run with IPUMS[12] dataset. This dataset is available on <http://www.ipums.umn.edu/>. Our goal is to show that

Table 4. The results of MVD on DS2

	$\delta=0.01$	$\delta=0.02$	$\delta=0.03$
MSD sublist of x_1	0.317 0.113 0.056 0.1 0.1 0.213	0.247 0.086	/
MNMSD(MMSD) of x_1	0.1502(0.317)	0.1669(0.247)	/
location of partitions on x_1	-3.439 -1.971 -0.230 1.393 2.030 3.330	-2.569 1.087	/
MSD sublist of x_2	0.327 0.122 0.12 0.08 0.105 0.273	0.2230.230	/
MNMSD(MMSD) of x_2	0.171(0.327)	0.226(0.23)	/
location of partitions on x_2	-3.645 -2.378 -1.484 0.774 2.330 3.573	-2.872 2.929	/

Table 5. The summary of results of OMVD on DS2

attribute	MMSD	MNMSD
x_1	0.463	0.3503
x_2	0.476	0.3113

Table 6. One result of OMVD on DS2

MSD sublist of x_1	0.434 0.195 0.096 0.676
location of partitions on x_1	-2.5696 -1.0531 -0.78077 2.0304
MSD sublist of x_2	0.363 0.497 0.171 0.227 0.298
location of partitions on x_2	-3.6456 -1.7895 1.4772 1.8771 3.573

Table 7. The summary of results of OMVD with IPUMS

attribute	MMSD	MNMSD
age	0.5293	0.3740
edurec	0.6359	0.4752
inctot	0.4691	0.3576

OMVD is feasible and can get the more interesting discretizations than MVD in practice. In this experiment, we only select three attributes. They are *age*, *edurec* and *inctot*. We randomly sample 3,000 out of total 88,443. In IPUMS, any value of *age* is an integer in [0 90]. It is split into 90 width-equal basic intervals. The value of *edurec* is an integer in [0 9] which is split into 10 width-equal basic intervals to reserve its semantic meaning. For *inctot*, it is continuous and its range is split into 20 depth-equal basic intervals. OMVD is run for ten times on the dataset and the result is summarized in table 7. We randomly select one from the ten results to compare with MVD in Fig.2. And the MSDs are list in table 8. In the following paragraphs, we will focus our discussion on comparing the semantic meanings of the discreziations made by OMVD with those of discretizations by MVD.

Age. In Fig. 2, OMVD gets fewer partitions than MVD(0.01) but more than MVD(0.03). Compared with MVD(0.01), OMVD takes the range between 2 and 34 as a complete interval while both of MVD methods discretize it into several

Table 8. The Comparison of MSDs got by OMVD with those by MVD on IPUMS(the values in bracket are the minimum different threshold)

attribute	methods	MSD sublist									
age	omvd	0.959 0.426 0.101 0.287									
	mvd(0.01)	0.647	0.772	0.568	0.412	0.414	0.428	0.080	0.079	0.103	0.107
	mvd(0.03)	0.602 0.599 0.618									
edurec	OMVD	0.619 0.441 0.290									
	mvd(0.01)	0.619	0.434	0.464	0.241	0.266	0.059	0.153			
	mvd(0.03)	0.619 0.434 0.464 0.367									
inctot	OMVD	0.726 0.387 0.092 0.270 0.325									
	mvd(0.01)	0.179 0.137 0.189 0.200									
	mvd(0.03)	0.183 0.348									

intervals. OMVD and MVD(0.01) get the same first intervals from left and it is [0 2) while they get very different MSD on this partition. Without any question, that is caused by the right next interval and the candidate contrast sets. It is interesting for us to find that MSDs of OMVD and MVD(0.01) are both caused by the combination of *edurec*=0 and *inctot*=[0 1,996). This fact shows that the biggest difference between those below two years old and those between 2 and 43 is that the latter have education attainment records and have positive income. Although that difference still exists between [0 2) and [2 6) got by MVD(0.01), it is not so much significant(0.647 is a little smaller when compared with 0.959).

Another big difference between the discretizations made by OMVD and MVD is that the former gets a narrow interval at [34 36). The reason that makes it separated from its left neighbor is *inctot*=[0 1,996). It suggests that there is a bigger proportion of people in *age*=[2 34) having their income in that range than that of those in *age*=[34 36). The reason that [34 36) is separated from [36 64) is that the proportion of those having income in [4,548 14,000) to those in *age*=[34 36) is larger than in [36 64).

edurec. There are no significant differences between MSD list of OMVD and that of MVD. A noticeable difference between partitions of OMVD and MVD is that the former takes the middle education records(below grade 12) as a interval and partitions nicely on the range with high education records(college records). Considering the contrast sets for OMVD and MVD, we get a similar conclusion. That is, the major support differences among middle or low education records are caused by different phases of age. But for high education records, such as 8(1 to 3 years of college) and 9(4+ years of college), the major difference between them is caused by different interval on *income* instead of *age*. It suggests that those people having education above college should have more chances to get high income.

inctot. Compared with MVD, OMVD partitions more nicely on the range of -5,484 and 14,000 and it brings much larger MSDs. We will find out the reasons from the contrast sets. All MSDs of MVD are brought by the basic intervals on the high range of *edurec*. Differently from MVD, the partitions of OMVD on

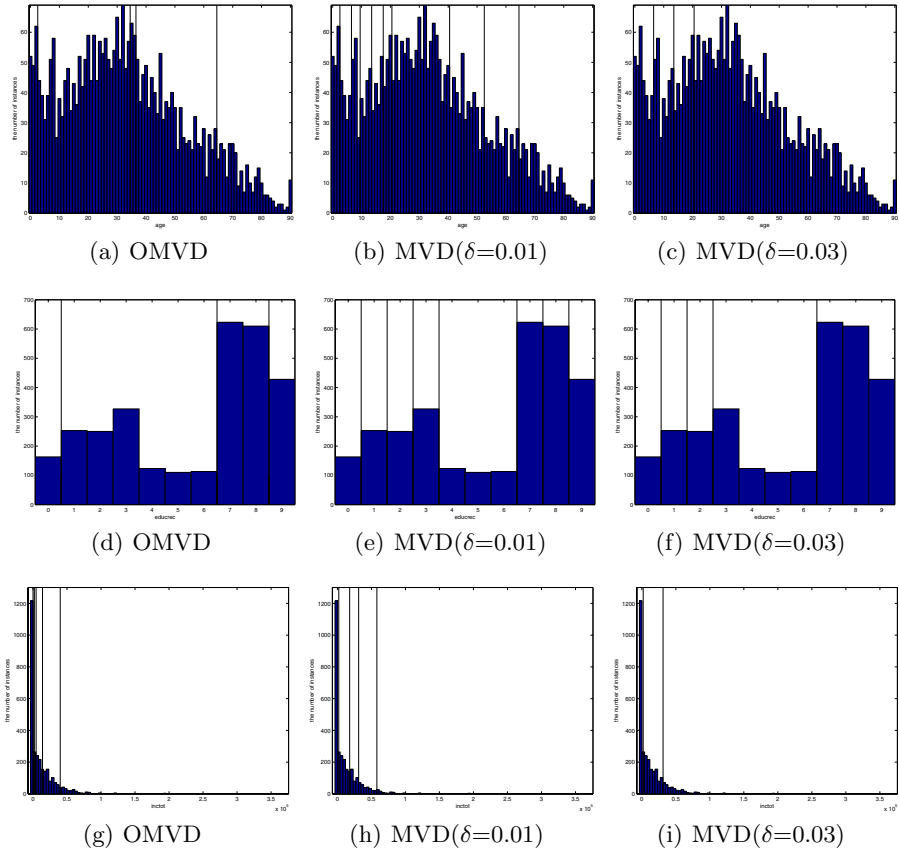


Fig. 2. Experiment results on IPUMS(a, b, and c for *age*; d, e, and f for *educrec*; g, h, and i for *inctot*. The horizontal axis represents the value of attribute and the vertical axis represents the number of instances falling into every basic interval.)

low range are caused by *age* and those on middle and high ranges are caused by *educrec*. The biggest MSD in the list of OMVD lies between $(-5,484 \ 0]$ and $[0 \ 1,996)$. The contrast set that brings it is $age=[2 \ 34)$. This looks more rational than MVD. The fourth partition in Fig.2(g) separates $[4,548 \ 1,4000)$ from $[14,000 \ 40,001)$. Its corresponding contrast set is $educrec=[1 \ 6)$. It tells us that a smaller proportion of those with higher income have their education records below 6. Similarly, the fifth partition is also caused by an interval on *educrec*. But it is with high education record($educrec=9$). We can get a conclusion that education contributes much to income for those having their incomes above some threshold(such as 14,000).

4.3 Experiments on Three Real Datasets

To show the scalability of OMVD on the large-scale database, we run OMVD on three real datasets that are selected from UCI Machine Learning Database

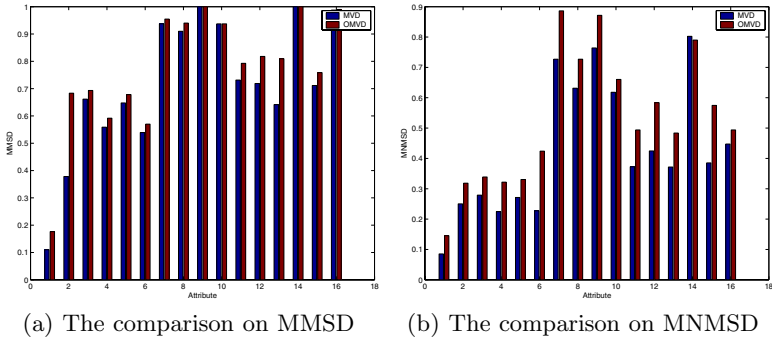


Fig. 3. Experiment results on IS

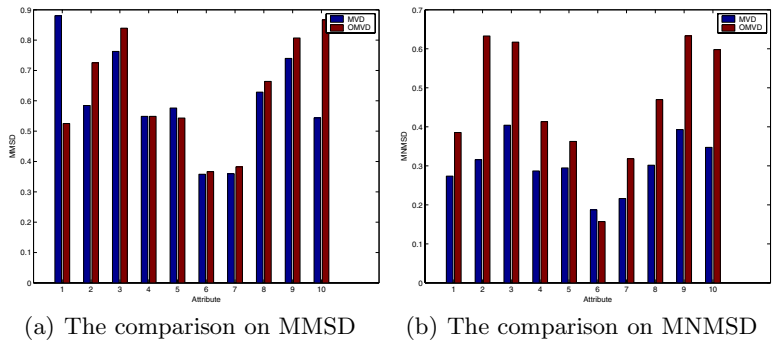


Fig. 4. Experiment results on PBC

Repository. They are Image Segmentation of Statlog Project Databases(IS), Page Blocks Classification Database(PBC), and Ionosphere Database(ID). All of the datasets can be downloaded from [http://www.ics.uci.edu / mlearn /ML-Summary.html](http://www.ics.uci.edu/ml-learn/ML-Summary.html).

For IS, we take all 2130 instances and 16 attributes(the 3rd, 4th, 5th and 20th attributes are excluded). For PBC, we take all 5473 instances and 10 attributes(the last attribute is excluded). For ID, we take all 351 instances and 32 attributes(the first, the second , and the last attributes are excluded). Because the fitness has to be computed for every chromosome during every generation, the time for computing fitness is the most important factor to save execution time of OMVD. In the following experiments, we randomly select only two other attributes and take their intervals as candidate contrast sets when we compute the values of MSD for one attribute.

We run MVD once and OMVD for ten times on each dataset. The values of MNMSD and MMSD for every attribute got by MVD($\delta=0.01$) and OMVD(averaged among the ten experiments) are compared in Fig. 3, Fig. 4, and Fig. 5.

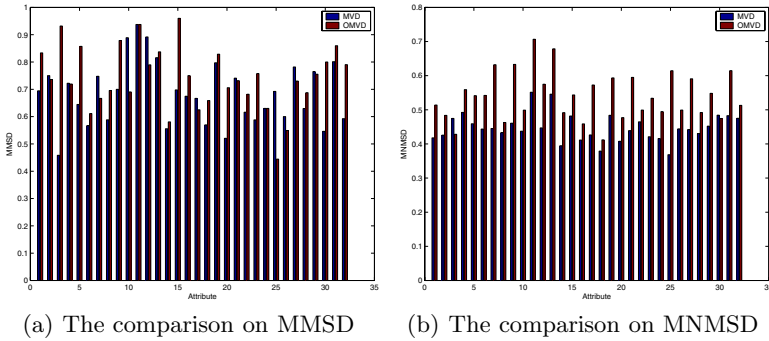


Fig. 5. Experiment results on ID

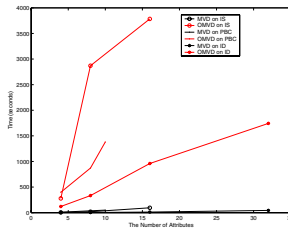


Fig. 6. The execution time of MVD and OMVD on IS, PBC and ID

Obviously, the results got by OMVD are better than those by MVD on all experiments. This stems from the stronger search power of OMVD. Then OMVD and MVD are run on every datasets with different subset of attributes to show their scalability. The execution time is show in Fig. 6. It shows that OMVD has a good scalability with the number of attributes. We also notice that OMVD is not seriously affected by the number of instances.

5 Conclusion

In this paper, we examine the limitations of MVD on the power of testing support difference. Then we make an improvement on MVD by maximizing the support difference and simultaneously set the partition for all attributes. In OMVD, we need not to set the threshold and do realize the simultaneous discretization. As a result, OMVD is more powerful than MVD on testing support differences. By analyzing the experiment results with synthetic and real datasets, we notice that MSDs got by OMVD are more interesting than those by MVD. By the experiments on large datasets with decades of attributes, we show the scalability of OMVD. In conclusion, OMVD can work better on finding the more interesting discretizations in terms of the semantic meanings of those partitions.

References

1. Agarwal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proc. ACM SIGMOD International Conference on Management of Data, Washington, D.C. (1993) 207 – 216
2. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco, CA, USA (1993)
3. Bay, S.D.: Multivariate discretization for set mining. *Knowledge and Information Systems* **3** (2001) 491 – 512
4. Bay, S.D., Pazzani, M.J.: Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery* **5** (2001) 213 – 246
5. Kwedlo, W., Kretowski, M.: An evolutionary algorithm using multivariate discretization for decision rule induction. In: *Principles of Data Mining and Knowledge Discovery*. (1999) 392 – 397
6. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In Jagadish, H.V., Mumick, I.S., eds.: *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Quebec, Canada (1996) 1 – 12
7. Miller, R.J., Yang, Y.: Association rules over interval data. In: *Proceedings ACM SIGMOD International Conference on Management of Data*. (1997) 452 – 461
8. Monti, S., Cooper, G.F.: A latent variable model for multivariate discretization. In: *The 7th Int. Workshop Artificial Intelligence and Statistics*, Fort Lauderdale (1999)
9. Ludl, M.C., Widmer, G.: Relative unsupervised discretization for association rule mining. In: *In Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Berlin, Germany, Springer (2000)
10. Mehta, S., Parthasarathy, S., Yang, H.: Toward unsupervised correlation preserving discretization. *IEEE Transactions on Knowledge and Data Engineering* **17** (2005) 1174– 1185
11. Eiben, A., Smith, J.: *Introduction to Evolutionary Computing*. Springer (2003)
12. Ruggles, S., Sobek, M., Alexander, T., et. al: *Integrated public use microdata series: Version 2.0 minneapolis: Historical census projects* (1997)

ZED: Explaining Temporal Variations in Query Volume

Maojin Jiang¹, Shlomo Argamon¹, Abdur Chowdhury², and Kush Sidhu²

¹ Laboratory of Linguistic Cognition, Illinois Institute of Technology, Chicago, USA

jianmao@iit.edu, argamon@iit.edu

² America Online, Inc., USA

Cabdur@aol.com, KSidhu35@aol.com

Abstract. We hypothesize that the variance in volume of *high-velocity queries* over time can be explained by observing that these queries are formulated in response to events in the world that users are interested in. Based on it, this paper describes a system, ZED, which automatically finds explanations for high velocity queries, by extracting descriptions of relevant and temporally-proximate events from the news stream. ZED can thus provide a meaningful *explanation* of what the general public is interested in at any time. We evaluated performance of several variant methods on top velocity “celebrity name” queries from Yahoo, using news stories from several sources for event extraction. Results bear out the event-causation hypothesis, in that ZED currently finds acceptable event-based explanations for about 90% of the queries examined.

1 Introduction

Web search is the second most popular activity on the Internet, exceeded only by e-mail. What people are searching for changes over time, due to cycles of interest and events in the world [1]. Of particular interest are *high-velocity* queries, which show a sharp increase in popularity over a short time (*velocity* refers to relative change in number of searches of a query during a given period.). Such queries may reflect important specific events in the world, as people hear about the current news and then search for more information on significant events (Fig. 1).

This paper describes a new system, ZED¹, which, given a celebrity name which is a top velocity query on a given day, automatically produces a description of a recent event involving that celebrity which is likely to be of widespread interest. We call such events *query-priming events*. The set of such explanations can serve on its own as a summary of the current ‘zeitgeist’, or may be used to index into more information about the events and individuals currently deemed most interesting by the querying public.

To the best of our knowledge, our specific task of explaining peak queries has not been addressed previously. We briefly discuss here prior work on extractive text summarization which ZED’s event extraction algorithm is based on. One relevant work is multiple document summarization [2] and its application to news [3]. Event-based text summarization [4,5] focuses on news summarization, which addresses the issues to detect events in news and to maintain a complete description of event constituents. Another relevant work on query-based summarization [6,7] takes a user query into account (as in our task) such that summarization is made only on relevant text.

¹ Zeitgeist Event Detector.

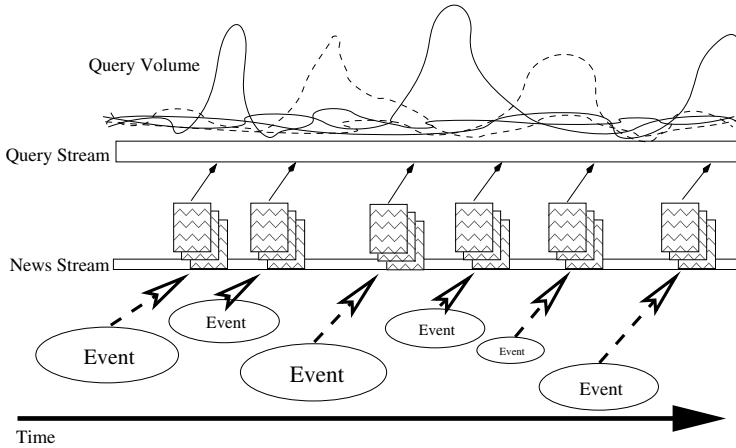


Fig. 1. The ‘event causation’ model of query volume variability. Events in the real world (of greater or lesser significance) cause news stories to be written, which spur user interest in the events, leading to sharp volume increases for related queries.

2 The Approach

Our approach is to enhance traditional text summarization methods with specific cues for finding query-focused sentences describing relevant query-priming events. We first give an overview of the ZED system architecture, then discuss its components in more detail, including variant methods for certain subtasks.

2.1 System Architecture

ZED comprises three components: *document indexing*, *query searching*, and *event extraction*. The system works as follows. First, news stories coming in from the stream are indexed. Then, each top velocity query containing a celebrity name is fed into a search engine to retrieve the most relevant news stories, restricting attention to stories which (a) came in within the last k days, and (b) contain somewhere the full query. After a set of most-relevant stories is obtained, each of these stories file split into individual sentences, and certain irrelevant sentences filtered out (e.g., numerical lists). Then sentences are ranked for rough relevance, based on a linear combination of three measures (as in MEAD [2]): *centroid distance*, *query relevance*, and *sentence position*. Sentences with high overall scores are then processed by one of several *sentence selection* criteria to choose the ‘best’ sentence purporting to describe the event which prompted the input query to suddenly rise in volume. Finally, the headline from the story in which this top sentence is found is returned together with the sentence as the explanation for the query under consideration. The various modules within ZED are described in more detail next.

2.2 News Story Retrieval

The AIRE IR engine [8] is used to index and retrieve news stories. News story files are parsed, tokenized, and downcased with stopwords removed, but without stemming.

Token positions are recorded to facilitate phrase searching for the full name in the query.

For retrieval, only queries which constitute a celebrity name *in toto* are considered. Query tokens are therefore not stemmed or elided. Only story files containing the full query as a phrase at least once are considered as possibly relevant. This serves to remove many false positives, and is based on the assumption that a complete news report on a person should mention the full name at least once. Also, only stories with a post date with the two days prior to the query date are considered since we hypothesize that a top-velocity name query reflects only latest events in news and two days should be a reasonable approximation. Relevance ranking is then done using the individual query terms via the standard bm25 [9] ranking function, giving headline words more weight by adding 10 to their term frequencies. This simple heuristic rule originates in our observation that a news report is more focused on events of a celebrity if his or her name appears in the headline. The top 10 retrieved documents are then passed on to the next module.

2.3 Sentence Splitting and Filtering

All relevant news story files returned from previous step are then segmented into sentences. Since many articles contain ‘sentences’ that aren’t really descriptive (lists of sports scores, for example), we remove at this stage all sentences which:

- Contain fewer than five words (this removes too short sentences in that they often do not contain enough information about an event), **or**
- Contain fewer than two function words, based on a preconstructed list of 571 common English function words (this ensures the sentence is regular English text), **or**
- Contain no query terms.

The position $\text{pos}(s)$ of each sentence s in the story is then recorded. This set of tagged sentences is then passed on to the sentence ranking module.

2.4 Sentence Ranking

The set of candidate event description sentences is then ranked by a quick approximate quality measure based on previous methods for multi-document summarization. We use a combination of three measures of sentence quality that are based on those used by the MEAD summarization system [2]:

Centroid similarity: Here we compute the word-frequency ‘centroid’ of the retrieved set and measure relevance of each sentence based on its similarity to this centroid. The centroid is computed by computing, for each content word in the retrieved set, the sum of its tf-idf score over all 10 documents. The 80 words with the highest scores are retained, and the resultant vector of tf-idf sums is the ‘centroid’ of the document set. Then the centroid similarity $f_C(s)$ for each sentence s is computed as its cosine distance from the centroid.

Query similarity: The query similarity measure $f_q(s)$ evaluates how directly relevant a sentence s is to the input query. It is calculated as the tf-idf cosine distance between a sentence and the input query.

Sentence position: The third measure is a LEAD-like measure of importance based on the idea that the more important sentences tend to appear near the beginning of news stories. The position measure, $f_P(s)$, is defined as reciprocal of square root of $pos(s)$, giving higher values to sentences nearer the beginning of a story.

After these three quality measures are computed for each sentence s , its overall score is computed by a manually-adjusted linear combination of the measures:

$$f(s) = 8f_C(s) + 12f_q(s) + f_P(s) \quad (1)$$

As in MEAD [2], the parameters in (1) are obtained by manually examining different scores of some sample sentences by using different coefficient values. In the future, optimal values of them may be set by applying some machine learning approach to well-established human-made summaries. The sentences are then ranked according to $f(s)$, duplicates are removed, and the top 10 sentences selected for further processing.

2.5 Sentence Selection

The next step is to select, from the candidate sentences provided by sentence ranker, the most likely sentence to constitute an effective explanation as a query-priming event description. We compare three methods for selecting the final event describing sentence:

Ranking: Select the sentence with the highest ranking score $f(s)$.

Recency: Select the most recent sentence from the candidate set of most highly ranked sentences, breaking ties by $f(s)$.

Grammar: This strategy uses a heuristic method for evaluating sentences based on its syntactic structure. Robust Accurate Statistical Parsing (RASP) system [10] is used to parse each candidate sentence, giving a syntactic parse tree represented as a set of grammatical dependency relations between sentence words. Based on the parse tree, three features of the sentence's syntactic structure are determined. First is the sentence's *complexity*—if it consists of a single clause it is 'simple', otherwise it is 'complex'. Second is the *position* of the query term(s) in the sentence—whether in the main clause or a sub-clause. Third is the syntactic *role* of the query term(s) in the sentence, whether in the subject, object, a prepositional phrase ('pp'), or other syntactic constituent.

The idea is that these features give some indication of the usefulness of the sentence under consideration as a description of a query-priming event. Based on this sort of consideration, we manually constructed the preference ordering for the various possible feature combinations, as shown in Table 1.

After the 'best' sentence is selected according to one of the above criteria, it is returned with the headline of its story as the final event description.

2.6 Comparison Baselines

ZED was compared to two baseline methods. The first, termed FirstSentence, based on the assumption that journalists will tend to put a summary sentence at the beginning of a news report, simply chooses the first sentence from the most relevant story, together with the story's headline, as the event description. The second baseline method,

Table 1. Preference order for sentences based on grammatical features. The candidate sentence with the highest position in this order is chosen as a priming event description.

Complexity	Position	Function	Complexity	Position	Function	Complexity	Position	Function
1. simple	main	subject	5. simple	main	pp	9. complex	sub	pp
2. complex	main	subject	6. complex	main	pp	10. simple	main	other
3. simple	main	object	7. complex	main	object	11. complex	main	other
4. complex	sub	subject	8. complex	sub	object	12. complex	sub	other

FirstRelevant, incorporates the constraint that the explanation contain the name in the query. It chooses the earliest sentence from the top-ranked document that contains at least one of the query terms, together with the story's headline, as the event description.

3 Evaluation

3.1 Corpus, Testbed and Assessment Metrics

The corpus we used consisted of news stories from the month of September 2005, downloaded from six on-line news-sources: AP, BBC, CNN, NYTimes, Time and Yahoo, in four relevant categories: Top Stories, Entertainment, Sports, and Obituaries. There were a total of 25,142 articles in the full corpus, giving nearly 900 on average per day. In experiment, after top 10 relevant stories are split into sentences and after sentence filtering, on average, a query gets 24 candidate sentences for ranking and selection.

We constructed an evaluation testbed by taking all celebrity name queries from Yahoo Buzz's "top velocity" queries for the week of 5 September through 11 September, 2005, termed *FirstWeek* and comprising 141 queries, and the week of 24 September through 30 September, 2005 termed *LastWeek* and comprising 128 queries.

Summaries generated by each of the five methods above (three ZED variants and the two baselines) were evaluated (using randomized ordering) by two human assessors (referred to as A and B respectively). Three types of metrics were used for evaluation:

Relevance: This is a traditional binary metric, wherein each explanation is adjudged either relevant or irrelevant to its query. The relevance criterion was whether the summary refers (even tangentially) to a possible priming event for that query.

Ranking: In this metric, the five explanations from different methods that were returned for a query were ranked from 1 to 5, with 1 being the best, and 5 being the worst. Irrelevant explanations were not ranked. Identical explanations, which were rather common, received the same rank, then the next one down received the appropriate rank; e.g., when two explanations both received rank 1 next best received rank 3.

Quality: Finally, the quality of each explanation was evaluated numerically according to three criteria: *completeness*, *sufficiency*, and *conciseness*. Each of these was evaluated on a coarse 3-point scale (from 0 to 2), as follows:

Completeness refers to how fully the explanation describes an event involving the query individual. A score of 2 meant a complete explanation of some event involving the individual (even circumstantially), a score of 1 meant that some essential

information about the event was missing, and a score of 0 meant that it did not describe an event involving the query individual at all.

Sufficiency refers to how strongly the event described would motivate people (those interested in the queried individual) to use the specific query, based on the importance of the event and centrality of the individual's participation in it. A score of 2 meant a strong motivation for querying on that individual, 1 meant that there would be just weak possible motivation to query, and 0 meant that there would be no such motivation.

Conciseness refers to how much extraneous information is included in the explanation besides the relevant event (if present). A score of 2 meant that nearly all the information in the explanation was about the relevant event, 1 meant that there was much irrelevant information, though the explanation was mainly about the relevant event, and 0 meant that the central focus of the explanation was on irrelevant matters.

These three values were summed, to give a total *Quality* score, ranging from 0 to 6.

3.2 Results and Discussion

Event descriptions generated by the five methods for each of the two weeks' data were evaluated separately, so we could also evaluate consistency of system performance for news of somewhat different periods of time.

We first evaluated the consistency of evaluation between the two raters, examining agreement percentage and Cohen's kappa [11]. For relevance, agreement percentages were 97.6% and 89.1% for FirstWeek and LastWeek, respectively, corresponding to kappa values of 0.91 and 0.69. For ranking, where ratings from an ordered set of values, linearly-weighted kappa [12] was used. Agreement percentages were lower at 73.0% and 78.3% for FirstWeek and LastWeek, with linearly-weighted kappas of 0.63 and 0.65. These results show substantive agreement between the raters; in what follows, we present results just for rater A.

Table 2 shows the precision for each variant, where precision is defined as the ratio of the number of relevant explanations returned over the total number of explanations returned. Results are slightly different for the two weeks studied. For FirstWeek, Grammar is beaten by Ranking, though the difference is small; all ZED variants do clearly improve over the baselines. On LastWeek, Grammar has a definite advantage, and the other two ZED variants are not clearly better than just choosing the first-ranked relevant sentence. Overall, the Grammar variant perhaps has a slight edge. It is clear, however, that using the first sentence of the most relevant document is not useful.

Histograms showing the rank distributions for the five variants are given in Fig. 2. We first of all see that, as precision also indicated, the FirstSentence heuristic does not work very well at all. Among the other four methods, however, we see a difference between the two weeks that were evaluated, with Grammar dominating for FirstWeek and FirstRelevant dominating for LastWeek. The good performance of FirstRelevant may lie in the hypothesis that we infer that many reporters tend to mention the name of major character in the first sentence that describes the major event in a news report. However, it is unclear to what extent these differences are really significant, particularly in view of the results of the quality evaluation, given in Table 2. These results show a different pattern, with Grammar attaining much higher quality for LastWeek, while Ranking is

Table 2. Average overall explanation precision (as well as with both week’s data pooled), quality, and average completeness, sufficiency, and conciseness, for the five query explanation methods

Method	FirstWeek					LastWeek					Pooled Prec.
	Prec.	Qual.	Comp.	Suff.	Conc.	Prec.	Qual.	Comp.	Suff.	Conc.	
Grammar	0.972	3.14	1.03	1.24	0.87	0.852	3.65	1.13	1.28	1.23	0.915
Recency	0.972	3.13	1.09	1.21	0.83	0.828	3.41	1.08	1.19	1.14	0.903
Ranking	0.979	3.29	1.15	1.25	0.89	0.813	3.59	1.12	1.25	1.21	0.900
FirstRelevant	0.950	3.27	1.15	1.26	0.86	0.828	3.63	1.16	1.24	1.23	0.892
FirstSentence	0.234	1.13	0.37	0.46	0.3	0.375	1.75	0.57	0.62	0.56	0.301

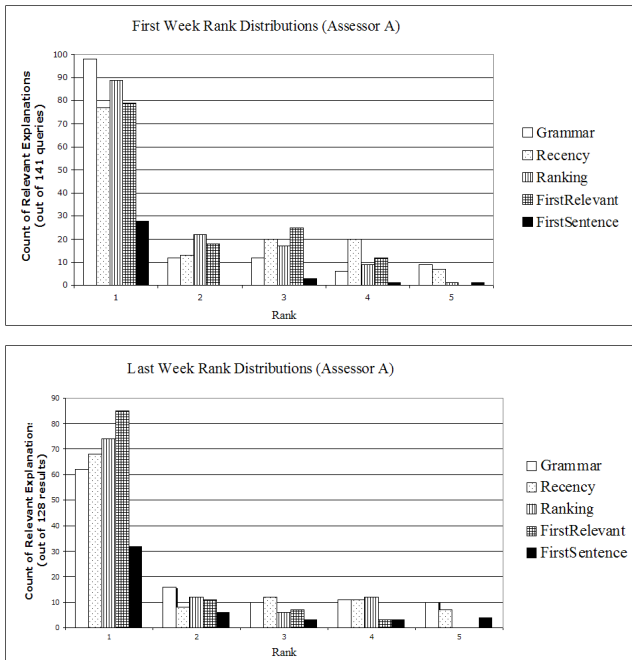


Fig. 2. Histograms of the respective ranks (by assessor A) of the explanations generated by the five event extraction methods for both weeks’ data

preferred somewhat for FirstWeek. Grammar’s quality dominance in LastWeek is due to greater sufficiency and conciseness, perhaps attributable to its ability to distinguish simple from complex sentences, and to ensure that the query appears in a salient syntactic position in the sentence. Regardless, all of the methods (other than FirstSentence) appear to find reasonable event descriptions, on average.

Overall, the three methods implemented in ZED outperform both baselines, though only FirstRelevant is a competitive baseline. FirstSentence’s abysmal performance indicates that a LEAD-like system will not work for this task. Among ZED’s three methods, however, results are inconsistent, though Recency does seem less useful than the more content-based measures. Grammatical cues do seem to provide some particular lever-

age, though more work is needed to elucidate this point. Future work will clearly need to address the development of larger evaluation sets for this task as well as examining inter-rater reliability measures for the evaluations.

4 Conclusions

We have presented ZED, a system which addresses the novel task of finding explanations of query-priming events in a news stream. This system thus provides an alternative view of “What is the interesting news today?” based on what recent events users as a whole have found most compelling. In the future, we intend to explore methods for improving the quality of ZED’s event descriptions. A more precise characterization of the circumstances in which one or another of the selection methods is preferred, if one can be found, may lead to improvements (for example, by helping us refine syntactic preferences). Also, lexical cues (such as ‘died’, ‘just released’) may help the system to recognize ‘significant’ events. Supervised machine learning may also be applied to build better models to combine evidence from different cues.

References

1. Chien, S., Immorlica, N.: Semantic similarity between search engine queries using temporal correlation. In: Proc. WWW-05, Chiba, Japan (2005) 2–11
2. Radev, D., Blair-Goldensohn, S., Zhang, Z.: Experiments in single and multidocument summarization using MEAD. In: Proc. Document Understanding Conference. (2001)
3. Radev, D.R., Blair-Goldensohn, S., Zhang, Z., Raghavan, R.S.: Newsinessence: a system for domain-independent, real-time news clustering and multi-document summarization. In: Proceedings of HLT '01. (2001) 1–4
4. Filatova, E., Hatzivassiloglou, V.: Event-based extractive summarization. In: ACL Workshop on Summarization, Barcelona, Spain (2004)
5. Vanderwende, L., Banko, M., Menezes, A.: Event-centric summary generation. In: Proc. Document Understanding Conference at HLT-NAACL, Boston, MA (2004)
6. Saggion, H., Bontcheva, K., Cunningham, H.: Robust generic and query-based summarization. In: Proceedings of EACL '03. (2003) 235–238
7. Amini, M.R.: Interactive learning for text summarization. In: Proceedings of the PKDD'2000 Workshop on Machine Learning and Textual Information Access. (2000) 44–52
8. Chowdhury, A., Beitzel, S., Jensen, E., Sai-lee, M., Grossman, D., Frieder, O., et. al.: IIT TREC-9 - Entity Based Feedback with Fusion. TREC-9 (2000)
9. Robertson, S.E., Walker, S., Hancock-Beaulieu, M.: Experimentation as a way of life: Okapi at TREC. *Information Processing and Management* **36** (2000) 95–108
10. Briscoe, E.J., Carroll, J.: Robust accurate statistical annotation of general text. In: Proceedings of LREC. (2002) 1499–1504
11. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*. **20** (1960) 37–46
12. Maclure, M., Willett, W.: Misinterpretation and misuse of the kappa statistic. *American Journal of Epidemiology*. **126** (1987) 161–169

An Effective Multi-level Algorithm for Bisecting Graph

Ming Leng and Songnian Yu

School of Computer Engineering and Science,
Shanghai University, Shanghai, PR China 200072
lengming@graduate.shu.edu.cn
snyu@staff.shu.edu.cn

Abstract. Clustering is an important approach to graph partitioning. In this process a graph model expressed as the pairwise similarities between all data objects is represented as a weighted graph adjacency matrix. The min-cut bipartitioning problem is a fundamental graph partitioning problem and is NP-Complete. In this paper, we present an effective multi-level algorithm for bisecting graph. The success of our algorithm relies on exploiting both Tabu search theory and the concept of the graph core. Our experimental evaluations on 18 different graphs show that our algorithm produces excellent solutions compared with those produced by MeTiS that is a state-of-the-art partitioner in the literature.

1 Introduction

An important application of graph partitioning is data clustering using a graph model [1], [2]. Given the attributes of the data points in a dataset and the similarity or affinity metric between any two points, the symmetric matrix containing similarities between all pairs of points forms a weighted adjacency matrix of an undirected graph. Thus the data clustering problem becomes a graph partitioning problem [2]. The *min-cut bipartitioning problem* is a fundamental graph partitioning problem and is NP-Complete [3]. Because of its importance, the problem has attracted a considerable amount of research interest and a variety of algorithms have been developed over the last thirty years [4],[5]. The survey by Alpert and Kahng [6] provides a detailed description and comparison of various such schemes which include *move-based* approaches, *geometric representations*, *combinatorial* formulations, and *clustering* approaches.

A graph $G=(V,E)$ consists of a set of vertices V and a set of edges E such that each edge is a subset of two vertices in V . Throughout this paper, n and m denote the number of vertices and edges respectively. The vertices are numbered from 1 to n and each vertex $v \in V$ has an integer weight $S(v)$. The edges are numbered from 1 to m and each edge $e \in E$ has an integer weight $W(e)$. A decomposition of a graph V into two disjoint subsets V_1 and V_2 , such that $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$, is called a *bipartitioning* of V . Let $S(A) = \sum_{v \in A} S(v)$ denote the size of a subset $A \subseteq V$. Let $ID(v)$ be denoted as v 's *internal degree* and is equal to the sum of the edge-weights of the adjacent vertices of v that

are in the same side of the partition as v , and v 's *external degree* denoted by $ED(v)$ is equal to the sum of edge-weights of the adjacent vertices of v that are in the different side of the partition. The *cut* of a *bipartitioning* $P=\{V_1, V_2\}$ is the sum of weights of edges which contain two vertices in V_1 and V_2 respectively. Naturally, vertex v belongs at boundary if and only if $ED(v)>0$ and the *cut* of P is also equal to $0.5\sum_{v\in V}ED(v)$. Given a balance constraint r , the *min-cut bipartitioning problem* seeks a solution $P=\{V_1, V_2\}$ that minimizes $cut(P)$ subject to $(1-r)S(V)/2 \leq S(V_1), S(V_2) \leq (1+r)S(V)/2$. A *bipartitioning* is *bisection* if r is as small as possible. The task of minimizing $cut(P)$ can be considered as the *objective* and the requirement that solution P will be of the same size can be considered as the *constraint*.

Most existing partitioning algorithms are heuristics in nature and they seek to obtain reasonably good solutions in a reasonable amount of time. Kernighan and Lin (KL) [4] proposed a heuristic algorithm for partitioning graphs, which requires $O(n^2 \cdot \log(n))$ computation time. The KL algorithm is an iterative improvement algorithm that consists of making several improvement passes. It starts with an initial bipartition $\{A, B\}$ and tries to improve it by every pass. A pass consists of the identification of two subsets of vertices $A' \subset A$ and $B' \subset B$ of equal size such that can lead to an improved partition if the vertices in the two subsets switch sides. Fiduccia and Mattheyses (FM) [5] proposed a fast heuristic algorithm for bisecting a weighted graph by introducing the concept of cell *gain* into KL algorithm. As problem sizes reach new levels of complexity recently, a new class of graph partitioning algorithms have been developed that are based on the multilevel paradigm. The multilevel graph partitioning schemes include three phases [7],[8],[9]. The *coarsening phase* is to reduce the size of the graph by collapsing vertex and edge until its size is smaller than a given threshold. The *initial partitioning phase* is to compute initial partition of the coarsest graph. The *uncoarsening phase* is to project successively the partition of the smaller graph back to the next level finer graph while applying an iterative refinement algorithm.

In this paper, we present a multilevel algorithm which integrates an effective matching-based coarsening scheme and a new refinement approach. Our work is motivated by the multilevel partitioners of Saab who promotes locked vertex to free for refining the partition in [9] and Karypis who introduces the concept of the graph *core* for coarsening the graph in [10] and supplies **MeTiS** [7], distributed as open source software package for partitioning unstructured graphs. We test our algorithm on 18 graphs that are converted from the hypergraphs of the ISPD98 benchmark suite [11]. Our experiments show that our algorithm produces partitions that are better than those produced by **MeTiS** in a reasonable time.

The rest of the paper is organized as follows. Section 2 provides some definitions and describes the notation that is used throughout the paper. Section 3 briefly describes the motivation behind our algorithm. Section 4 presents an effective multilevel algorithm for *bisecting* graph. Section 5 experimentally eval-

uates the algorithm and compares it with **MeTiS**. Finally, Section 6 provides some concluding remarks and indicates the directions for further research.

2 Motivation

During each pass of KL and FM, they have the same restriction that each vertex can move only once and the restriction may prevent the exploration of certain promising region of the search space. In the terminology of Tabu Search [12], the KL and FM strategy is a simple form of Tabu search without aspiration criterion whose prohibition period is fixed at n . In [9], Saab introduces the concept of *forward move step* and *restore balance step* for refining the partition and adopts aspiration criterion to allow a locked vertex to become free in the same passes of ALG2 algorithm. As a consequence of allowing locked vertices to move in an iterative pass, ALG2 algorithm can explore a certain promising regions of the search space. However, Saab limits the vertices *move-direction* of two kinds of steps in $A \rightarrow B$ and $B \rightarrow A$ respectively and is obliged to adopt two different aspiration criterions for two kinds of steps respectively to avoid cycling issue.

In both FM and ALG2 refinement algorithms, we have to assume that all vertices are free and insert the *gain* of all vertices in free bucket. However, most of *gain* computations are wasted since most of vertices moved by FM and ALG2 refinement algorithms are boundary vertices that straddle two sides of the partition. As opposed to the non-boundary refinement algorithms, the cost of performing multiple passes of the boundary refinement algorithms is small since only boundary vertices are inserted into the bucket as needed and no work is wasted. In [7], the boundary KL (BKL) refinement algorithm presented by Karypis swaps only boundary vertices and is much faster variation of the KL algorithm. In this paper, we present the boundary Tabu search (BTS) refinement algorithm that combines the Tabu search theory with boundary refinement policy. It has three distinguishing features which are different from ALG2 algorithm. First, we initially insert into the free bucket the gains for only boundary vertices. Second, we remove the above limitation by introducing the conception of *step-status* and *move-direction*. Finally, we derive aspiration criterion from lots of experiments that is simpler than that of ALG2 algorithm.

In [7], Karypis presents the sorted heavy-edge matching (SHEM) algorithm that identifies and collapses together groups of vertices that are highly connected during the *coarsening phase*. Firstly, SHEM sorts the vertices of the graph ascendingly based on the *degree* of the vertices. Next, the vertices are visited in this order and SHEM matches the vertex v with unmatched vertex u such that the weight of the edge $W(v,u)$ is maximum over all incident edges. In [10], Amine and Karypis introduce the concept of the graph *core* for coarsening *power-law* graphs and address the issue whether the information provided by the graph cores can also be used to improve the performance of traditional matching-based coarsening schemes. In this paper, we also present the core-sorted heavy-edge matching (CSHEM) algorithm that combines the concept of the graph *core* with the SHEM scheme.

3 An Effective Multilevel Algorithm for Bisecting Graph

In our multilevel algorithm, we adopt the CSHEM algorithm during the *coarsening phase* and the BTS algorithm in the *refinement phase*. The following describes the CSHEM and BTS algorithms based on multilevel paradigm. In [13], Sediman introduced the concept of the graph *core* firstly that the *core* number of a vertex v is the maximum order of a *core* that contains that vertex. In [14], Vladimir gave an $O(m)$ -time algorithm for cores decomposition of networks. In [15], Vladimir also gave an $O(m \cdot \log(n))$ -time algorithm to compute the *core* numbering in the context of sum-of-the-edge-weights whose complexity is not significantly higher than that of existing matching-based schemes. CSHEM sorts the vertices of the graph descendingly based on the *core* number of the vertices by the algorithm in [15]. Next, the vertices are visited in this order and CSHEM matches the vertex v with its unmatched neighboring vertex whose edge-weight is maximum. In case of a tie according to edge-weights, we will prefer the vertex that has the highest *core* number.

BTS uses free and tabu buckets to fast storage and retrieval the gains of free and tabu vertices respectively. At the beginning of a pass, all vertices are free and the internal and external degrees of all vertices are computed and two free buckets are inserted the gains of boundary vertices of two sides respectively that are computed by $ED(v) - ID(v)$. After we move a vertex v , the gains of itself and its neighboring vertices should be changed. First, we must lock the vertex v by deleting its original *gain* from bucket and insert its new *gain* (negative value of original *gain*) into the tabu bucket of the other side. Second, we update the gains of the neighboring vertices of vertex v . If any of these neighboring vertices becomes a boundary vertex due to the move of vertex v , we insert its *gain* into the free bucket of side in which it locates. If any of these neighboring vertices becomes a non-boundary vertex due to the move of vertex v , we delete its original *gain* from bucket that maybe free or tabu. If any of these neighboring vertices is already a boundary free vertex, we only update its *gain* in free bucket. If any of these neighboring vertices is a boundary locked vertex, we must delete its original *gain* from tabu bucket and insert its new *gain* into the free bucket of side in which it locates. In the terminology of Tabu Search, the tabu restriction forbids moving vertices which are designated as tabu status and the prohibition period of tabu vertex can be changed dynamically and decided by the above promotion rule. The purpose of promotion rule is to increase their chances of following neighbors to the other side of the partition and to allow the chance for the movement of a cluster from one side of the partition to the other.

BTS introduces the conception of *move-direction* and *step-status* that consists of both *forward-move* and *restore-balance*. Given an input partition Q and balance tolerance t , if the input partition Q satisfies the balance tolerance t , current *step-status* is *forward-move* and we initially choose to move a vertex from the side whose bucket contains the highest *gain* vertex among all boundary vertices. Otherwise current *step-status* is *restore-balance* and we choose to move a vertex from the larger side of the partition. As BTS enters into the next step, if new partition satisfies the balance tolerance t , current *step-status* is *forward-move*

and we choose the last *move-direction* to apply the current step, else current *step-status* is *restore-balance* and the *move-direction* of current step starts from the larger side of the partition. The strategy of BTS eliminates the limitation of *move-direction* of steps and its goal is to increase the chances of vertices that are closely connected migrating together from one side of the partition to the other by allowing moving sequences of vertices at one pass from one side of the partition to the other.

```

BTS (initial partition Q, balance tolerance t, Total Moves k){
1  BEST=P;
2  COUNTER=0;
3  compute the internal and external degrees of all vertices;
4  compute the gains of boundary vertices of two sides;
5  insert the gains of boundary vertices in free bucket respectively;
6  while COUNTER <= k do {
7      decide the step-status and move-direction of the current step;
8      select the vertex to move by choice rule and aspiration criterion;
9      move the vertex and lock it;
10     original cut minus its original gain as the cut of new partition;
11     update the internal and external degrees of its neighboring vertices;
12     update the gains of its neighboring vertices by promotion rule;
13     if (the cut is minimum and satisfies balance constraints) then
14         BEST=P;
15     end if
16     COUNTER = COUNTER +1;
17 }end while
18 }

```

Fig. 1. The pseudocode of the BTS algorithm

The remains problem in BTS is how to select a vertex to move in current step. The vertex to move must be selected from free bucket or tabu bucket of the side that is start point of the *move-direction* of current step. When the current *step-status* is *forward-move*, the next vertex to move is the highest *gain* vertex in the free bucket if it is not empty. Otherwise, the highest *gain* vertex in tabu bucket is chosen to move. If the current *step-status* is *restore-balance*, the next vertex to move is the highest *gain* vertex in both free and tabu buckets. It is not possible to run out of moves as long as tolerance t satisfies $0 < t < 1$. The choice rule of Tabu search is to select the highest *gain* vertex in free bucket and the aspiration criterion we have selected to override the tabu restriction is simple criterion that allows tabu vertex as candidate of vertex to move in the current step if current *step-status* is *restore-balance* or the free bucket is empty.

The pseudocode of the BTS algorithm is given in Fig. 1. The while loop (lines 6-17) of BTS is iterated as long as improvements can be made and it is necessary in BTS that setting an upper limit on the parameter k . Because Tabu search aggressively selects the best admissible vertex based on the tabu restriction and aspiration criterion, it must examine and compare a number of

boundary vertices by the bucket that allows to storage, retrieval and update the gains of vertices very quickly. It is important to obtain the efficiency of BTS by using the bucket with the last-in first-out (LIFO) scheme, as Tabu search memory structure, can enforce the “locality” in the choice of vertices to move. The internal and external degrees of all vertices, as complementary Tabu search memory structures, help BTS to facilitate computation of vertex *gain* and judgement of boundary vertex.

4 Experimental Results

We use the 18 graphs in our experiments that are converted from the hypergraphs of the ISPD98 benchmark suite [11] and rang from 13,000 to 210,000 vertices. Each benchmark comes with 3 files, a .net file, a .are file and a .netD file. Each hyperedge is a subset of two or more vertices in hypergraph and is stored in .net file. We convert hyperedges into edges by the rule that every subset of two vertices in hyperedge can be seamed as edge. We create the edge with unit weight if the edge that connects two vertices didn't exist, else add unit weight to the weight of the edge. Next, we get the weights of vertices from .are file. Finally, we store 18 edge-weighted and vertex-weighted graphs in graph format of **MeTiS** [7].

We implement our algorithm in ANSI C and integrate it with the leading edge partitioner **MeTiS**. In the evaluation of our algorithm, we must make sure that the results produced by our algorithm can be easily compared against those produced by **MeTiS**. We use the same balance constraint r and random seed in every comparison. In the scheme choices of three phases offered by **MeTiS**, we use the SHEM algorithm during the *coarsening phase*, the greedy graph growing partition algorithm during the *initial partitioning phase* that consistently finds smaller edge-cuts than other algorithms, the BKL algorithm during the *uncoarsening and refinement phase* because BKL can produce smaller edge-cuts when coupled with SHEM algorithm. These measures are sufficient to guarantee that our experimental evaluations are not biased in any way.

The quality of partitions produced by our algorithm and those produced by **MeTiS** are evaluated by looking at two different quality measures, which are the minimum *cut* (Mincut) and the average *cut* (AveCut). To ensure the statistical significance of our experimental result, two measures are obtained in twenty runs whose random seed is different with each other. For all experiments, we use a 49-51 *bipartitioning* balance constraint by setting r to 0.02. Furthermore, we set the number of vertices of the current level graph as the value of parameter k and 5% as the value of parameter t .

Table 1 presents *min-cut bipartitioning* results allowing up to 2% deviation from exact bisection and illustrates the Mincut and AveCut comparisons of two algorithms on 18 graphs. As expected, our algorithm reduces the AveCut by 1.4% to 52.1% and reaches 32.2% average AveCut improvement. Although our algorithm produces partition whose Mincut is up to 1.1% worse than that of

Table 1. Min-cut bipartitioning results with up to 2% deviation from exact bisection

benchmark	vertices	hyperedges	edges	Metis		CSHEM+BTS		percent	
				Mincut	AveCut	Mincut	AveCut	Mincut	AveCut
ibm01	12752	14111	109183	517	1091	259	685	0.501	0.628
ibm02	19601	19584	343409	4268	11076	3810	7300	0.893	0.659
ibm03	23136	27401	206069	10190	12353	6384	8300	0.626	0.672
ibm04	27507	31970	220423	2273	5716	2205	3044	0.970	0.533
ibm05	29347	28446	349676	12093	15058	8058	10223	0.666	0.679
ibm06	32498	34826	321308	7408	13586	4866	9224	0.657	0.679
ibm07	45926	48117	373328	3219	4140	2483	4081	0.771	0.986
ibm08	51309	50513	732550	11980	38180	12115	18293	1.011	0.479
ibm09	53395	60902	478777	2888	4772	2921	3615	1.011	0.758
ibm10	69429	75196	707969	10066	17747	5850	9083	0.581	0.512
ibm11	70558	81454	508442	2452	5095	2402	3637	0.980	0.714
ibm12	71076	77240	748371	12911	27691	10952	15476	0.848	0.559
ibm13	84199	99666	744500	6395	13469	4769	8068	0.746	0.599
ibm14	147605	152772	1125147	8142	12903	8229	10172	1.011	0.788
ibm15	161570	186608	1751474	22525	46187	14502	28806	0.644	0.624
ibm16	183484	190048	1923995	11534	22156	9302	14536	0.806	0.656
ibm17	185495	189581	2235716	16146	26202	15110	20089	0.936	0.767
ibm18	210613	201920	2221860	15470	20018	15338	18332	0.991	0.916
average								0.814	0.678

MeTiS on three benchmarks, we still obtain 18.6% average Mincut improvement and between -1.1% and 49.9% improvement in Mincut. All evaluations that twenty runs of two algorithms on 18 graphs are run on an 1800MHz AMD Athlon2200 with 512M memory and can be done in half an hour.

5 Conclusions

In this paper, we have presented an effective multilevel algorithm. The success of our algorithm relies on exploiting both Tabu search theory and the concept of the graph core. We obtain excellent *bipartitioning* results compared with those produced by **MeTiS**. Although it has the ability to find cuts that are lower than the result of **MeTiS** in a reasonable time, there are several ways in which this algorithm can be improved. We raise three questions about possible improvement below. The first question is how to find an optimal value for balance tolerance t . The second question, how to find an optimal value for k , is similar with Saab's question about r [9] because we observe that larger values of k lead to better partitions at the expense of a proportional increase in running time and the improvement in the quality of partitions is not linearly related to k . In the Mincut evaluation of benchmark ibm08, ibm09 and ibm14, our algorithm is 1.1% worse than **MeTiS**. Therefore, the third question is how to guarantee find good approximate solutions by setting appropriate value for k and t .

Acknowledgments

This work was supported by the Science Foundation of Shanghai Municipal Commission of Science and Technology, grant No. 00JC14052, and by “SEC E-Institute: Shanghai High Institutions Grid” project. Meanwhile, the authors would like to thank professor Karypis of university of Minnesota for supplying source code of **MeTiS**. The authors also like to thank Alpert of IBM Austin research laboratory for supplying the ISPD98 benchmark suite.

References

1. Zha, H., He, X., Ding, C., Simon, H., Gu, M.: Bipartite graph partitioning and data clustering. Proc. ACM Conf Information and Knowledge Management(2001) 25-32
2. Ding, C., Xiaofeng, H., Hongyuan, Z., Ming, G., Simon, H.: A Min-Max Cut Algorithm for Graph Partitioning and Data Clustering. Proc. IEEE Conf Data Mining (2001) 107–114
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. WH Freeman, New York (1979)
4. Kernighan, B.W., Lin, S.: An Efficient Heuristic Procedure for Partitioning Graphs. Bell System Technical Journal, Vol. 49 (1970) 291–307
5. Fiduccia, C., Mattheyses, R.: A Linear-Time Heuristics for Improving Network Partitions. Proc. 19th Design Automation Conf(1982) 175–181
6. Alpert, C.J., Kahng, A.B.: Recent Directions in Netlist Partitioning. Integration, the VLSI Journal, Vol. 19 (1995) 1–81
7. Karypis, G., Kumar, V.: MeTiS 4.0: Unstructured Graphs partitioning and sparse matrix ordering system. Technical Report, Department of Computer Science, University of Minnesota (1998) Available on the WWW at URL <http://www.cs.umn.edu/~metis>
8. Selvakumaran, N., Karypis, G.: Multi-Objective Hypergraph Partitioning Algorithms for Cut and Maximum Subdomain Degree Minimization. IEEE Trans. Computer Aided Design, Vol. 25 (2006) 504–517
9. Saab, Y.G.: An Effective Multilevel Algorithm for Bisecting Graphs and Hypergraphs. IEEE Trans. Computers, Vol. 53 (2004) 641–653
10. Amine, A.B., Karypis, G.: Multilevel Algorithms for Partitioning Power-Law Graphs. Technical Report, Department of Computer Science, University of Minnesota (2005) Available on the WWW at URL <http://www.cs.umn.edu/~metis>
11. Alpert, C.J.: The ISPD98 Circuit benchmark suite. Proc. Intel Symposium of Physical Design(1998) 80–85
12. Glover, F., Manuel, L.: Tabu search: Modern heuristic Techniques for Combinatorial Problems. Blackwell Scientific Publications, Oxford (1993) 70-150
13. Seidman, S.B.: Network structure and minimum degree. Social Networks (1983) 269–287
14. Batagelj, V., Zaveršnik, M.: An $O(m)$ Algorithm for cores decomposition of networks. Journal of the ACM(2001) 799-809
15. Batagelj, V., Zaveršnik, M.: Generalized cores. Journal of the ACM(2002) 1-8

A New Polynomial Time Algorithm for Bayesian Network Structure Learning

Sanghack Lee¹, Jihoon Yang^{2,*,**}, and Sungyong Park²

¹ Diquest Inc.

Sindo B/D, 1604-22, Seocho-dong, Seocho-gu
Seoul 137-070, Korea
shlee@diquest.com

² Department of Computer Science, Sogang University
1 Shinsoo-dong, Mapo-gu, Seoul 121-742, Korea
{yangjh, parksy}@sogang.ac.kr

Abstract. We propose a new algorithm called *SCD* for learning the structure of a Bayesian network. The algorithm is a kind of constraint-based algorithm. By taking advantage of variable ordering, it only requires polynomial time conditional independence tests and learns the exact structure theoretically. A variant which adopts the Bayesian Dirichlet scoring function is also presented for practical purposes. The performance of the algorithms are analyzed in several aspects and compared with other existing algorithms. In addition, we define a new evaluation metric named *EP power* which measures the proportion of errors caused by previously made mistakes in the learning sequence, and use the metric for verifying the robustness of the proposed algorithms.

1 Introduction

Pattern classification is a crucial and growing field with applications in many domains. Generally, classification is the task for discovering an unknown class of an instance with observed features of the instance which may or may not be closely related to each other in a given domain. Discovering the relationships among the features and making use of them in classification can shed light on the understanding of the domain. A Bayesian network has the capability of finding the relationships among the features. It decomposes variables with independence relations. Therefore, it has become necessary to construct Bayesian networks from the domain of interest in order to understand the relationships among the features for pattern classification.

* This work was supported by grant No. R01-2004-000-10689-0 from the Basic Research Program of the Korea Science & Engineering Foundation and by the Brain Korea 21 Project in 2006.

** Corresponding author.

2 Bayesian Networks

A Bayesian network [1] is a probabilistic graphical model that encodes variables and their dependence relationships into nodes and edges, respectively. It is a formalism for representing and reasoning with models of problems involving uncertainty. For this reason, Bayesian networks are used in many domains where chasing causes and inferring effects are important (e.g. medical diagnosis [2]).

Bayesian networks can be acquired from data or domain experts' knowledge. For complex domains, a number of learning algorithms that generate Bayesian networks given data were developed. The algorithms are divided into two approaches - score-based [3] and constraint-based [4,5,6]. Simply speaking, the former is practical and the latter is theoretical in their use.

In this paper, we use the same symbols those appeared in [7] for consistency. Additionally, $X \prec Y$ means that a variable X precedes Y in a variable ordering \mathcal{O} . $S_{\prec X}$ is a structure of variables preceding X . \mathcal{A} and \mathcal{V} are arcs and vertices, respectively.

3 Sequential Causal Discovery

In this section, a new Bayesian network structure learning algorithm called *SCD* (Sequential Causal Discovery) is described with two theorems.

3.1 SCD Algorithm

The algorithm sequentially determines parents of a node. The sequence is the variable ordering (from the top to the bottom nodes of a given Bayesian network). Let's assume that we have n nodes. The first node trivially has no parents. The second node may or may not hold the first node as a parent. In the same manner, the last node determines its parent nodes from all preceding nodes. The algorithm decides the existence of the edge between two nodes with single conditional independence (CI) test. The test is executed by conditioning Markov blanket of the preceding node between two nodes. The Markov blanket is derived from the partially constructed Bayesian network on the learning process and may not be the same one from the original Bayesian network.

Now we review two important concepts - *d-separation* and a *Markov blanket* [8] - that will be used in the description and the proof for the correctness of the algorithm. D-separation [8,9] is a criterion for deciding the independency between two sets of variables given a conditioning set in a causal graph. A Markov blanket $MB(X)$ of a variable $X \in \mathcal{V}$ is any subset \mathbf{Z} of variables for which satisfies $I(X, \mathbf{Z}, \mathcal{V} - \mathbf{Z} - X)^1$ and $X \notin \mathbf{Z}$. Generally, a Markov blanket of a

¹ The independency between A and C given B is expressed as $I(A, B, C)$, and conditional mutual information $I(A; C|B)$ is another expression for the independency. $I(A; C|B) > 0$ describes the dependency in this paper.

In the experiments, mutual information is acquired from $I(X; Y|\mathbf{Z}) = \hat{I}(X; Y|\mathbf{Z}) + \frac{r_{XZ}^* + r_{YZ}^* - r_{XZ}^* - r_{XYZ}^*}{2N} + z_N \sigma$. For detailed explanations, see [10]. Like other independence tests, we apply a parameter $\alpha = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{z_N}{\sqrt{2}} \right) \right]$. $\alpha = 0.5$ matches $z_N = 0$.

variable X in directed graphs are defined by the following equation, $MB(X) = (\Pi_X \cup \mathbf{Y} \cup \Pi_{\mathbf{Y}}) \setminus \{X\}$ where $\mathbf{Y} = \{Y | X \in \Pi_Y\}$. Based on this background, we have the following theorem.

Theorem 1. *Given two variables X and Y where $X \prec Y$, $I(X; Y | MB(X)_{S_{\prec Y}})_S$ holds if and only if $X \notin \Pi_Y$.*

Proof. As known, $\langle X | \mathbf{Z} | Y \rangle_S \Leftrightarrow I(X; Y | \mathbf{Z})$. So we can prove conditional independency with d-separation $\langle X | MB(X)_{S_{\prec Y}} | Y \rangle_S$. Obviously, if $X \in \Pi_Y$ holds, two nodes are dependent.

When all the paths between X and Y are not active paths, d-separation is satisfied. All unidirectional paths can be divided into the following four cases². We assume that $\mathcal{A}_{XY} \notin \mathcal{A}$ because directly connected variables are trivially d-connected and can not be d-separated by conditioning on any subset.

1. $X \rightarrow Z$ where $Z \in \mathcal{V}_{\prec Y}$: We have to consider the paths in two ways - $X \rightarrow Z \rightarrow$ or $X \rightarrow Z \leftarrow$ - from the viewpoint of d-separation. If the path continues as $Z \rightarrow P$ then the variables are d-separated by Z as an intermediate cause. In another situation $Z \leftarrow Q$, Q is a parent of the child Z and is in the Markov blanket. So, Q becomes intermediate or common cause of Z and another neighbor node of Q . And Q d-separates X and Y , and it can not be Y since $Q \prec Z \prec Y$.
2. $X \rightarrow Z$ where $Z \notin \mathcal{V}_{\prec Y}$: Z will not be an element of $MB(X)_{S_{\prec Y}}$. The path from X to Y through Z should have at least one node as a common effect. But the common effect is not in $\mathcal{V}_{\prec Y}$. Surely, the node is not in conditioning set and the path is not active.
3. $X \leftarrow Z$ where $Z \in \mathcal{V}_{\prec Y}$: Z is one of $MB(X)_{S_{\prec Y}}$ and it blocks the route from X to Y , since Z is a common or intermediate cause of a variable next to Z and X .
4. $X \leftarrow Z$ where $Z \notin \mathcal{V}_{\prec Y}$: Z can not be $Z \notin \mathcal{V}_{\prec Y}$ since $X \prec Y$.

Above cases cover all the paths from X to Y . In all cases, X and Y are d-separated by $MB(X)_{S_{\prec Y}}$ if $X \notin \Pi_Y$. Trivially, X and Y are d-connected if $X \in \Pi_Y$. Thus $I(X; Y | MB(X)_{S_{\prec Y}})$ if and only if $X \notin \Pi_Y$. □

Several constraint-based algorithms find the Markov blanket of the nodes (which d-separates the nodes and others) and orient the directions of the edges. If we have an assumption that a variable ordering is given, then the algorithms find parents of a node (those d-separate the node and its ancestors except the parents) rather than its Markov blanket. The procedure for deciding an exact set of parents of a node is to find a subset of ancestors of the node which d-separates other ancestors from the node, which takes exponential time. To find the parents of a node, our algorithm has a *one-to-one* scheme which is not the same as the *all-to-one* approach. More precisely, *SCD* determines one parent of a variable at a time not a set of parents simultaneously. This difference makes our algorithm time efficient. Algorithm 1 is the *SCD* algorithm, whose correctness is proved in Theorem 2.

² $X \rightarrow Z$ and $X \leftarrow Z$ combined with $Z \prec Y$ and $Y \prec Z$.

Algorithm 1. *SCD*

```

1: Input: Variable ordering  $\mathcal{O}$ , Data  $D$ 
2: Output: Bayesian network structure  $S$ 
3:  $\mathcal{A} \leftarrow \phi$ 
4:  $S \leftarrow (\{\mathcal{O}_1\}, \mathcal{A})$ 
5: for  $i = 2$  to  $n$  do
6:    $Y \leftarrow \mathcal{O}_i$ 
7:   for all  $j$  such that  $1 \leq j < i$  do
8:      $X \leftarrow \mathcal{O}_j$ 
9:     if  $I(X; Y | MB(X)_S) > 0$  then
10:       Add  $\mathcal{A}_{XY}$  to  $\mathcal{A}$ 
11:     end if
12:   end for
13:    $S \leftarrow (\mathcal{V}_{\prec Y}, \mathcal{A})$ 
14: end for

```

Theorem 2. *The SCD algorithm always constructs a correct structure of the Bayesian network.*

Proof. We use the mathematical induction to prove the correctness of the algorithm.

1. $S_{\prec \mathcal{O}_2}^h$ is a true structure. A Bayesian network is a directed acyclic graph. This fact implies that the \mathcal{O}_1 (the only element in $\mathcal{V}_{\prec \mathcal{O}_2}$) itself has no arcs. Thus, an initial structure $S_{\prec \mathcal{O}_2}^h = S_{\prec \mathcal{O}_2}$.
2. Assume that $S_{\prec \mathcal{O}_k}^h$ is a true structure for some k . By the Theorem 1, X and \mathcal{O}_k is d-separated by the Markov blanket of X for all $X \prec \mathcal{O}_k$ if and only if $\mathcal{A}_{X\mathcal{O}_k} \notin \mathcal{A}$. Therefore, $S_{\prec \mathcal{O}_{k+1}}^h$ is also true for some $1 < k \leq n$.

By the steps of mathematical induction, we proved that the algorithm *SCD* always constructs the true structure given the variable ordering. \square

We have introduced two theorems. Theorem 1 showed that the Markov blanket of a node X , which precedes Y , made under subgraph $S_{\prec Y}$ showed precise independence relationships. Theorem 2 showed the correctness of *SCD* algorithm.

3.2 Variant Algorithm

Here, we propose a variant of *SCD* algorithm. The variation is a hybrid algorithm with the *K2 metric*. *SCD* algorithm uses mutual information for determining independency. Like Chi-square and Fisher's z test, CI tests adopt an arbitrary level of decision boundary such as $\alpha = 0.05$. We can set a new decision boundary (usually used term - threshold) for determining independent relation using soft (i.e. relative) value rather than hard (i.e. absolute) value 0. The K2 algorithm [3] stops adding new parents to a variable when the addition can not improve K2 score further. Originally, our constraint-based algorithm's

Algorithm 2. *SCD-K2 boundary*

```

1: Input: Variable ordering  $\mathcal{O}$ , Data  $D$ 
2: Output: Bayesian network structure  $S$ 
3:  $\mathcal{A} \leftarrow \phi$ 
4:  $S \leftarrow (\{\mathcal{O}_1\}, \mathcal{A})$ 
5: for  $i = 2$  to  $n$  do
6:    $Y \leftarrow \mathcal{O}_i$ 
7:    $MI \leftarrow \phi$ 
8:   for all  $j$  such that  $1 \leq j < i$  do
9:      $X \leftarrow \mathcal{O}_j$ 
10:     $MI_j \leftarrow I(X; Y | MB(X)_S)$ 
11:   end for
12:    $\mathbf{I} \leftarrow$  Sort  $MI$  in decreasing order / Return indices
13:    $k \leftarrow \min \{\arg \max_k K2(Y, \mathcal{O}_{1:k})\}$ 
14:   Add  $\mathcal{A}_{\mathcal{O}_{1:k} \setminus Y}$  to  $\mathcal{A}$ 
15:    $S \leftarrow (\mathcal{V}_{\leq Y}, \mathcal{A})$ 
16: end for

```

arc determining mechanism did not take advantage of the conditional independence between the parents and other ancestors of a variable. Adopting K2 will partly make up for the foible in score-based approaches. Algorithm 2 depicts this *SCD* variant.

4 Experimental Results

Experiments have been done with various Bayesian networks. We explain experimental settings and a new metric. The performance of proposed algorithms is compared and analyzed with other algorithms.

4.1 Data

Data for evaluating structure learning algorithms may be acquired from real world domains or sampled from artificial models. Usually, the evaluation for score-based learning algorithms can be achieved with both kinds of data sets. However, the data sets for constraint-based algorithms should satisfy the assumptions previously introduced. By this reason, we evaluate the algorithm with the data sampled from some Bayesian networks.

We've generated data from random Bayesian networks³ varying size and complexity⁴ using BNGenerator [11]. The probabilities for conditional probability tables are generated under between 'deterministic' and 'uniform' distribution.

³ 20 networks have generated with the same size and complexity (induced width) to get accurate results.

⁴ The complexity of the ALARM network in the below looks similar with random Bayesian network with induced width 2. Bayesian networks with induced width as 3 and 5 is complex enough.

Table 1. Results of SCD, Variant, K2, and Hill Climbing

Size of networks		7		10		30		ALARM
Induced width		3	5	3	5	3	5	
<i>SCD</i> ^a	E ^b	1.22	1.17	4.97	6.22	93.25	344.8	61.03
	P	98 ± 4	98 ± 4	97 ± 6	95 ± 7	84 ± 11	66 ± 15	96
	R	92 ± 11	86 ± 11	91 ± 10	86 ± 14	86 ± 7	88 ± 9	96
	F	94 ± 7	91 ± 7	94 ± 7	90 ± 10	85 ± 8	74 ± 9	96
<i>SCD-K2 boundary</i>	E	1.41	1.4	5.49	7.04	103.77	206.53	53.73
	P	99 ± 3	100 ± 0	100 ± 1	99 ± 3	92 ± 8	92 ± 5	98
	R	97 ± 8	96 ± 9	94 ± 8	90 ± 12	86 ± 6	83 ± 5	98
	F	98 ± 5	98 ± 5	97 ± 5	94 ± 8	89 ± 6	87 ± 4	98
K2	E	0.38	0.4	1.69	1.86	30.5	38.49	33.94
	P	98 ± 4	99 ± 3	99 ± 2	99 ± 3	98 ± 3	97 ± 2	82
	R	99 ± 6	97 ± 7	96 ± 7	98 ± 4	98 ± 2	98 ± 3	96
	F	98 ± 4	97 ± 4	97 ± 4	98 ± 2	98 ± 2	97 ± 2	88
Hill-climbing	E	40.47	43.92	230.04	262.02	-	-	-
	P	46 ± 23	57 ± 22	57 ± 14	54 ± 13	-	-	-
	R	51 ± 23	57 ± 20	62 ± 15	60 ± 13	-	-	-
	F	48 ± 23	57 ± 21	59 ± 14	57 ± 12	-	-	-

^a α of *SCD* is 0.48 and of *SCD-K2 boundary* is 0.5.

^b E, P, R, and F stands for elapsed time in seconds, precision, recall, and f-measure.

In addition, algorithms were evaluated with the ALARM network [12] which is a paragon of Bayesian networks and most widely used for structure learning. With models, 10000 samples are generated.

4.2 Experimental Results

Evaluation of algorithms is carried out by the accuracy of determining existence of edges in the Bayesian network. Thus, precision and recall are used that are related to false negatives and false positives. In addition, we introduce a new metric called *EP power* which is appropriate for analyzing the robustness of an algorithm.

Error Propagation Analysis. We analyze the characteristic of an algorithm called error propagation which is useful for measuring the degree of its *fragility*.

Definition 1 (Error Propagation). A false positive which caused by one or more false negatives is called an error propagation, if

1. A false positive between variables X and Z .
2. The set of false negatives between the variable X and the set of variables \mathbf{Y} .
3. $X \notin \Pi_Z$, $Y \in \Pi_Z$, $X - Y$, and $X, \mathbf{Y} \prec Z$

Simply speaking, all $Y \in \mathbf{Y}$ satisfy $X - Y \rightarrow Z$. This relation can be divided into two different relations: Y is an intermediate cause or same cause of X and Z . Thus, missing any one of \mathbf{Y} in the hypothesis (one or more false negatives) causes an extra arc $X \rightarrow Z$ (a false positive).

Table 2. The *EP power* analysis of *SCD* and its variants

	α	$p(FP)$	$p(FP_{XZ})$	$p(EP_{XZ})$	<i>EP power</i>	$p(FP FP \cup FN)$
<i>SCD</i>	0.48	0.0706	0.1001	0.1762	0.0693	0.6918
	0.45	0.0242	0.0390	0.1051	0.1609	0.2854
	0.50	0.0135	0.0243	0.0819	0.0937	0.2922
<i>SCD-K2 boundary</i>	0.48	0.0114	0.0207	0.0761	0.1131	0.2411
	0.45	0.0144	0.0251	0.0771	0.1144	0.2268

Definition 2 (Error Propagation Power). *The increased or decreased portion of errors purely caused by error propagation over total errors.*

$$\frac{(p(EP_{XZ}) - p(FP_{XZ}|\mathcal{A}_{XY} \setminus \mathcal{A}_{XY}^h = \phi)) |XZ|}{|FP| + |FN|} \tag{1}$$

$p(EP_{XZ})$ can be rewritten as $p(FP_{XZ}|\mathcal{A}_{XY} \setminus \mathcal{A}_{XY}^h \neq \phi)$.

In Table 1, the original algorithm have shown the curse of false positives. Even though the probability of the false positive is low (about 5%), large number of false positives causes poor precision. By mixing *SCD* with the K2 metric, it properly filtered many false positives even though the K2 metric was just for determining threshold. In addition with a reasonable Bayesian network model ALARM network, our algorithms perform more accurate than the K2 algorithm (For more results on ALARM network, see [6]), because the results of the CI tests are more reliable and the algorithms do not fall into local maxima.

In *EP power* analysis in Table 2, we have shown that the error cascading effect is smaller (about 10%) than general false positives and negatives. Because the *EP power* is a part of false positives, the quality of CI tests should be improved to reduce such errors.

5 Conclusion and Future Work

Novel approaches to the structure learning of Bayesian network (*SCD* and variant) have been proposed and those were compared with existing algorithms through carefully designed experiments, yielding improved performance. The results of our original algorithm *SCD* and the results from experiments of the variation were variously analyzed in the previous section.

In conclusions, *SCD* algorithm itself is theoretically proved algorithm but is under the control of the quality of CI tests. To lessen the effect, we hybridize the algorithm with the K2 metric and obtained promising results. In this context, we suggest the following as future works:

- **Variable ordering:** In general, a variable ordering for a structure is not given. Hence, we have to find such ordering before running the algorithm. The ordering can be derived from a wrapper approach [13] for score-based algorithms. Transforming such methods for constraint-based algorithms should be achieved.

- **Heuristics and post process:** Constraint-based algorithms look compact and solid. But an imperfect probability distribution from limited finite data makes the algorithms fallacious. For the robust CI tests, more heuristics and rules [14] might be added to the algorithm.

References

1. J. Pearl, “Fusion, propagation and structuring in belief networks,” in *Uncertainty in Artificial Intelligence* (Kanal and Lemmer, eds.), pp. 357–370, North-Holland, 1986.
2. P. Spirtes, C. Glymour, R. Scheines, S. Kauffman, V. Aimale, and F. Wimberly, “Constructing bayesian network models of gene expression networks from microarray data,” in *Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology*, 2001.
3. G. F. Cooper and E. Herskovits, “A bayesian method for the induction of probabilistic networks from data,” *Machine Learning*, vol. 9, pp. 309–347, 1992.
4. P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. Springer-Verlag, 1993.
5. J. Pearl and T. S. Verma, “A theory of inferred causation,” in *KR’91: Principles of Knowledge Representation and Reasoning* (J. F. Allen, R. Fikes, and E. Sandewall, eds.), (San Mateo, California), pp. 441–452, Morgan Kaufmann, 1991.
6. J. Cheng, R. Greiner, J. Kelly, D. A. Bell, and W. Liu, “Learning bayesian networks from data: An information-theory based approach.,” *Artif. Intell.*, vol. 137, no. 1-2, pp. 43–90, 2002.
7. D. Heckerman, “A tutorial on learning bayesian networks,” Tech. Rep. MSR-95-06, Microsoft Research, 1995.
8. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988.
9. J. Pearl, *Causality: Modeling, Reasoning, and Inference*. Cambridge: Cambridge University Press, 2000.
10. M. S. Roulston, “Estimating the errors on measured entropy and mutual information,” *Physica D: Nonlinear Phenomena*, vol. 125, pp. 285–294, 1 1999.
11. J. S. Ide, F. G. Cozman, and F. T. Ramos, “Generating random bayesian networks with constraints on induced width.,” in *ECAI* (R. L. de Mántaras and L. Saitta, eds.), pp. 323–327, IOS Press, 2004.
12. I. Beinlich, H. Suermondt, R. Chavez, and G. Cooper, “The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks,” in *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, (London), pp. 247–256, 1989.
13. W. H. Hsu, “Genetic wrappers for feature selection in decision tree induction and variable ordering in bayesian network structure learning.,” *Inf. Sci.*, vol. 163, no. 1-3, pp. 103–122, 2004.
14. H. Steck, *Constraint-Based Structural Learning in Bayesian Networks using Finite Data Sets*. PhD thesis, Munich University of Technology, 2001.

Personalized Recommendation Based on Partial Similarity of Interests

Ming-Hua Yang and Zhi-Min Gu

School of Computer Science and Technology, Beijing Institute of Technology, Beijing
100081, China
minghuay@bit.edu.cn, zmgu@x263.net

Abstract. The current collaborative recommendation approaches mainly measure users' similarity by comparing user's entire interests and don't consider user's interest quality, especially interest span. We propose a new approach to provide inter-website recommendation on proxy server based on partial similarity of interests, and construct corresponding user's interest model to realize this method. According to psychological characteristic of interests, this approach divides user's interest into several interest-points, which are correlated each other and farther divided into long-term interest and short-term interest. We mine the interest quality and correlation of interest-points from proxy log to construct user's interest model. This method adopts different recommendation mechanism separately for long-term interests and short-term interests, which provides recommendation to target user's long-term interests based on neighbors with partially similar interests and recommendation to user's short-term interests based on experienced users. Experimental results indicate that this method can recommend interesting and unexpected inter-website pages to target users and improve the precision of personalized recommendation service on proxy server.

1 Introduction

Owing to the dramatic growth of the web, users have to face a large number of web pages and often waste a lot of time to search the information they need. Personalized recommendation has been used to solve the problems of information overload. Collaborative recommendation is the most successful technology of personalized recommendation and has been researched widely at present. It recommends the user with items that people with similar tastes and preferences liked in the past. So this approach must measure the similarity of users and cluster users into groups. At present, there have been many collaborative recommendation systems [1][2]. But these systems have many problems, which can be summarized as follows:

First, According to Personality Psychics, human's interests are different in tendency, span, stability and functionality, which are called interest quality [3]. User's interests are relative centralized and have several aspects of interests called interest-points. The existing recommendation systems don't consider user's interest quality, especially interest span. They construct user's neighbors based on user's entire interests by comparing their all history behavior and neglect all aspects of interests. Hence,

the system can't have the expected result even if the user only expects to focus on an aspect of his interests.

Second, each of user's interest-points varies in intensity and has different importance of status. Interests can be divided into long-term interests and short-term interests according to their lasting time [4]. For happening very casually and stochastically, short-term interests always are the noise data in measuring the users' similarity. On the other hand, if the target user's current interest is his short-term interest, the items recommended with neighbors may be not the information he need. Because long-term interests inevitably act as kernel role when comparing users' similarity, the neighbors are also unfamiliar with the content when the target user's current interest is his short-term interest.

Finally, the existing recommendation systems mainly focus on individual servers and the items recommended are usually intra-website. The proxy server log reflects multi-users/multi-sites access patterns, can revert the correlation of websites and provide inter-website recommendation. But for involving multi-sites and having more intricate data, measuring users' similarity for collaborative recommendation on proxy is more difficult.

In order to solve these problems, this paper propose a new approach to realize inter-website recommendation on proxy server based on partial similarity of interests according to psychological characteristic of user's interests, and design a proxy-based personalized recommendation system to realize the method.

2 Mining User's Interest Model

To realize recommendation based on partial similarity of interests, we divide user's interest into several interest-points and mine the relationship of interest-points when constructing user's interest model. Using these models, we can measure users' similarity on a certain interest-point and carry on recommendation based on neighbors with partially similar interests.

The page hierarchy structure of a website is constructed by the website designer according to pages semantic category. It provides good enough hints of classification information. The pages in the same content have high similarity on their semantic content, so user's long time access to a content of a website manifests that the user has strong interest in the content. To fully utilize the classification information of website hierarchy structure, we define an interest-point as the path between the root of a website and one of its leaves. Therefore an interest-point includes all the pages under the path of the website, namely $\text{PageS} = \langle p_1, p_2, \dots, p_k \rangle$.

2.1 Quality of Interest-Points

Interest span reflects how many interest-points the user has, which is the total characteristic of user's interests. Other characteristics of interest quality are reflected by the quality of each interest-point. So quality of each interest-point can be represented by tendency, stability and functionality, as table 1 shows.

Table 1. Representation of quality of interest-point

Quality	Representation	Description
tendency	Degree	Interest degree to interest-point IP
	WeightS	Interest weight vector to each page of PageS
stability	Count(IP)	The number of access times to IP
	RAD	Recent access density on IP
functionality	Experience	User's web online experience on IP
	Correlation	Correlation of pages in IP

The interest degree to interest-point IP is defined as the ratio of the access times to IP to the user's total access times. WeightS is the interest weight vector to each page of IP, $\text{WeightS} = \langle w_{x,1}, w_{x,2}, \dots, w_{x,k} \rangle$, $w_{x,i}$ is the interest degree to the page p_i .

Count(IP) and RAD are used to measure user's interest stability and distinguish user's long-term interests and short-term interests. Count (IP) is the user's access times to interest-point IP, which is defined as the number of all the sessions that contain interest-point IP seen so far from the user's browsing history. RAD (Recent Access Density) is used to filter out the interest-point that appears infrequently in the recent sessions and reveal the interest-point that appears frequently in the recent sessions.

Definition 1 (Recent Access Density, RAD). Given user A and an interest-point IP, the first session user A accessed IP is First(IP), the last session A accessed IP is Last(IP), the current session is S_c , the number of all sessions between First(IP) and S_c is N_s . Count (IP) is the number of sessions that contain IP between First(IP) and S_c . Then user A's recently access density on IP is:

$$\text{RAD}(A, IP) = \frac{\text{Count}(IP)}{N_s}. \quad (1)$$

Here, we use the expired time (denoted by θ) to restrict the interval between the last and the current session. When the interval exceeds θ , both First(IP) and Last(IP) are changed to the current session.

Given thresholds of the minimal RAD (denoted by α) and the minimal Count(IP) (denoted by β), for an interest-point IP of user A, if the Count(IP) and RAD are both greater than their thresholds, the interest-point IP is the user A's long-term interest, otherwise if the RAD is greater than α but the Count(IP) is less than β , the interest-point IP is the user A's short-term interest. If the Count(IP) is greater than β , but the RAD is less than α , this means that user U has been interested in IP once, but not interested in it now.

Moreover, users usually trust the recommendation from like-minded people with high experience, and even expect the recommendation coming from experts in some field. To copy successful user's online experience to many users, we measure user's online experience. Frequency and longevity of use can represent user's online experience, so we use online time and frequency to measure user's online experience. Suppose user A has accessed the interest-point IP $\text{Count}_A(IP)$ times, the maximum of the

access times of all users to IP is $Count_{max}(IP)$, then user A's online experience on the interest-point IP is defined as $E(A,IP) = Count_A(IP) / Count_{max}(IP)$.

2.2 Measuring Correlation of Interest-Points

Definition 2 (Correlation of interest-points). Correlation of interest-points is used to express the relationship between any two interest-points. It also can describe the possibility of transferring mutually between interest-points. Suppose user A has two interest-points I_i, I_j in his interest-points, the correlation of A's interest-point I_i, I_j is:

$$Corr_A(I_i, I_j) = \frac{p(I_i, I_j)}{P(I_i)P(I_j)} \quad (2)$$

Where $p(I_i, I_j)$ is the probability of user A's sessions containing interest-point I_i and I_j at the same time, and defined as the ratio of the user A's sessions that contain interest-point I_i and I_j at the same time to the total user A's sessions seen so far. If $Corr_A(I_i, I_j)$ is less than 1, then the interest-points I_i and I_j are negatively correlated; If $Corr_A(I_i, I_j)$ is greater than 1, then the interest-points I_i and I_j are positively correlated, meaning that each event implies the other; If $Corr_A(I_i, I_j)$ is equal to 1, then I_i and I_j are independent and there is no correlation between them.

The definition gives the correlation of interest-points. These interest-points and their correlation constitute user's interest model. Based on these interest models, we can cluster user into interest groups with similar interests and then carry on personalized recommendation service.

3 Partial Similarity of Users' Interests

The existing recommendation systems construct user's neighbors based on user's entire interests and don't consider interest quality, neglect that users usually have several interest-points. Hence, when measuring the similarity of users, even if two users are very similar in an aspect of interests, the whole similarity is not always great. In order to obtain the similar users, these systems have to reduce the similar threshold to cluster users, which causes the dissimilar users clustered into the same group and then reduces the precision of the system. When comparing users' similarity, the long-term interests inevitably act as kernel role. For happening very casually and stochastically, short-term interests always are the noise data in measuring the users' similarity.

For example, as figure1 shows, user A's interest set is $I_A = \{\text{personalization, music}\}$, and user B's interest set is $I_B = \{\text{personalization, soccer}\}$. According to existing recommending algorithm, if A and B are highly similar on "personalization", and the entire interest similarity also is higher than the system threshold, then "music" will be recommended to B and "soccer" will be recommended to A. It is inappropriate obviously. Moreover, when calculating users' similarity, although A and B are highly similar on "personalization" and can be neighbors mutually, the existence of user A's

“music” and user B’s “soccer” may cause that A, B are not neighbors, then lose the significant content to recommend.

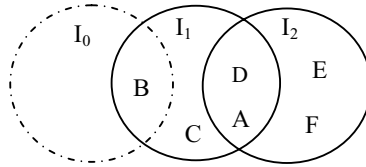


Fig. 1. Clustering users into IG (I₀: Soccer; I₁: Personalization; I₂: Music)

In view of user’s interest span, this paper constructs neighbors by clustering users into groups based on partial similarity of users’ interests and assigns target user to a certain interest-group for one of his interest-point, so a user can belong to many interest-groups. Here interest-group (IG) is defined as the user group with high similarity on certain aspect of interests. Figure 1 demonstrates how to cluster users into IGs. I₀, I₁, I₂ respectively represents “soccer”, “personalization”, “music” IG. The target user A is similar to user BCD on interest-point I₁, and similar to DEF on I₂.

Given an interest-point IP, the similarity of user A and B on IP is defined as:

$$Sim(x, y, IP) = \frac{1}{K} \sum_{k=1}^K (w_{x,i} - \bar{w}_x)(w_{y,i} - \bar{w}_y) \tag{3}$$

Where K is the number of pages in the interest-point IP, $PageS = \langle p_1, p_2, \dots, p_K \rangle$ is page sequence in IP, $w_{x,i}$ is the interest weight of user x on page p_i .

4 Recommendation Mechanism

We adopt different recommendation mechanisms for user’s long-term interests and short-term interests. The recommendation to target user’s long-term interests is based on neighbors with partially similar interests and the recommendation to user’s short-term interests is based on experienced users.

If target user A’s current interest-point I_i is one of his long-term interests, we provide recommendation to him based on neighbors with partially similar interests on interest-point I_i . Suppose n is the members of interest-group I_i , u_k is user A’s neighbor, $Sim(A, u_k, I_i)$ is similarity between user A and user u_k on interest-point I_i , $E(u_k, I_i)$ is the experience of user u_k to interest-point I_i , $Corr_{u_k}(I_i, I_j)$ is the correlation of interest-point I_i and I_j in user u_k ’s interest model, then the recommendation score $R_{A,li}(I_j)$ for user A of interest-point I_j correlated to I_i is defined as follows:

$$R_{A,li}(I_j) = \frac{1}{n} \sum_{k=1}^n sim(A, u_k, I_i) * E(u_k, I_i) * Corr_{u_k}(I_i, I_j) \tag{4}$$

After obtaining the recommendation score $R_{A,li}(I_j)$ according to formula 4, we sort I_j in descending order of recommendation score and select the TOPN interest-points to recommend to target user A.

If target user A's current interest-point I_i is one of his short-term interests, we provide recommendation to him based on the experienced users on interest-point I_i . The function $R_{A,I_i}(P_j)$ is used to calculate the recommendation score of page P_j in I_i based on TOPN experienced users in IG I_i when target user A's current interest-point I_i is his short-term interests, where w_{u_k, P_j} is the interest weight of user u_k to page P_j , and $E(u_k, I_i)$ is the experience score of user u_k on interest-point I_i , N is the variable of TOPN.

$$R_{A,I_i}(P_j) = \frac{1}{N} \sum_{k=1}^N E(u_k, I_i) * w_{u_k, P_j} \tag{5}$$

5 Experiment

5.1 System Architecture

Using the above recommendation method, this paper designs and implements a proxy-based personalized recommendation service system. This system mines user's interest model from proxy log, and clusters users into interest-group according to partial similarity of interests, and then carries on recommendation for target user's long-term interests based on partially similar neighbors and recommendation to target user's short-term interests based on experienced users. The system architecture is as figure 2 showed.

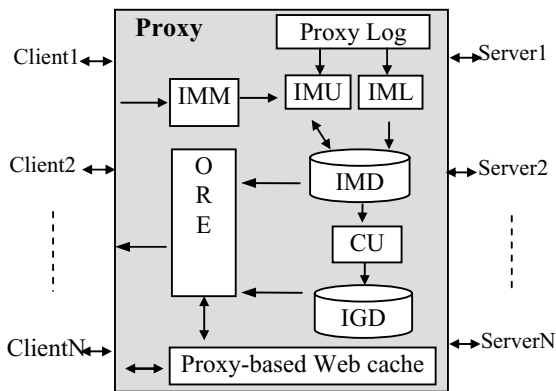


Fig. 2. System architecture of proxy-based personalized recommendation service system

The system mainly includes five parts: IML (Interest Model Learner), IMM (Interest Model Manager), IMU (Interest Model Updater), CU (Clustering User), ORE (Online Recommendation Engine). IML analyzes users' interest characteristic from proxy log, establishes users' interest models, and stores them in IMD (Interest Model Database). IMM provides a self-controlling platform for user to maintain his individual interests on web; IMU updates user's interest model according to user's recent

behavior or the result returned from IMM, and retains the result updated in IMD. CU is to cluster user into IGs according to partial similarity of interests and store them into IGD (Interest-Group Database). ORE recognizes current user's online interest and shows a list of recommendation in descending order of recommendation score.

5.2 Experimental Results

The experimental data we used is the real-world proxy log of a domestic university about one month. We extract ten users' access records from log to establish interest models and provide recommendation to them. The performance metrics are defined as follows: recommendation *accuracy* is the ratio of the number of recommended items that the user is interested in to the total number of recommended items. Here, we also analyse user's behavior to judge if the user is interested in the recommended items. If the user clicks the recommended item and visits it exceeding the threshold time (2 minutes) or open a link from it, we regard that the user is interested in this item. *Unexpected ratio* is the fraction of the number of recommended items that the user is interested in but not in his interest model to the total number of recommended items.

The ten users extracted have participated in our experiment and responded to the recommendation. We show the average score of our method in table 2, compared with other methods.

Table 2. Experimental Results

Method	Accuracy	Unexpected ratio
Existing Method	77%	12%
Method of literature8	52%	30%
Our Method	80%	78%

From the experimental result, we can see that our method can recommend unexpected items compared with existing method. It's because recommendation from proxy server can recommend inter-website pages from multiple sites. Users maybe not know all of these websites related to his interests, and once they know the websites from recommendation, they can be relatively easy to find the intra-website information. For a user, recommending an authoritative and related website from his experienced neighbors may be more meaningful.

6 Conclusion

To provide collaborative recommendation service, the current approaches mainly construct user's neighbors based on user's entire interests by comparing their history behavior and don't consider user's interest quality, especially interest span. According to psychological characteristic of user's interests, we propose a new approach to realize inter-website recommendation on proxy server based on partial similarity of interests, and design a proxy-based personalized recommendation system to realize the method. This approach divides user's interest into several interest-points, which are farther divided into long-term and short-term interest. It mines user's interest quality

and the correlation of interest-points from proxy log to construct user's interest model and adopts different recommendation mechanism separately for long-term interests and short-term interests. It carries on recommendation to target user's long-term interests based on neighbors with partially similar interests, and recommendation to user's short-term interests based on experienced users. Experimental results indicate that this method can recommend interesting and unexpected inter-website pages to target users and improve the precision of personalized recommendation service on proxy server.

References

1. Zeng Chun, Xing Chun-xiao and Zhou Li-zhu: A Survey of Personalization Technology. *Journal of Software*, Vol.13, No.10. (2002)1952-1961
2. Gediminas Adomavicius, Alexander Tuzhilin: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE transactions on Knowledge and Data Engineering*, Vol.17, No.6. (2005)734-749
3. Ye yi-qian and Kong ke-qin: *Personality Psychics*. East China Normal University Press. Shanghai (1993)
4. Guo Yan, Bai Shuo, Yang Zhi-feng and Zhang Kai: Analyzing Scale of Web Logs and Mining Users' Interests. *Chinese Journal of Computers*, Vol.28, No.9. (2005)1483-1496
5. Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom and John Riedl: GroupLens: An open architecture for collaborative filtering of netnews. *Proceeding of the ACM 1994 Conference on Computer Supported Cooperative Work*, New York (1994)175-186
6. John S. Breese, David Heckerman and Carl Kadie: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceeding of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI1998)*, Morgan Kaufman (1998) 43-52
7. Yi-Hung Wu, Yong-Chuan Chen and Arbee L.P.Chen: Enabling personalized recommendation on the Web based on user interests and behaviors. In the 11th Int. Workshop on Research Issues in Data Engineering (RIDE-DM 2001), Heidelberg, Germany (2001)17-24
8. Junzo Kamahara, Tomofumi Asakawa, Shinji Shimojo, Hideo Miyahara: A Community-based Recommendation System to Reveal Unexpected Interests. *Proceeding of the 11th International Multimedia Modeling Conference (MMM2005)*, Melbourne, Australia (2005) 433-438
9. Wenwu Lou, Hongjun Lu: Efficient Proxy-based Web Access Prediction Service. *Proceeding of the 11th International Conference on Information and Knowledge Management (CIKM2002)*, McLean, Virginia (2002)169-176

A Fast Implementation of the EM Algorithm for Mixture of Multinomials

Jan Peter Patist

Free University Amsterdam,
Amsterdam,
The Netherlands,
jpp@few.vu.nl

Abstract. We propose several simple techniques which dramatically reduce both the memory demand and computational effort in building multinomial mixture models using the EM algorithm. The reason of the dramatic improvement in performance is that the techniques make use of certain properties of the data. These properties are: the data is sparse and there are many repeating records. We claim that particular sources of data consistently satisfy these properties. Excellent examples are Clickstream and retail data which are very sparse and consist of many repetitions. Using simple techniques huge speed-ups and compression rates, on real life clickstream data sets, are observed compared to the standard implementation of the EM.

1 Introduction

Every day a huge amount of data is generated in the form of click streams, telephone records, multimedia data, sets of retail chain transactions, et cetera. With the increase of the amount of data the need grows for fast algorithms to analyze these huge datasets.

In the past it has been claimed [1] that an iterative procedure like the EM-algorithm[2] is an unsuitable method for large data sets. In the meantime there has been a lot of work performed on improving the run-time of the EM algorithm [3,4,5,6,7].

In this paper we propose some simple techniques for efficiently implementing the EM for the estimation of a multinomial mixture model. The effectiveness of these techniques is based on certain assumptions about the data. Naturally, because of the nature of multinomial models we assume the data is discrete. More importantly, we assume that the data is sparse, (i.e. that the data contain lots of zero-values) and the data contain repeating records.

A lot of research has been performed on speeding up the EM-algorithm. Our techniques differ from others in that they are not generally applicable. These techniques are only suitable for sparse, redundant and discrete data. Our techniques are not necessarily in competition with other techniques; such as in [3,4,5,6,7]. Some of these techniques can be combined with our techniques. The main goal of this paper is showing the improvement that can be obtained

solely by the application of our techniques, by comparing it with the standard implementation of the EM-algorithm.

In practice we observed that datasets from particular sources often satisfy prerequisites. This makes our techniques important for the analysis, using multinomial mixtures, of domain specific data. For example, data containing page visits within sessions from web-log-files are often highly sparse and redundant. Therefore, in this paper the effectiveness of our techniques is tested on datasets originating from page visits.

An example data source is an on-line webshop logging the page visits of customers. This data is then stored in a table of sessions of counts of page views. A common technique in analyzing customer behavior is finding clusters using mixture of multinomial models [8].

Is it not surprising that clickstream data is sparse and contain lots of repetitive records. This is because visitors are usually focussed only on a few pages.

In this paper we propose three techniques, which make use of repetitions of records and sparsity. These techniques are: We transform the data into a weighted dataset. In the weighted data set duplicate records are replaced with the unique record and its frequency. We introduce a sparse representation, and propose a lookup table to store partial results.

The paper is structured in the following way: We give a list of related work with respect to the improvement of the EM-algorithm and application of multinomial mixture models to webdata, then we introduce the multinomial mixture model and mixture estimation using EM. Hereafter we specify the new representation of the data. We show that the proposed techniques maintain equivalence to the standard EM. The effectiveness is shown by comparing the run-time and memory load of the standard implementation of the EM with the “enhanced” EM on several real world data sets.

2 Related Work

In reporting on related work we restrict our attention to techniques of speeding up the EM-algorithm and the application of multinomial mixtures in the context of webdata. Other techniques such as listed in the Related Work section are not necessarily in competition with our techniques, and consequently also important with respect to potential further improvement of the performance of the EM-algorithm.

Several techniques have been proposed to speed-up the EM algorithm for large datasets. In [3] a fast EM-algorithm for Gaussian mixture models is proposed based on kd-trees. In experiments big speed-ups were observed, which increased up to linear by the amount of data points.

For incremental and sparse views [7] of the EM-algorithm theoretical justification is given which can speed-up convergence. The incremental EM algorithm replaces an EM-step in the basic EM-algorithm by k EM-steps on k blocks of approximately the same size. Sparse EM only recalculates class probabilities of non-negligible probabilities and freezes the negligible class probabilities.

In [9] optimal block sizes are determined for the incremental EM and another sparse EM view is suggested which determines negligible data points based on the change of the class probabilities observed in the previous EM-step.

A greedy EM procedure is introduced in [6]. The greedy method starts with the allocation of one component and iteratively adds a component while fixing the other components. It is shown that greedy EM is less sensitive to initial configuration and fast (in combination with kd-trees) without significant difference in accuracy. In their paper this technique was applied to a mixture of multivariate Gaussian distributions. In the paper of Blekas [10], the method was adjusted for the mixture of multinomial distributions.

In [5] Hulten and Domingos propose a general method for scaling learning algorithms. The methods posts an upper bound, as a function of the number examples, of the loss between a model based on n examples instead of building it an infinite number of data points. The method iteratively adds data points until the probability that the model will improve is small. In the paper the technique is applied to Gaussian mixture models.

In [4] the authors propose a one-scan EM-algorithm for large datasets. The effectiveness of this procedure is based on the fact that regions can be identified which can be compressed by replacing the data points with sufficient statistics. The procedure starts with basic EM on a sample. Then all records in the data set not used in the previous step are then discarded, retained or compressed. Discarded are those records which are “almost certain” with respect to its class probability, retained are highly uncertain records. The data is compressed when the data maintained in the buffer exceeds a buffer size. Data points are compressed into statistics or discarded based on their “clusterness”. The method seems scalable with respect to the number of records compared to basic EM. The scalable EM was shown to be more accurate than sampling. Because the EM is sensitive to initialization, the procedure is extended to make multiple model learning memory more efficient.

Multinomial mixtures proved usefull in [11] in the context of clickstream data . In this paper multinomial mixture models are used to estimate user profiles in the domain of retail data. It is shown that multinomial mixture models outperform simple histograms. For an overview on clustering techniques in webdata, see [12].

3 EM for Mixture of Multinomials

A d -dimensional multinomial mixture is a probability distribution on \mathcal{N}^d , the discrete domain, that is given by a convex combination of k multinomial mass functions

$$p(x) = \sum_{s=1}^k \alpha_s p(x|s), \quad \text{and} \quad p(x|s) = \binom{N}{\prod_{j=1}^d x_j!} \prod_{j=1}^d \pi_j^{x_j}$$

where $N = \sum_{j=1}^d x_j$ and $\sum_{j=1}^d \pi_j = 1$.

The components of the mixture are indexed by a random variable s that takes values from 1 to k , and $\alpha_s = p(s)$ defines a discrete prior probability distribution

over the components. Given a set $\{x_1, \dots, x_n\}$ of independent and identically distributed samples from $p(x)$, the learning task is to estimate the parameter vector $\theta = \{\alpha_s, \pi_s\}_{s=1}^k$ of the k components that maximizes the log likelihood. A common algorithm to estimate the parameters is the Expectation Maximalization algorithm (EM) [2]. The algorithm starts with initial values of the parameters and iteratively performs two steps, called the E-step and M-step, trying to improve the value of $L(\theta)$ by adjusting θ . In the E-step the expected value of the latent components are calculated. In the M-step new parameter values are calculated by setting it equal to its expectation. The procedure terminates after a pre-specified number of iterations, or when the improvement rate drops below a certain threshold.

The update equations for α and π are:

$$\gamma_{i,s} = \frac{\alpha_s p(x_i|\theta_s)}{\sum_s \alpha_s p(x_i|\theta_s)}, \alpha_s = \frac{\sum_{i=1}^n \gamma_{i,s}}{n}, \pi_s = \frac{\sum_{i=1}^n \gamma_{i,s} x_i}{\sum_{i=1}^n \sum_{s=1}^k \gamma_{i,s} x_i}$$

Because the log of the Multinomial coefficient, as defined in equation 1, only adds a constant factor to the log Likelihood it is not involved in the estimation of θ .

4 Data Representation

The data is assumed to be discrete. The standard EM-algorithm for multinomial mixtures uses the standard representation of the data, namely a table of n records. Where each record consist of values of d variables.

As mentioned earlier the success of our method depends on the assumption of sparsity and redundancy. The more redundant and sparse the data is, the more effective our techniques are.

Table 1. Two different data representations: the left table is the standard representation and the right the weighted sparse representation

Record	X ₁	X ₂	X ₃	X ₄
1	1	0	2	0
2	1	0	2	0
3	1	1	0	0
4	1	0	0	1

$$D = [(1,1),(-3,1),(1,1),(-2,1),(1,1),(-4,1)]$$

$$W = [2,1,1]$$

$$U = [\{1\},\{1\},\{2\}]$$

We represent the data by the triplet (W, D, U) . W represents the weight of the record (i.e. the number of occurrences of the record in the dense view of the data). U represents the unique values per attribute, thus U is a list of d lists of unique values (note that zero values are not stored).

The attribute-value pairs are stored within D . Every non-zero attribute-value combination within the list is represented by a tuple (pos, val) , where pos is an attribute index, and val the index to the value of the attribute of that particular record. As an example the tuple $(pos = 1, val = 1)$ refers to the first element of the first element in U . The vector D is ordered in the sense that the tuples

(*pos, val*) from the same record succeed each other. The minus sign in the *pos* vector is used to indicate the last element of the record. See Table 1 for an example of our representation of the data.

The amount of memory needed to store the sparse data representation is 2 times the number of non-zero values plus the sum of the number of unique values per variable. Note that because we assume that the number of unique values are small, U and D can be stored using a small bit integer.

The cost of the transformation of the standard representation of the data to the new representation is negligible. Transforming the data set into a weighted set without duplicates can be done efficiently by hashing and sorting, thus $O(n \log n)$, where n is the number of records. Also the transformation has to be performed only once, whereas building a mixture model involves several iterations, several re-runs because of sensitivity to initialization and with different number of components.

5 Efficient Implementation of the EM-Algorithm

In this section we present our efficient implementation of the EM-algorithm. The efficiency is obtained by representing repetitions of records as a single record with its accompanying weight, by not calculating multiplications and exponentials including zero-value and by creating a lookup table to store exponentials.

The proposed EM-algorithm is in fact identical to the standard EM-algorithm as we will show.

In this section we will show that by the transformation of the data set, the EM can be adjusted in such a way that the algorithm is still identical to the standard EM-algorithm. From now on we will call the EM-algorithm using the (W, D, U) data representation the “enhanced EM” algorithm.

First we will show that weighting will not change the result of the standard EM-algorithm. Assume the set X of all identical records which equal x_i with a size of m records. By the definition of the posterior probability, this probability is equal for all records for all components in X . As a result, the contribution of X to α is m times the contribution of the single x_i . Using the same reasoning it can be shown that the estimated π is also identical to that of the standard implementation of the EM algorithm.

In the sparse representation we removed all zero-values. Implicitly they are known because the sparse representation is reversible, however in applying enhanced EM, the zero-values are not used. When an attribute value is zero the contribution of this value to calculating class probabilities is a multiplication by one, because $\pi^0 = 1$. And in the M-step the contribution is zero times γ . Therefore the zero-values don’t have to be calculated at all.

Using the three techniques of eliminating duplicate records, only representing non-zero values and the use of a reference table we obtain three different “types” of computational speed-up. Eliminating duplicates saves us the most computational effort. For these records we have to calculate γ_i and the contribution to θ only once, using only one extra multiplication. For zero-values we don’t have to

calculate exponentials, however for each unique record we still have to calculate γ_i and its contribution to θ . We used a lookup table which allows us to calculate the needed exponentials (π^{x_j}) only once.

The amount of memory needed for the lookup table is kdu , where u is the average amount of the unique values per variable. In the experiments we observed that the size of the lookup table is small and feasible.

Table 2. Statistics of several data sets. The size is expressed in the total amount of numbers. “Compression rate replicates” is calculated by dividing the number of records in standard representation of the data set by the number of unique records. “The mean unique values” is the average number unique values per variable or dimension. The “Total Compression” is the compression obtained by combining replication compression and the sparse data representation. The vector D was stored using 8-bit integers instead of 32-bit double precision numbers.

DataSet	Size	% Non-Zeros	Compr. Repl.	Mean Unique Val.	Total Comp.
cs.vu.nl	249,280x16	0.083 %	32	30	148
Msnbc	989,818x17	0.101 %	8.7	87	48
Webshop1	203,023x22	0.147 %	3.5	24	23.5
Webshop2	2,950,708x51	0.044 %	5.4	55	67

6 Experiments

We compared the running time of standard EM and the enhanced EM on several real world data sets. The datasets are: webshop1, webshop2, cs.vu.nl and msnbc. All the data sets originate from web log files of user behavior. Webshop1 is a data set from the pkdd challenge[13] of 2005. The data is a collection of sessions with counts of page views of customers visiting a webshop. The data is obtained after some preprocessing. Preprocessing entails page-type and session definition, search- and testbot removal and more, as described in [14]. Webshop2 is also from a webshop. Cs.vu.nl is extracted from the cs.vu.nl domain, the computer science domain of the Free University of Amsterdam. A thorough analysis of the data can be found in [15]. The data set Msnbc[16] is a data set available at the Irvine Data Mining Repository. Important statistics of the data sets are shown in Table 2. In the table the percentage of non-zero values, the percentage of duplicate records and the average number of unique values are listed. These statistics are important for the effectiveness of the “enhanced EM”.

The standard EM- and enhanced EM-algorithm were implemented in mex-matlab files, which are publicly available at [17]. Each experiment was repeated 5 times, which was observed to be sufficient for a reliable estimation of run-times.

6.1 Discussion

In figure 1 we show the speed-up factors and compression rate obtained by the enhanced version of the EM. The speed-up factor is the ratio between the time needed to run the basic EM and the “enhanced EM”. The speed-up factors were

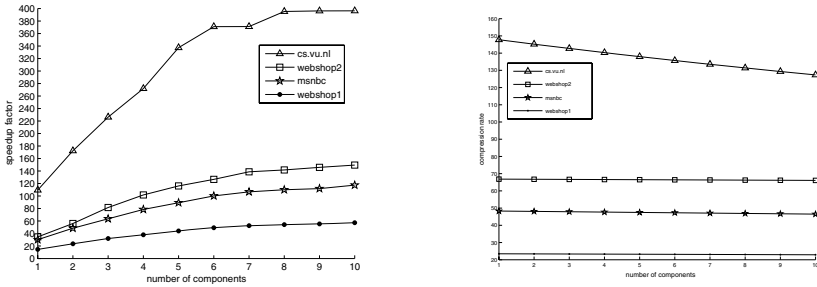


Fig. 1. Speed-up- and Compression ratio of enhanced EM as a function of the number of components on the data sets webshop1, webshop2, msnbc and cs.vu.nl

calculated using different number of components. The variance in the ratios are of an order of 10^{-3} , thus negligible. The speed-ups are in the range 15 to 110 for a very small number ($k = 1$) of components and 50 to 400 for the number of components ($k = 10$). The compression rate was calculated by calculating the amount of memory of the dense view divided by the amount of memory needed to store the sparse representation plus the reference table. Compression rates of 20 up to 150 were obtained. From the figure we see that the speed-up ratio is not linear in the number of components. With the increase of the number of components a decrease of the speed-up is observed. The data sets with higher number of duplicates achieved higher speed-up.

7 Conclusion

We proposed some simple techniques to boost the performance of the standard implementation of EM for mixture of multinomial in terms of computation effort and memory load.

These simple techniques were: removal of duplicate records, not storing and not calculating zero-values, and building a reference table of all encountered combinations of counts and parameter values to save the number of exponentials to be calculated.

We have shown that using these techniques huge speed-up, in the order of 50 to 400, and memory compression 20 up to 150, could be achieved. We conclude that these simple techniques are very effective in the context of internet data.

References

1. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: an efficient data clustering method for very large databases. In: SIGMOD '96: Proceedings of the 1996 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM Press (1996) 103–114
2. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B* **39** (1977) 1–38

3. Moore, A.: Very fast EM-based mixture model clustering using multiresolution kd-trees. In Kearns, M., Cohn, D., eds.: *Advances in Neural Information Processing Systems*, 340 Pine Street, 6th Fl., San Francisco, CA 94104, Morgan Kaufman (1999) 543–549
4. Bradley, P., Fayyad, U., Reina, C.: Scaling EM clustering to large databases, Microsoft research report, msr-tr-98-35 (1998)
5. Domingos, P., Hulten, G.: Learning from infinite data in finite time. *Advances in Neural Information Processing Systems* **14** (2002) 673–680
6. Verbeek, J., Nunnink, J., Vlassis, N.: Accelerated EM-based clustering of large datasets. *Data Mining and Knowledge Discovery*. In Press. (2006)
7. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M.I., ed.: *Learning in Graphical Models*, Kluwer (1998)
8. Cadez, I.V., Smyth, P., Mannila, H.: Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction. In: *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, ACM Press (2001) 37–46
9. Thiesson, B., Meek, C., Heckerman, D.: Accelerating EM for large databases. *Machine Learning* **45** (2001) 279–299
10. Blekas, K., Likas, A.: Incremental mixture learning for clustering discrete data. *Proc. 3rd Hellenic Conference on AI, SETN 2004* (2004) 210–219
11. Cadez, I.V., Heckerman, D., Meek, C., Smyth, P., White, S.: Visualization of navigation patterns on a web site using model-based clustering. In: *Knowledge Discovery and Data Mining*. (2000) 280–284
12. Vakali, A., Pokorn, J., Dalamagas, T.: An overview of web data clustering practices. *Lecture Notes in Computer Science* **3268** (2004) 597–606
13. Nasraoui, O., Zaiane, O.R., Spiliopoulou, M., Mobasher, B., Masand, B., YU, P.S.: *Webkdd 2005: web mining and web usage analysis post-workshop report*. *SIGKDD Explor. Newsl.* **7**(2) (2005) 139–142
14. Hofgesang, P., Kowalczyk, W.: Analysis clickstream data: From anomaly detection to visitor profiling, *ECML/PKDD Discovery Challenge 2005* (2005)
15. Hofgesang, P.: *Web usage mining - Structuring enriched clickstream data*, Msc. Thesis, Free University Amsterdam, The Netherlands (2005)
16. UCI-KDD-archive: (msnbc.com, University of California, Irvine, <http://kdd.ics.uci.edu/>)
17. Patist, J.P.: Toolbox for multinomial mixture building (www.few.vu.nl/~jpp/multimix.rar)

A Novel Approach to Pattern Recognition Based on PCA-ANN in Spectroscopy

Xiaoli Li and Yong He*

College of Biosystems Engineering and Food Science, Zhejiang University, 310029,
Hangzhou, China
yhe@zju.edu.cn

Abstract. Pattern recognition problems that involve functional predictors has developed, specifically for spectral data. The classification of three peach varieties based on near infrared spectra was researched in the practical context. Principal component analysis (PCA) and artificial neural networks (ANN) were used for pattern recognition in this research. PCA is a very effective data mining way; it is applied to enhance species features and reduce data dimensionality. ANN with back propagation algorithm was used for the data compression tasks as well as class discrimination tasks. The first 9 principal components computed by PCA were applied as inputs to a back propagation neural network with one hidden layer. This model was used to predict the varieties of 15 unknown samples. The recognition rate of the model for the unknown sample was 100%. So this paper could offer an effective pattern recognition way.

1 Introduction

Pattern recognition with functional predictor data has become prominent in many fields in recent years. NIR combined with pattern recognition techniques have attracted considerable attention for the purpose of discrimination between sets of similar biological materials such as coffee varieties [1], leaves of strawberry plants of five different varieties [2], melon genotypes [3], soybean varieties [4], apple varieties [5].

A NIR spectrum of a sample is typically measured by modern scanning instruments at hundreds of equally spaced wavelengths. The information in the curve is used to predict the chemical composition of the sample by extracting the relevant information from many overlapping peaks. So, how to extract useful information from mass original spectral data is a pivotal step. Osborne et al. [6] described standard approaches, such as linear discriminant analysis (LDA), however these methods failed with many variables and different approaches need to be taken [7]. Wu et al. [8] compared several methods for classification based on mass spectra, including linear and quadratic discriminant analysis and classification trees methods. In their conclusions the authors emphasized the need for methods to remove noise from the data and select relevant features. Wavelet transforms was employed to reduce dimension and noise removal [9] [10].

In qualitative and quantitative analysis, artificial neural network is more and more widely applied during the past several years. The better advantage of ANN is its

* Corresponding author.

anti-jamming, anti-noise and robust nonlinear transfer ability. In proper model, ANN results in lower calibration errors and prediction errors. Back-propagation (BP) is an ANN algorithm most widely used in chemometrics practice. But if the original spectral data were directly used as the input of the ANN, the train time of ANN model would be excessive, and overfitting model would be produced. So the original spectral data must be compressed. Principal component analysis is a very effective data mining technique for spectroscopic data. It summarizes data by forming new variables, the new variables (principal components) are a set of orthogonal variables and represent the most common variations to all the data.

In this research, we put forward a new method of pattern recognition called PCA-ANN. The dimension of the spectral data was reduced by using principal components analysis, and the relevant features were extracted from mass spectra. The BP-ANN was used to build the classification model based on the relevant features. The new method not only can qualitative analyze the varieties of the peach, but also can accurately predict the peach varieties of unknown samples.

2 Material and Methods

2.1 Fruit Samples and Spectra Measurements

A total of 75 peaches were used in this research. There were three species, which were Milu peach (from Fenghua of Zhejiang, China), Dabaitao peach (from Jinhua of Zhejiang, China) and Hongxianjiu peach (from Shandong, China). The number of each species was 25. From each fruit, three reflection spectra (325–1075 nm) were taken at three equidistant positions around the equator [11] by a spectrometer. To reduce the noise, the smoothing way of moving average was used. The segment size of smoothing was 3. The second type of preprocessing was the S. Golay second-derivative. This technique was used to eliminate the disturbance from the spectra and the background. It can also extract information from the overlap peak. So the resolution and the sensitivity of the model can be improved. The pre-process and calculations were carried out using "The Unscrambler V9.2". Due to of noise the first 75 and the last 75 wavelength values were take out of all analysis, starting from here all the considerations were based on this range of wavelengths (400-1000 nm) [12].

2.2 Principal Components Analysis

We will use principal component analysis, (PCA), frequently throughout the paper to view the multi-dimensional data. That is, the PCA computes a linear projection of the multidimensional measured data onto a 2-D or 3-D viewable surface. By allowing the observations to depend on the N -dimensional random variable $X^T = [X_1, \dots, X_N]$, the principal components (PCs), P_1, \dots, P_N can be derived from both the covariance matrix C_V as well as from the correlation matrix C_r . All variables in the original data are believed to be equally important. It is therefore most natural to work with standardized variables. That is, for each variable x_i , subtract the mean μ of x_i , $(x_i - \mu)$, so that all the variables will have unit variance, and thus find the PCs from the correlation matrix, C_r . Theory of PCA can be found in almost any textbook about cluster analysis.

Each principal component PC_i is derived as a linear combination of X,

$$PC_i = pc1i X1 + pc2i X2 + \dots + pcNi XN = pciTX \tag{1}$$

where pci^T is a vector of constants. It can be shown that the variance of PC_i is,

$$\text{Var}(PC_i) = \text{Var}(pciTX) = \lambda_i \tag{2}$$

where λ_i is the eigenvalues of the C_r , thus the corresponding PC is the eigenvector. The first PC, PC_1 , should account for the maximum variance, thus we arrange the eigenvalues λ_i in a decreasing order so that $\lambda_1 > \lambda_2 > \dots > \lambda_N$. PC_1 will then correspond to the eigenvector of λ_1 . The next PC, PC_2 , is then the second eigenvector and so on. This, of course, includes the assumption that the eigenvalues can be arranged in a decreasing order. If we have several eigenvalues of equal magnitude, then the corresponding PC cannot be identified in this manner. However, it is usually pointless trying to identify too many PCs and furthermore, if the eigenvalue sequence stops and becomes equal, then no more PCs should be taken into account.

A well known feature of the correlation matrix is that all diagonal terms are equal to unity. Thus, the sum of the diagonal terms will be equal to N (the diagonal sum is equal to the sum of the variance of the standardized variables). This automatically gives that and the percentage of the total variance that each PC explains is simply λ_i/N .

$$\sum \lambda_i = N \tag{3}$$

2.3 Artificial Neural Network (ANN)

ANN with back propagation algorithm can be used for the data compression tasks as well as class discrimination tasks. Three-layered back propagation network architecture was used for developing neural classifiers for sorting peaches based on varieties. A schematic diagram of multilayer neural network architecture is shown in Fig. 1.

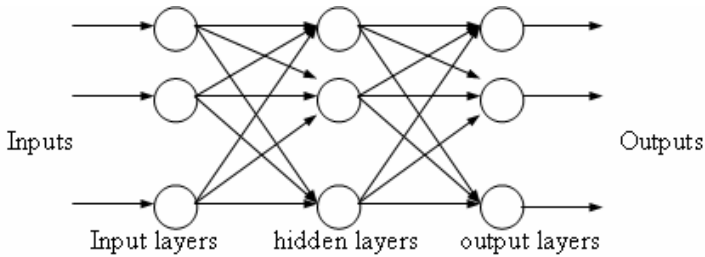


Fig. 1. Schematic diagram of the topological structure of BP neural network

The learning process of BP neural network consists of the feature of back-propagation training. The inputs run through hidden units towards the output layers and neurons in each layer influence only those in the immediate one. If the outputs are not desired, the latter goes instead, i.e., the outputs run back through the hidden neuron connection and reduce the total error by adjusting the weight of each layer until all the errors are within the required tolerance.

3 Results and Discussion

3.1 Features of Vis/NIR Spectra

Fig. 2 shows the average absorbance spectra of three varieties of peach: Hongxianjiu peach, Dabaitao peach, Milu peach obtained from the spectrograph. The spectral curves of three varieties peaches were similar, however after careful observation, the tiny differences among these varieties can be detected in the Vis/NIR spectra from 600 nm to 700 nm, which makes it possible to discriminate the three peaches varieties. Qualitative clustering is achievable based on these spectral differences. The differences may be caused by the different internal attribute of the three varieties peaches, such as the sugar and acidity. The sugar and acidity contents can be seen in Table 1.

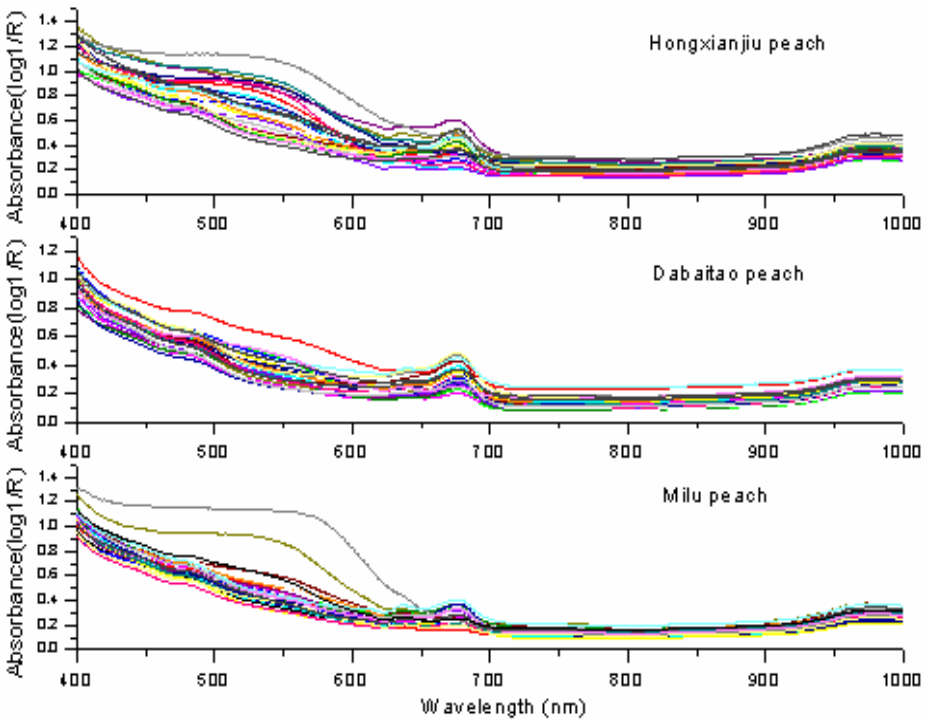


Fig. 2. Visible/Near infrared absorbance spectra of three different varieties peaches

Table 1. Important internal attribute of these peaches

Characteristic Item	Sugar content			Acidity content		
	Mean value	Range	Std. dev.	Mean value	Range	Std. dev.
Milu	11.6	8.8-13.9	1.35	4.3	3.6-5.0	0.41
Hongxianjiu	8.8	6.0-11.0	1.22	4.4	4.1-4.9	0.19
Dabaitao	10.4	7.5-13.0	1.42	4.6	4.4-4.8	0.13

3.2 Clustering Based on PCA

PCA was performed on the absorbance of spectra of the 75 samples, and hence many principal components could be achieved to replace the mass spectral data, which is rich but information poor. If the scores of one certain principal component were organized according to the number of the sample, a new score plot could be created. The new image was then called 'PCA scores image' [13]. When the principal component scores were plotted they may reveal natural patterns and clustering in the samples. If the first principal component (PC1) scores and the second principal component (PC2) scores were manipulated, the resultant image was then called PC1 and PC2 scores image, just as Fig. 3.

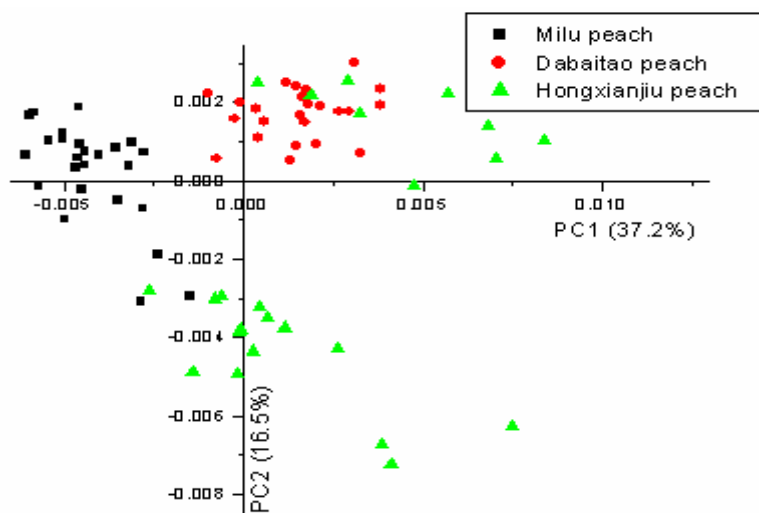


Fig. 3. Scores plots obtained from the PCA (PC1×PC2) of 75 samples

The scatter plot of PC1 (variability; 37.2%)×PC2 (variability; 16.5%) is shown in Fig. 3. The plot shows clear separation based on the peach species. In this scatter plot, the differences among Milu peach, Hongxianjiu peach and Dabaitao peach were pronounced. In short, Dabaitao peach samples closely cluster near to the positive of Y-coordinate; while Milu peach samples are almost situated round the negative of abscissa. In contrast, the Hongxianjiu samples are scattered over the quadrant where PC1 is positive and PC2 is negative. In a word, the three varieties of peaches can be discriminated base on the PC1×PC2 score plot, but the effect of clustering wasn't very well, especially the Hongxianjiu peach was dispersed. So, it can be concluded that PCA can extract the effective information from the mass original spectral data; and qualitatively differentiate the varieties of peaches [14].

3.3 Classification of Peach by ANN

PCA finds an alternative set of coordinate axes, PCs, about which data set may be represented. The PCs are orthogonal to each other and they are ranked so that each

one carries more information than any of the following ones. The number of principal components was decided by cross-validation. The first 9 principal components were enough to explain the 85.3% of the data variance. So the 9 new variables can replace the spectral variables. The new matrix that contains 75 rows and 9 columns comes into being.

Table 2. The prediction result for unknown samples by BP-ANN model

No.	Calculated Value		No.	Calculated Value		No.	Calculated Value	
(1)	1.5E-11	0.0062	(6)	09E-7	0.9999	(11)	1	0
(2)	3.9E-4	0	(7)	0.0001	0.9999	(12)	1	2.6E-8
(3)	7.6E-14	0	(8)	2.9E-8	1	(13)	1	0
(4)	3.9E-4	2.0E-15	(9)	0.0038	1	(14)	1	0
(5)	0	0.0062	(10)	0	1	(15)	1	2.2E-8
SV	0	0	SV	0	1	SV	0	1

Note: (1)-(5), Milu peach;(6)-(10),Dabaitao peach;(11)-(15), Hongxianjiu peach.
SV--Standard Value, No.--sample number.

The total wavebands were replacing by the 9 characteristic variables. The whole samples were separated randomly into two parts, one part that contains 60 samples was used as reality validation samples, and the other was used as predicting samples. The matrix that contained 60 samples and 9 variables was used to build the BP-ANN model. The optimal architecture of neural network can be achieved by adjusting nodes of the hidden layer [15]. A three-layer ANN was built. A BP neural network model with three-layers was built. The transfer function of hidden layer was tansig function. The transfer function of output layer was logsig function. The train function was trainlm. The goal error was set as 0.00001. The time of training was set as 1000. As there were three different classes samples, the output vectors of these samples were assumed to be the two binary codes, So, the output binary system vectors (0 0) be denoted as the Milu peach, (0 1)be denoted as the Dabaitao peach, (1 0)be denoted as the Hongxianjiu peach. This BP model had been used to predict the varieties of the 15 unknown samples; the recognition rate is 100% (seen in table 2).

The result was superior to those obtained by Lenio Soares Galvao [16] in sugarcane varieties with 87.5% classification accuracy. And better than those obtained by Haluk Utku in wheat varieties with recognition of unknown samples (82%, 81%) [17] based on orthonormal transformation. The tradition classical evaluation method (PCA, linear discriminate analysis (LDA) [18], PCA+LDA) almost can make qualitative analysis with the sample variety; however they didn't have a high precision to predict the varieties of unknown samples compared to this research.

4 Conclusion

In this research, we put forward a new method called PCA-ANN, By means of this new method a specific correlation was established between reflectance spectra and varieties of peaches. The model for the varieties of peach showed an excellent prediction performance. The recognition rate of 100% was achieved. The above results

indicate that the PCA-ANN is an effective pattern recognition way for discriminating. This model can not only make qualitative analysis but also quantitatively classify the sample; moreover the precision of classification is superior to former models.

Acknowledgements

This study was supported by the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of MOE, P. R. C., Natural Science Foundation of China, Specialized Research Fund for the Doctoral Program of Higher Education (Project No: 20040335034), and Science and Technology Department of Zhejiang Province (Project No. 2005C21094, 2005C12029).

References

1. Esteban, D.I., Gonzalez-Saiz, J.M., Pizarro, C.: An Evaluation of Orthogonal Signal Correction Methods for the Characterisation of Arabica and Robusta Coffee Varieties by NIRS. *Analytica. Chimica. Acta.* 514 (2004) 57-67
2. Lopez, M.: Authentication and Classification of Strawberry Varieties by Near Infrared Spectral Analysis of Their Leaves. In: Cho R.K., Davies A.M.C. (Eds.): *Near Infrared Spectroscopy: Proceedings of the 10th International Conference*, NIR Publications. Chichester, UK. (2002) 335-338
3. Seregely, Z., Deak, T., Bisztray, G. D.: Distinguishing Melon Genotypes Using NIR Spectroscopy. *Chemometrics and Intelligent Laboratory Systems.* 72 (2004) 195-203
4. Turza, S., Toth, A., Varadi, M.: Multivariate Classification of Different Soybean Varieties, In: Davies A.M.C. (Ed.). *Journal of Near Infrared Spectroscopy: Proceedings of the 8th International Conference*, NIR Publications. Chichester, UK. (1998) 183-187
5. He, Y., Li, X.L., Shao, Y.N.: Discrimination of Varieties of Apple Using Near Infrared Spectral by Principal Component Analysis and BP model. *Spectroscopy and Spectral Analysis.* 5 (2006)
6. Osborne, B.G., Fearn, T., Hindle, P.H.: *Practical NIR Spectroscopy*. Longman, Harlow, U.K. (1993)
7. Krzanowski, W.J., Jonathan, P., McCarthy, W.V., Thomas, M.R.: Discriminant Analysis with Singular Covariance Matrices: Methods and Applications to Spectroscopic Data. *Applied Statistics.* 44 (1995) 105-115
8. Wu, B., Abbott, T., Fishman, D., McCurray, W., Mor, G., Stone, K., Ward, D., Williams, K., Zhao, H.: Comparison of Statistical Methods for Classification of Ovarian Cancer Using Mass Spectrometry Data. *Bioinformatics.* 19 (2003) 1636-1643
9. Qu, Y., Adam, B.L., Thornquist, M., Potter, J.D., Thompson, M.L., Yasui, Y., Davis, J., Schellhammer, P.F., Cazares, L., Clements, M.A., Wright Jr., G.L., Feng, Z.: Data Reduction Using a Discrete Wavelet Transform in Discriminant Analysis of Very High Dimensionality Data. *Biometrics.* 59 (2003) 143-151
10. Vannucci, M., Sha, N.J., Brown, J. P.: NIR and Mass Spectra Classification: Bayesian Methods for Wavelet-based Feature Selection. *Chemometrics and Intelligent Laboratory System.* 77 (2005) 139-148
11. Pereira, A.G., Gomez, A.H., He, Y.: Advances in Measurement and Application of Physical Properties of Agricultural Products. *Transactions of the CSAE.* 19.No.5 (2003)7-11

12. Qi, X.M., Zhang, L.D., Du X.L.: Quantitative Analysis Using NIR by Building PLS-BP Model. *Spectroscopy and Spectral Analysis*. 23.No.5 (2003) 870-872
13. He, Y., Feng, S.J., Deng, X.F., Li, X.L.: Study on Lossless Discrimination of Varieties of Yogurt Using the Visible/NIR-spectroscopy. *Food Research International*, Vol. 39. No. 6 (2006)
14. Dai, S.X., Xie, C.J., Chen, D.: Principal Component Analysis on Aroma Constituents of Seven High-aroma Pattern Oolong Teas. *Journal of South China Agriculture University*. 20.No.1 (1999) 113-117
15. Zhao, C., Qu, H.B., Cheng Y.Y.: A New Approach to the Fast Measurement of Content of Amino Acids in *Cordyceps Sinensis* by ANN-NIR. *Spectroscopy and Spectral Analysis*. 24.No.1 (2004) 50-53
16. Galvao, L.S., Formaggio, A.R., Tisot, D.A.: Discrimination of Sugarcane Varieties in Southeastern Brazil with EO-1 Hyperion Data. *Remote Sensing of Environment*. 94 (2005) 523-534
17. Utku, H.: Application of the Feature Selection Method to Discriminate Digitized Wheat Varieties. *Journal of Food Engineering*. 46 (2000) 211-216
18. Krzanowski, W.J., Jonathan, P., McCarthy, W.V., Thomas, M.R.: Discriminant Analysis with Singular Covariance Matrices: Methods and Applications to Spectroscopic Data. *Applied Statistics*. 44 (1995) 105-115

Semi-supervised Dynamic Counter Propagation Network

Yao Chen and Yuntao Qian

College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China
rhyemenchen@163.com, y tqian@zju.edu.cn

Abstract. Semi-supervised classification uses a large amount of unlabeled data to help a little amount of labeled data for designing classifiers, which has good potential and performance when the labeled data are difficult to obtain. This paper mainly discusses semi-supervised classification based on CPN (Counter-propagation Network). CPN and its revised models have merits such as simple structure, fast training and high accuracy. Especially, its training scheme combines supervised learning and unsupervised learning, which makes it very conformable to be extended to semi-supervised classification problem. According to the characteristics of CPN, we propose a semi-supervised dynamic CPN, and compare it with other two semi-supervised CPN models using Expectation Maximization and Co-Training/Self-Training techniques respectively. The experimental results show the effectiveness of CPN based semi-supervised classification methods.

1 Introduction

Traditionally, designing a classifier is a supervised learning task, which needs a lot of labeled data or expert knowledge. Manually labeling data is not only time-consuming, but also difficult. In many applications (e.g. web classification, handwritten digits recognition), there is a large supply of unlabeled data but limited labeled data. Consequently, semi-supervised classification, using a large amount of unlabeled samples to help a little amount of labeled samples for designing classifier, has become a topic of significant recent interest.

Up to now, there are many semi-supervised classification methods [1,2]. They can be divided into two categories according to their nature of using unlabeled data, iterative method and incremental method. In the iterative method, an initial classifier is built by the labeled data and then it is used to estimate the temporary labels for all unlabeled data. By the combination of labeled data and unlabeled data with temporary labels, the classifier adjusts its parameters. Iteratively estimating and adjusting, the iterative algorithm finally reaches a local optimum (e.g. Expectation Maximization, EM [3]). In contrast with the iterative method, an incremental method converts the unlabeled data to labeled data bit by bit. It uses labeled data to build an initial classifier. The classifier then estimates the labels for the unlabeled data and pick up some data the classifier is most confident with. These data with their class labels are added to labeled data set. The classifier updates its parameters using the updated labeled data set. The algorithm repeats until the unlabeled pool is empty, (e.g. Co-Training [4], Self-Training [5]). These two different semi-supervised methods both need a specific

classifier to do classification. In this paper, we use CPN (Counter Propagation Network) [6] as the basic classifier. CPN has merits such as simple structure, fast training and high accuracy. Especially, its training scheme combines supervised learning and unsupervised learning, which makes it very conformable to be extended to semi-supervised classification problem. According to the characteristics of CPN, we propose a semi-supervised dynamic CPN (SSDCPN) which dynamically generates middle layer neurons using the unlabeled data so that the model can better reflect data distribution. Experimental results have shown that semi-supervised classifier outperforms supervised CPN and SSDCPN is also better than the other ordinary semi-supervised CPNs that directly use semi-supervised learning scheme, such as CPN+EM, CPN+Co-Training, and CPN+Self-Training.

This paper is organized as follows. In the next section, we introduce the principle of CPN, semi-supervised learning schemes and problems with some semi-supervised CPN models. In Section 3, we specify the algorithm of SSDCPN. The experiment results are discussed in Section 4. Finally, we present the conclusions and summary in Section 5.

2 Related Work

2.1 Counter Propagation Network (CPN)

CPN is a neural network which can do data compression, pattern recognition and function approach. Fig.1 shows the structure of CPN. CPN is composed of three layers, an input layer, a middle layer called K-layer and an output layer. The K-layer is based on the Self-Organizing Map proposed by Kohonen. The output layer is based on the out-star structure proposed by Grossberg. Every neuron in K-layer has its class label. The weight \mathbf{w}_j between the input layer and the K-layer presents common characteristics of the input data belonging to class j and the weight \mathbf{v}_j between the K-layer and the output layer stores the common characteristics of the output of class j .

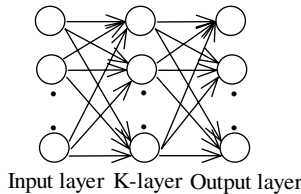


Fig. 1. The CPN

All the input vector $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ should be normalized first, so that

$$\left(\sum_i x_i^2\right)^{1/2} = 1 \tag{1}$$

The training of CPN can be divided into two steps, first adjusting the weight matrix W between input layer and the K-layer to reflect the distribution of input data and second adjusting the weight matrix V between K-layer and the output layer, to simulate the output of each class.

In the first step, when an input vector \mathbf{x} is applied to the input layer, the activation s_j is calculated for every neuron.

$$s_j = \mathbf{w}_j \cdot \mathbf{x} = \sum_i w_{ji} x_i \tag{2}$$

A competition is held between each K-layer neurons and the neuron with the largest activation is the winner. The output is one for the winner and zero for all other neurons.

$$z_j = \begin{cases} 0 & j \neq c \\ 1 & j = c \end{cases} \tag{3}$$

In ordinary CPN, K-layer is trained in an unsupervised manner. And only the weight \mathbf{w}_c between the winner neuron c and the input layer is to be updated.

$$w_{ci}(t+1) = w_{ci}t + \alpha(x_i - w_{ci}(t)) \tag{4}$$

In the second step, algorithm adjusts the weight between the K-layer and the output layer.

$$v_{ji}(t+1) = \begin{cases} v_{ji}(t) & i \neq c \\ v_{ji}(t) + \beta(y_j - y'_j) & i = c \end{cases} \tag{5}$$

$$y'_j = \sum_i z_i v_{ji}(t) \tag{6}$$

y_j is the expecting output.

In fact, in the first step of training, the weight matrix W can also be trained in a supervised manner [7,8]. The weight vector \mathbf{w}_c is updated using equation (7). Since each neuron has its own class label, if the winner gives the correct output, the winning weight vector will be moved towards the input vector. Otherwise, it will be moved further away from it.

$$w_{ci}(t+1) = \begin{cases} w_{ci}t + \alpha(x_i - w_{ci}(t)) & \text{Classification is correct} \\ w_{ci}t - \alpha(x_i - w_{ci}(t)) & \text{Classification is incorrect} \end{cases} \tag{7}$$

In this case when K-layer is trained in a supervised manner, and every neuron has its own class label, we only need to store the class information in the V in the second step.

2.2 Combine the Semi-supervised Schemes with CPN

EM+CPN. EM is an iterative statistical technique for maximum likelihood estimation in problems with incomplete data. In implementation, EM is an iterative two-step

process, E-Step and M-Step. The E-step calculates probabilistically-weighted class labels, for every unlabeled data. The M-step estimates new classifier parameters using all the data. It iterates the E-steps and M-steps until the training converges.

The EM+CPN algorithm first calculates the parameters using the labeled data and iterates the following two steps:

E-Step: For every unlabeled data $x \in U$, calculate the expected label $l(x)$ using the current classifier.

M-Step: Combining the labeled data and the unlabeled data with their temporary labels to update the parameters of CPN.

The algorithm converges when difference of class labels between consecutive two iterations is smaller than ε .

Co-Training+CPN and Self-Training+CPN. Co-Training is an incremental method for semi-supervised learning [5]. Its approach is to incrementally build two different classifiers over two different feature sets of the data and allow them to learn from each other. When Co-Training was proposed, there are two assumptions: 1. The two feature sets should be independent. 2. Each of the two sets should be sufficient to classify the data set. But in real applications, these two assumptions are hardly satisfied. In many cases, we can only get one single feature set. In fact, as long as the feature is redundant enough, using random split of the feature set can be useful [9]. The algorithm for Co-Training is as following:

Input: Labeled data pool L , unlabeled data pool U , feature sets F_1, F_2 .

While U is not empty

Build classifier C_1 using feature F_1

Build classifier C_2 using feature F_2

For each classifier C_i do

C_i labels the unlabeled data in U

Select p data from U which C_i is most confident as an incremental set E

Delete E from U , add E and the estimated label to L

End For

End While

Output: Two classifier C_1, C_2 that predicate label for new data. And these predictions can be combined by multiplying together

Self-Training is a hybrid of Co-Training. It is also an incremental algorithm, but it does not use the split of the features [5]. We use the self-Training as a contrast to Co-Training to test if the feature split improves the accuracy.

3 Semi-supervised Dynamic CPN (SSDCPN)

The principle idea of CPN is to create appropriate clusters in K-layer, so that the weights between the input layer and the K-layer can present characteristics of each class. In complicate classification problems, data from the same class can be widely

distributed in space, so there should be more than one neuron in K-layer to recognize them. For example, in handwritten digits classification, different writing style leads to different subsets [7]. An important factor that determines the accuracy of CPN is whether appropriate clusters can be generated in K-layer and these clusters should present both probability density distribution and the label information. So combining the unsupervised probability density distribution estimation and the supervised guiding for labels is an effective way to improve the classification accuracy, which provides a natural base for extending CPN to semi-supervised classification.

SSDCPN uses a scheme similar to EM. The algorithm builds an initial classifier using labeled data and estimates the labels for unlabeled data. Then it updates the parameters of CPN using the combination of them. In the process, SSDCPN uses a large amount of unlabeled data to dynamically generate new nodes in K-layer to present the complicate data distribution. In CPN, supervised training is applied in K-layer; and the activation $s_i = \mathbf{w}_i \cdot \mathbf{X}$ presents the similarity between the input data and the characteristics of that class. A threshold T_s of the activation for each neuron is given, presenting the area of each neuron currently covers. When unlabeled data is applied, if the winning activation is less than T_s , we consider this input vector belonging to an unknown subset. To certain possibility, a new neuron is generated to present it. Because it is difficult to determine T_s , the model dynamically updates it during the training.

Given labeled data set and the corresponding label set

$$L = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4 \dots \mathbf{x}_l\}, T = \{y_1, y_2, y_3, y_4 \dots y_l\},$$

Unlabeled data set:

$$U = \{\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \mathbf{x}_{l+3}, \mathbf{x}_{l+4} \dots \mathbf{x}_{l+u}\}, \text{ usually } u \gg l$$

The algorithm of SSDCPN is as follows:

The initial neuron number in K-layer is set to be class number c , initial weight $\mathbf{w}_i = \mathbf{x}_i$ ($0 \leq i \leq c$), it is the mean vector of labeled data from class i .

Step1 (Training using labeled data) :Using labeled data set L , train the CPN until convergence and in the mean time we record the smallest activation of each neuron as the threshold of creating neuron in step 2

Step2 (Training using unlabeled data) :For each unlabeled data $\mathbf{x}_i \in U$, K_i is the winning neuron, and the temporary label of \mathbf{x}_i is set to be the class label of K_i . If the activation of K_i , $s_i < T_s$, there is a probability p that a new neuron will be created.

$$p = \rho (T_s - s_i)(1 - N^2 / R^2)$$

R: maximum number of neuron in K-layer, N: current neuron number, ρ : a parameter to control the probability

The further the input vector is from the class characteristics, the larger $(T_s - s_i)$ would be. And it leads to higher probability to create new neuron. The $(1 - N^2 / R^2)$ part limits the number of neurons. If a new neuron is generated, its weight $\mathbf{w}_{N+1} = \mathbf{x}_i$, and N increases by one. The class label for the new neuron is set by the dominating class associated with this neuron.

If the activation $s_i \geq T_s$, update the weight using equation (4)

If $\|W(t) - W(t-1)\| < \varepsilon$, go to end. Otherwise go to Step 2.

4 Experiments and Analysis

We apply Co-Training+CPN, Self-Training+CPN, EM+CPN, SSDCPN to four data sets from UCI Database [11] and compare the results with that of CPN using only labeled data. We randomly select some data from the training set as labeled data and other data as unlabeled data. Table 1 shows the features of the four algorithms. Since Co-Training using a random split of feature needs enough redundancy in data sets, we test the redundancy of each data set as follows [5]. We split the feature into two different halves, Feature set 1 and Feature set 2, then build three different CPN classifiers C_0 , C_1 and C_2 . C_0 uses all the features. C_1 and C_2 use Feature set1 and Feature set2 respectively. The difference between the accuracy of the two kinds of classifier indicates the redundancy of the features. If the difference is small, there is significant redundancy. Otherwise, there is not enough redundancy.

Table 1. Comparison among four semi-supervised CPN algorithms

Methods	Incremental/Iterative	Using feature split	CPN structure
Co-Training+CPN	Incremental	Yes	Fixed
Self-Training+CPN	Incremental	No	Fixed
EM+CPN	Iterative	No	Fixed
SSDCPN	Iterative	No	Dynamic

4.1 Experiment A

The first experiment uses the data set named ‘‘Optical Recognition of Handwritten Digits’’, which has 3823 training data and 1797 test data of 10 classes. The dimension of the data is 64. In this data set, the accuracy between the classifier C_0 and classifier C_1, C_2 is less than 4% on training data, and less than 5.5% on test data. This indicates significant redundancy in the feature.

Fig.2(left) shows the results of Co-Training+CPN, Self-Training+CPN and CPN. It is clear that the former two semi-supervised algorithms making use of unlabeled data outperform the latter one and Co-Training+CPN outperforms Self-Training+CPN due to the enough redundancy of the feature.

Fig.2(right) shows the comparison of four semi-supervised algorithms and supervised CPN algorithm. For fixed CPN structures, the maximum neuron number is 40, and for SSDCPN the maximum neuron number is 60. The result shows semi-supervised algorithms have higher accuracy than the supervised algorithm using only labeled data. But as the labeled rate continues to increase, the improvement reaches a saturated point. When the labeled data is very rare, SSDCPN does not give a pleasing result. That is because too few labeled data can not properly conduct the dynamic growing of K-layer. As the labeled data increases, it has a better performance than other three semi-supervised algorithms.

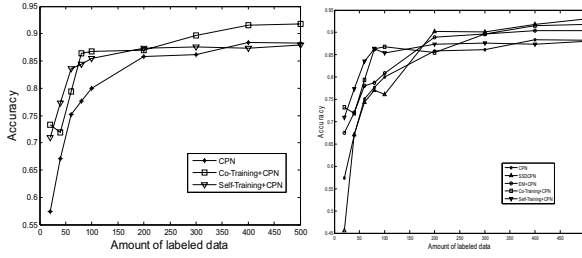


Fig. 2. Experimental results on Optdigits

4.2 Experiment B

The second data set used is “Isolated Letter Speech Recognition”, which contains voices of 26 English letters by 150 different people. It has 617 dimensions and 26 classes. The training set contains 6238 samples from 120 people and the test set contains 1559 samples from other 30 people. The accuracy difference between classifier C_0 and classifiers C_1, C_2 is less than 2% both on training data and test data. This indicates significant redundancy in the feature.

Fig.3 shows the results five algorithms. For fixed CPN structures, the maximum neuron number is 200, and in the SSDCPN the maximum neuron number is 260.

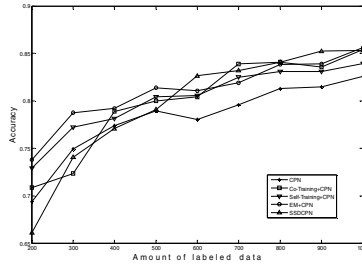


Fig. 3. Experimental results on Isolet

4.3 Experiment C

The third data set used is “Pen-Based Recognition of Handwritten Digits”, which contains 7494 training data and 3498 test data with 16 dimensions and 10 classes. The accuracy difference between classifier C_0 and classifiers C_1, C_2 is as much as 10% on test data set. This indicates less redundancy in the feature than the former two data sets we use.

Fig.4 shows the results of the five algorithms. For fixed CPN structures, the maximum neuron number is 60, and in the SSDCPN the maximum neuron number is 100. We can see from the result that on data set which does not have enough redundancy, Co-Training+CPN does not outperform Self-Training+CPN.

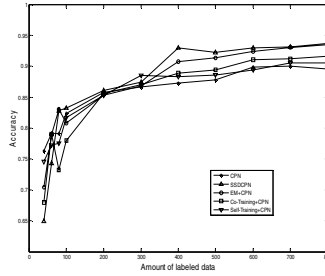


Fig. 4. Experimental results on Pen-digits

4.4 Experiment D

The forth data set used is “Image segmentation database”, which contains 2100 data with 19 dimensions and 7 classes. We randomly pick up 600 data as test set and other as training set. The accuracy difference between classifier C_0 and classifiers C_1, C_2 is as much as 65%. This indicates the redundancy in the feature is very low. In this case, Co-Training does not apply.

But the SSDCPN and EM+CPN still work. Table 2 shows the results of these two semi-supervised algorithms and supervised CPN algorithm. In CPN and EM+CPN, the maximum neuron number is 60, and in the SSDCPN the maximum neuron number is 100.

Table 2. Experimental results on Imageseg

Labeled data amount	CPN	EM+CPN	SSDCPN
100	63.17%	64.83%	69.17%
200	71.50%	73.67%	77.50%
300	72.00%	74.33%	78.00%

In the experiments above, it is shown that although semi-supervised algorithms gain higher accuracy than supervised algorithms, when the labeled rate is too low, the improvement is small and decrease may occur. This is because when the assumed probabilistic model does not match the true data generating distribution, using unlabeled data can be detrimental to the classification accuracy [10]. This often happens when labeled data set is too small. When the data set has enough redundancy, Co-Training+CPN has a higher accuracy than Self-Training+CPN. And SSDCPN outperforms other semi-supervised algorithms in many cases.

5 Conclusions

In this paper, a novel semi-supervised CPN model, SSDCPN, is proposed. This model makes full use of the advantage of the original algorithm of CPN which combines supervised and unsupervised learning schemes, and dynamically generates the model

structure using unlabeled data. The experiment shows its good performance. In addition, we introduce other semi-supervised CPN models which directly use incremental and iterative schemes and compare their performances on several data sets. In most cases, the semi-supervised algorithms can improve the classification accuracy, but the improvement is influenced by some factors such as redundancy in feature and labeled rate. Therefore, in practice we should choose different algorithms for different applications.

References

1. K. P. Bennett, A. Demiriz. Semi-Supervised Support Vector Machines. In: Proceedings of Neural Information Processing Systems. Denver: MIT Press, 1999. 368-374
2. K. Nigam, A. McCallum, S. Thrun, T. Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 2000, 39(2-3): 103-134
3. T. K. Moon. The Expectation Maximization Algorithm. *Signal Processing Magazine*, 1996, 13(6): 47-60
4. B. Avrim, T. Mitchell. Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory. Madison, Wisconsin, United States: ACM Press, 1998: 92-100.
5. K. Nigam, R. Ghani. Analyzing the Effectiveness and Applicability of Co-training . Ninth International Conference on Information and Knowledge Management. McLean, Virginia, United States: ACM Press, 2000: 86-93.
6. R. H. Nielsen. Counter propagation networks. *Applied Optics*, 1987, 26(23): 4979-4983.
7. I. P. Morns, S. S. Dlay. The DSFPN, a New Neural Network for Optical Character Recognition. *IEEE Transactions on Neural Networks*, 1999, 10(6): 1465-1473.
8. Shi Zhongzhi. *Knowledge Discovery (in Chinese)*. Beijing: Tsinghua University Press, 2002
9. J. Chan, I. Koprinska, J. Poon. Co-training on Textual Documents with a Single Natural Feature. Set Proceedings of the 9th Australasian Document Computing Symposium. Melbourne, Australia: ADSC, 2004: 47-54.
10. I. Cohen, F. G. Cozman, N. Sebe, M. C. Cirelo, T. S. Huang. Semi-supervised Learning of Classifiers: Theory, Algorithms and Their Application to Human-Computer Interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, 26(12): 1553-1567.
11. D. J. Newman, S. Hettich, C. L. Blake, C. J. Merz. UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science.

The Practical Method of Fractal Dimensionality Reduction Based on Z-Ordering Technique*

Guanghai Yan^{1,2}, Zhanhuai Li¹, and Liu Yuan¹

¹ Dept. Computer Science & Software NorthWestern Polytechnical University, Xian 710072, P.R. China

{yangh, yuanl}@mail.nwpu.edu.cn, lizhh@nwpu.edu.cn

² Key Laboratory of Opto-Electronic Technology and Intelligent Control (Lanzhou Jiaotong University), Ministry of Education, Lanzhou 730070, P.R. China

Abstract. Feature selection, the process of selecting a feature subset from the original feature set, plays an important role in a wide variety of contexts such as data mining, machine learning, and pattern recognition. Recently, fractal dimension has been exploited to reduce the dimensionality of the data space. FDR(Fractal Dimensionality Reduction) is one of the most famous fractal dimension based feature selection algorithm proposed by Traina in 2000. However, it is inefficient in the high dimensional data space for multiple scanning the dataset. Take advantage of the Z-ordering technique, this paper proposed an optimized FDR, ZBFDR(Z-ordering Based FDR), which can select the feature subset through scanning the dataset once except for preprocessing. The experimental results show that ZBFDR algorithm achieves better performance.

1 Introduction

The volume of information gathered by digital systems has grown not only in the amount of data items but also in the number and complexity of attributes. This happens mostly due to the intrinsic high dimensional nature of the data, and the need of the recent advanced application fields such as data mining, text matching[1], time series analysis[2], Gene Expression Patterns analysis and DNA sequences analysis[3][4]. The high dimensional data and the low dimensional data have differences in many aspects. Moreover it causes the so-called 'curse of dimensionality'[5]. Thus, the dimensionality reduction has become important techniques for automated pattern recognition, exploratory data analysis, and data mining.

Generally there are two ways of dimension reduction: Feature selection and feature extraction. The feature extraction method generates a new low dimensional feature space from the original feature space. The new feature space is

* This work is sponsored by the National Natural Science Foundation of China (No.60573096), and the Opening Foundation of the Key Laboratory of Opto-Electronic Technology and Intelligent Control (Lanzhou Jiaotong University), Ministry of Education, China, Grant No. K04116.

artificially generated (e.g. generated by some machine learning algorithms) and is difficult for human understanding. The feature selection method reduces the dimensions of the old feature space by carefully selecting the features subset as the new feature space. In contrast to the feature extraction, it does not do rotation or transformation of the features, thus leading to easy interpretation of the resulting features.

This paper emphasizes on the fractal dimension based feature selection method, investigates the current method and proposes the optimized algorithm. The remainder of the paper is structured as follows. In the next section, we present a brief survey on the related techniques. Section 3 introduces the concepts needed to understand the proposed method. Section 4 presents the optimized Z-ordering based fractal dimension algorithm. Section 5 discusses the experiments and the comparison of ZBFDR with OptFDR and FDR. Section 6 gives the conclusions of this paper and indicates our future work trend.

2 Related Work

A large number of attributes selection methods have been studied in the fields of pattern recognition and machine learning, including genetic algorithms[6]; sequential feature selection algorithms such as forwards, backwards and bidirectional sequential searches; and feature weighting[7][8]. A recent survey on attribute selection using machine learning techniques is presented in[9] and a recent review on dimension estimation of data space can be founded in[10]. A common disadvantage in attribute selection method so far is the exponential growth of computing time required. Additionally, these approaches are highly sensitive to both the number of irrelevant or redundant features present in the dataset, and to the size of the dataset, provided avoiding the use of samples[11].

Fractal theory, initialized in the 70's of the last century, is a useful tool for the analysis of spatial access methods [12], multidimensional indexing[13], and the analysis of web workloads and internet traffics[14].

Traina firstly suggested using fractal dimension for feature selection. The first famous one is the fractal dimension based feature selection algorithm FDR proposed by Traina in 2000 [15]. The main idea of FDR is to use the fractal dimension of the dataset, and sequentially to drop the attributes which contribute minimally to the fractal dimension until the terminal condition holds. In contrast to other methods, the FDR does not rotate the attribute, leading to easy interpretation of the resulting attributes. But the FDR is usually intractable for the large dataset in practice for its $\frac{K*(2*E-K+1)}{2}$ (K is the number of attributes to drop, E is the total number of attributes of the dataset) scanning of the dataset. In order to overcome the performance bottleneck of the FDR in 2004 BaoYubin et al. proposed the OptFDR, which scans the dataset only once and adjusts the FD-tree dynamically to calculate the fractal dimension of the dataset [16]. But the adjust process of the FD-tree is complicated and the computational complexity is high.

3 Preliminaries and the FDR Algorithm

If a dataset has self-similarity in an observation range, that is, the partial distribution of the dataset has the similar structure or feature as its whole distribution, and then the dataset is said as fractal. Next, we give some related concepts.

3.1 Preliminaries

The dimensionality of the Euclid space where the data points of a dataset exist is called the embedding dimension of the dataset. In other words, it is the number of attributes of the dataset. The intrinsic dimension of a dataset is the dimension of the spatial object represented by the dataset, regardless of the space where it is embedded. Whenever there is a correlation between two or more features, the intrinsic dimensionality of the dataset is reduced accordingly. So if we know its intrinsic dimension, it is possible for us to decide how many features are in fact required to characterize a dataset. Due to its computational simplicity, the correlation fractal dimension is successfully used to estimate the intrinsic dimension of the dataset in real application[17].

Definition 1 (Correlation Fractal Dimension). Suppose a dataset that has the self-similarity property in the range of scales $[r_1, r_2]$, its Correlation Fractal Dimension D_2 for this range is measured as:

$$D_2 = \frac{\log \sum_i C_{r,i}^2}{\log(r)} , \quad r \in [r_1, r_2] \tag{1}$$

where r is the edge length of the *Cell* (abbr. of the hyper-rectangle) which covering the vector space, and $C_{r,i}$ is the count of points in the *ith Cell*.

Definition 2 (Partial fractal dimension). Suppose a dataset A with E attributes, this measurement is obtained through the calculation of the correlation fractal dimension of this dataset excluding one or more attributes from the dataset [15].

3.2 FDR Algorithm

According the definition of the fractal dimension details of calculating the fractal dimension follow. Firstly, construct the E-dimensional cell grid that embedded in the E-dimensional data space. Focusing the *ith r Cell*(r is the edge length of the *Cell*), let $C_{r,i}$ be the count ('occupancies') of points in the *ith Cell*. Then, compute the value $\sum_i C_{r,i}^2$ for all *Cells* with the same edge length r . The fractal dimension is the derivative of $\log \sum_i C_{r,i}^2$ with respect to the logarithm of the edge length r . In [15]a data structure called FD-tree is built in memory in order to record the number of points that falls in each *cell* with respect to different *celledge* length r . The performance bottle neck of FDR is the $\frac{K*(2*E-K+1)}{2}$

(K is the number of attributes to drop, E is the total number of attributes of the dataset) scanning the dataset to initialize the FD-tree. Based on the Z-ordering technique, ZBFDR, proposed in this paper, which generally outperforms FDR for scanning the dataset only once and consumes lower space than OptFDR.

4 Z-Ordering Based FDR

4.1 Integer-Coded Z-Ordering Index

Z-ordering[18] is based on the Peano curve. Starting from the (fixed) universe containing the data object, space is split recursively into two subspaces of equal size by $(E - 1)$ -dimensional hyperplanes. The subdivision continues until some given level of accuracy has been reached. The data object is thus represented by a set of cells, called Peano regions or Z-regions. Each such Peano region can be represented by a unique bit string, called Peano code, Z-value, or DZ-expression. Using those bit strings, the cells can then be stored in a standard one-dimensional index, such as a B-tree. Z-ordering can not afford the high dimensional space and the overmany multilevel subdivision because of the length of its bit string. Here we adapt the variation: Integer-coded Z-ordering. Without lose of generality, suppose features are arranged in a fixed sequence, and the integers serve as the correspondent coordinates. In figure 1, take two dimensional space as example, the coordinate sequence is: first X axes and then Y axes, the integer sequence in each *Cell* is the coordinate sequence of this *Cell*, and the integer sequence at the cross point is the coordinate sequence of the upper level *Cell*.

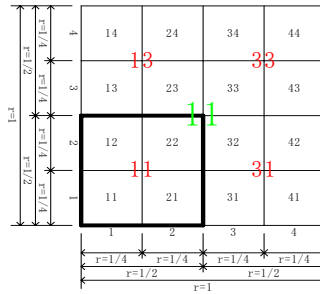


Fig. 1. A 2-dimensional integer Z-ordering code

4.2 Evaluation of the Fractal Dimension

The evaluation of the fractal dimension of the dataset is the foundation of the algorithm. In figure 1, for each $Cell_i, i = 1, 2, \dots, M$ (where M is the number of Cells in the lowest level which at least contains one data point), the Z-ordering coordinate is $Coor_i = (Z_{i_1}, Z_{i_2}, \dots, Z_{i_E}), Z_{i_j} \in 2^k | k = 0, 1, 2, \dots, Maxlevel$ (where $Maxlevel$ is the maximum level number). In order to evaluate the fractal dimension of the dataset we must count the number of points in each *Cell* at

every level. FDR calculates the fractal dimension by constructing the FD-tree. In ZBFDR, we suggest only to construct the lowest *Cell* queue, dynamically map the lower *Cell* queue into the upper *Cell* queue, and evaluate $\sum_i C_{r,i}^2$ of each level queue respectively. This solution consumes lower space than FDR and has equivalent time complexity to FDR simultaneously (e.g. FDR constructs the FD-tree from the root node to the leaf node and keep the whole structure in the main memory during the evaluate process, ZBFDR constructs the FD-tree inversely and only keep the leaf node in the main memory).

In figure 1, we can simply count the points contained by every minimum *r Cell*. For the bigger *r Cell* in which the count of points can be calculated with the sum of the points in the $4(2^2)$, the number is 2^E in E dimensional space) minimum *r Cell* contained by the upper *r Cell*. This process continued until the maximal *r Cell* is processed. For instance, the bigger *Cell* which coordinate is enclosed by the bold line has the same point number equivalent to the sum of the point number of the four minimum *r Cell* (which coordinate is: (1, 1), (1, 2), (2, 2) and (2, 1)). This work can be done through changing the coordinates of the four minimum *r Cell* into the coordinate (1, 1) and counting the point number of the *cells* with the coordinate (1, 1).

Generally, we can adjust the E dimensional coordinate of each *Cell* according to equation 2. Where $Coordinate_{old}$ is the coordinate before adjust, $Coordinate_{new}$ is the resulting coordinate after adjust, j is the level number for merge, $j = 1$ means to map the lowest level *Cell* queue into its upper (the second lowest) level *Cell* queue, $j = 2$ means to map the second lowest level *Cell* queue into its upper (the third lowest) level *Cell* queue, and so on. Repeat this process until we get one *cell* which contains the total data points.

We can get the $\sum_i C_{r,i}^2$ through merging *Cells* which has the identical coordinate and summing the point number of each *Cell*. Repeat the preceding process we can get a series of $(\log \sum_i C_{r,i}^2, \log(r))$. Thus, through plotting $\log \sum_i C_{r,i}^2$ versus $\log(r)$, and calculating the slope of the scaling range of the resulting line, we can obtain the correlation fractal dimension D_2 of the dataset.

$$\begin{cases} Coordinate_{new} = Coordinate_{old}, & (\frac{Coordinate_{old}-1}{2^{j-1}}) MOD 2 = 0 \\ Coordinate_{new} = Coordinate_{old} - 2^{j-1}, & (\frac{Coordinate_{old}-1}{2^{j-1}}) MOD 2 = 1 \end{cases} \quad (2)$$

4.3 Backward Eliminating Attribute

It is important to point out that the elimination of the selected dimension does not mean deleting the data points. So we can view the elimination of the one selected dimension as the projection from E dimensional space to $E - 1$ dimensional space. Suppose the two same level cells $Cell_i$ and $Cell_j$ have coordinate sequence $Coor_{iE} = (Z_{i1}, Z_{i2}, \dots, Z_{ii-1}, Z_{ii}, Z_{ii+1}, \dots, Z_{iE})$ and $Coor_{jE} = (Z_{j1}, Z_{j2}, \dots, Z_{ji-1}, Z_{ji}, Z_{ji+1}, \dots, Z_{jE})$ respectively. If the i th dimension is eliminated from the E dimensional space, it is equivalent to project the E dimensional space onto the $E - 1$ dimensional spaces. The coordinate sequence after projecting as follows: $Coor_{iE-1} = (Z_{i1}, Z_{i2}, \dots, Z_{ii-1}, Z_{ii+1}, \dots, Z_{iE})$ and

$Coor_{j_{E-1}} = (Z_{j_1}, Z_{j_2}, \dots, Z_{j_{i-1}}, Z_{j_{i+1}}, \dots, Z_{j_E})$. The condition to merge the $Cell_i$ and $Cell_j$ after eliminating the i th dimension is: $Coor_{i_{E-1}} = Coor_{j_{E-1}}$. In fact if the i th dimension coordinate value of $Coor_{i_E}$ and $Coor_{j_E}$ is marked zero (that means $Coor_{i_E}$ is identical with $Coor_{j_E}$), we can merge the $Cell_i$ and $Cell_j$ directly after eliminating the dimension. The coordinate value of the new derived cell is $Coor_{i_E}$ or $Coor_{j_E}$ with the i th dimension coordinate value is zero. The partial fractal dimension of the $E - 1$ dimensional dataset can be evaluated by the process described in the section 4.2.

4.4 ZBFDR Algorithm

The ZBFDR algorithm is omitted for space limit. It scans the dataset only once to initialize the lowest $Cell$ queue. Each time we eliminate the dimension which causes the minimum effect on the fractal dimension of the current dataset. In other words, suppose FD is the fractal dimension of the current dataset and $PFD_i, i \in (1, 2, \dots, E)$ is the partial fractal dimension of the dataset after eliminated the i th dimension, the dimension for eliminating is the i th dimension which leads the minimum $(FD - PFD_i)$.

5 Experiments and Evaluation

The performance experiments on fractal dimension calculation and backward attribute reduction are made for evaluating FDR algorithm, OptFDR algorithm

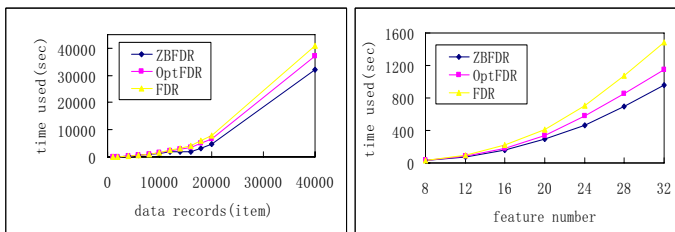


Fig. 2. The performance comparison result on synthetic and real dataset (The left column presents feature selection performance comparison on 8 dimensional stochastic dataset (where the two dimensions randomly generated and the others are nonlinear functions of the two former, the $Cell$ level is 32 and the data item number varies from 1000 to 40000) and the right column presents the experimental result on BRITAIN dataset (where the $Cell$ level is 32))

and ZBFDR algorithm using real dataset and synthetic dataset with fractal characteristics. The hardware environment includes Intel Pentium IV 1.7GHz CPU, 512MB RAM, 40GB hard disk, and the software environment includes Windows 2003 and Delphi 7.

The experimental results list in figure 2. The feature number of the synthetic dataset varies from 8 to 32, and the point number varies from 1000 to 40000.

The real dataset is BRITAIN dataset (A classic fractal dataset which include the datum of the coast line of England)[17], the data item number is 1292 and the attribute number varies from 8 to 32.

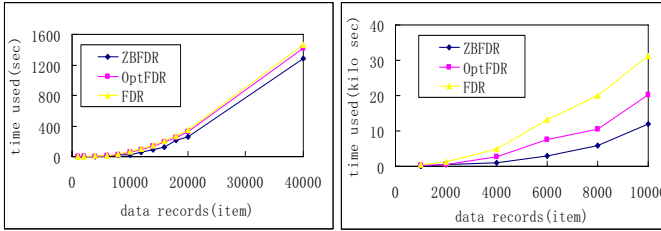


Fig. 3. Experimental result of FD calculation and feature selection (The left column presents the calculation of FD on 32 dimensional linear dataset and the right column presents the performance comparison of feature selection on the same dataset(data records number is 10000))

Considering the calculation of fractal dimension is the key factor that affects the performance of feature selection, figure 3 presents the calculation of fractal dimension on 32 dimensional dataset and the performance comparison of feature selection on the same dataset.

6 Conclusion

The ZBFDR algorithm is proposed, which can complete the feature selection process through scanning the dataset once except for preprocessing. The experimental results show that ZBFDR algorithm outperforms FDR algorithm and OptFDR algorithm. In the future, our research work will concentrate on the efficient algorithm for evaluating the fractal dimension, the popularization of the algorithm on non-numerical dataset, and the combination with other feature selection algorithms.

References

1. R Baeza-Yates, G Navarro. Block-addressing indices for approximate text retrieval. In: Forouzan Golshani, Kia Makki. (Eds.): Proc of the 6th Int'l Conf on Information and Knowledge Management. New York: ACM Press (1997) 1-8
2. R Agrawal, C Faloutsos, A Swami. Efficient similarity search in sequence databases. In: David B Lomet. (Eds.): Proc of the 4th Int'l Conf Foundations of Data Organization and Algorithms. Berlin : Springer-Verlag (1993) 69-84
3. Daxin Jiang, Chun Tang, Aidong Zhang. Cluster Analysis for Gene Expression Data: A Survey. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING Vol. 16, No. 11. (2004) 1370-1386
4. M.D. Schena, R. Shalon, R. Davis, P. Brown. Quantitative Monitoring of Gene Expression Patterns with a Complementary DNA Microarray. Science, vol. 270. (1995) 467-470

5. R. Bellman. Adaptive Control Process: A Guide Tour. Princeton University Press, Princeton, New Jersey (1961)
6. H. Vafaie and K. A. D. Jong. Robust Feature Selection Algorithms. In Intl. Conf. on Tools with AI, Boston, MA (1993) 356-363
7. D. W. Aha, R. L. Bankert. A Comparative Evaluation of Sequential Feature Selection Algorithms. In Artificial Intelligence and Statistics V, Springer-Verlag, New York, New York (1996) 199-206
8. M. Scherf, W. Brauer. Feature Selection by Means of a Feature Weighting Approach. Technical Report FKI-221-97, Technische Universität München, Munich (1997)
9. A. Blum, P. Langley. Selection of Relevant Features and Examples in Machine Learning. AI, vol. 97. (1997) 245-271,
10. Camastra Francesco. Data dimensionality estimation methods: a survey. Pattern Recognition, vol. 36. (2003) 2945-2954.
11. P. Langley, S. Sage. Scaling to Domains with many Irrelevant Features. In R. Greiner (eds.): Computational learning theory and natural learning systems, vol. 4, Cambridge, MA: MIT Press (1997)
12. C. Faloutsos, B. Seeger, A. J. M. Traina, C. Traina, Jr. Spatial Join Selectivity Using Power Laws. In ACM SIGMOD, Dallas, TX, (2000) 177-188
13. I. Kamel, C. Faloutsos. Hilbert R-tree: An Improved R-tree using Fractals. In 20th VLDB, Chile (1994) 500-509
14. D. Chakraborty, A. Ashir, T. Sukanuma G., Mansfield Keeni, T. K. Roy, N. Shiratori. Self-similar and fractal nature of Internet traffic. INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT, vol. 14. (2004) 119-129
15. Caetano Traina Jr., Agma Traina, et al. Fast feature selection using fractal dimension. In XV Brazilian DB Symposium, João Pessoa-PA-Brazil, (2000) 158-171
16. Yubin Bao., Ge Yu., Huanliang Sun., Daling Wang. Performance Optimization of Fractal Dimension Based Feature Selection Algorithm. In: Q. Li, G. Wang, and L. Feng. (Eds.): WAIM 2004, LNCS, Vol. 3129. Springer-Verlag Berlin Heidelberg (2004) 739-744
17. Ls Liebovitch, T. Toth. A Fast Algorithm to Determine Fractal Dimensions by Box Counting[J]. Physics Letters, Vol. 141A(8). (1989) 386-390
18. J. Orenstein, T. H. Merrett. A class of data structures for associative searching. In Proceedings of the Third ACM SIGACT- SIGMOD Symposium on Principles of Database Systems (1984) 181-190

Feature Selection for Complex Patterns

Peter Schenkel¹, Wanqing Li², and Wanquan Liu³

¹ University of Karlsruhe, Germany
p.schenkel@t-online.de

² University of Wollongong, Australia
wanqing@uow.edu.au

³ Curtin University of Technology, Australia
wanquan@cs.curtin.edu.au

Abstract. Feature selection is an important data preprocessing step in data mining and pattern recognition. Many algorithms have been proposed in the past for simple patterns that can be characterised by a single feature vector. Unfortunately, these algorithms are hardly applicable to what are referred as complex patterns that have to be described by a finite set of feature vectors. This paper addresses the problem of feature selection for the complex patterns. First, we formulated the calculation of mutual information for complex patterns based on Gaussian mixture model. A hybrid feature selection algorithm is then proposed based on the formulated mutual information calculation (filter) and Bayesian classification (wrapper). Experimental results on XM2VTS speaker recognition database have not only verified the performance of the proposed algorithm, but also demonstrated that traditional feature selection algorithms designed for simple patterns would perform poorly for complex patterns.

1 Introduction

Feature selection (FS) is an important data preprocessing step in data mining and pattern recognition. It aims to rank a set of features in the order of their discriminative power based on the classification of observed patterns or to select a minimum subset of features that can most effectively describe the patterns [11,7]. Many feature selection algorithms (FSAs) have been proposed in the past and thorough literature reviews can be found in [3,11,7]. Broadly, FSAs are divided into three categories [12,11]: filter model [14], wrapper model [9] and hybrid model [18,12].

The filter model exploits the general characteristics of the data and selects features without involving any classification. One popular technique for filter model is mutual information (MI) [2]. MI is a measure of correlation between two variables. It is often used to calculate the relevance of a feature with respect to a target class and redundancy among features. The fact that MI is independent of the chosen coordinates permits a robust estimation of the relevance and redundancy [16,1]. The wrapper model selects features by directly using classification results as a criterion for ranking the features or selecting the minimum subset of

features. Although the subset found by a wrapper algorithm is best suited to the employed classifier, wrapper model tends to be more computationally expensive than the filter algorithms [9,11]. The hybrid model is a combination of both approaches, taking the advantages of each model at different search stages [12].

To our best knowledge, most existing FSAs were designed for the problems in which patterns can be characterised by a single feature vector. We refer this type of patterns as simple patterns since one single feature vector carries enough information for its mapping onto a target class. For instance, Lymphoma (LYM) [5] is one of the widely used simple pattern data sets for FS study. LYM has 96 samples classified into nine classes. However, in applications that involve spatial and temporal data, such as recognition of human voice, face and gait [17], patterns can not be characterized by a single feature vector, instead, they have to be defined using a set of feature vectors. For instance, in speaker recognition, a two second voice sample is often needed in order to identify or verify a person. The voice signal is segmented into 20 millisecond frames with 10 millisecond overlap between consecutive frames. From each frame, Mel Filter-bank Cepstral Coefficients (MFCC) and delta MFCC [8] are calculated as features to characterise the frames. Depending on the number of filters used in the calculation, a frame is usually described by a 12 MFCCs and 12 delta MFCCs, which forms a 24 dimensional feature vectors. A speaker is identified or verified by about 400 feature vectors calculated from the voice signal. Although the 400 feature vectors may have redundant information about the speaker, none of the single feature vector is sufficient to identify or verify the speaker!

As more and more features are derived and proposed for temporal and spatial pattern recognition, FSAs for these complex patterns are highly expected. This paper concerns the problem of ranking features for complex patterns. A hybrid algorithm is proposed based on the mutual information (MI) for complex patterns (filter) and Bayesian classification (wrapper). The MI is used to measure the relevance of the features to the target classes and the Bayesian classification is used to deal with ambiguous sample data. The calculation of mutual information for complex patterns is based on Gaussian mixture model. Experimental results on XM2VTS [13] speaker recognition database have not only verified the performance of the proposed algorithm, but also demonstrated that traditional feature selection algorithms designed for simple patterns would perform poorly for complex patterns.

2 MI Based FSAs for Simple Patterns

An MI based FSA (MIFSA) in general involves four steps [12]: feature subset generation, subset evaluation, stopping criteria and verification. In the case that the FSA outputs the ranking of a feature set, sequential forward strategy [10,16,1] is often adopted in the subset generation where one feature at a time is selected, then in evaluation step, the relevance between the selected feature and the target classes is measured using MI. All features will be ranked according to their

relevance to the targeted classes. Finally such selected ranking will be verified by single feature based classification.

Let $F = \{f_1, f_2, \dots, f_m\}$ be the feature set containing m features that characterise the patterns to be classified. There are k classes denoted by $C = \{c_1, c_2, \dots, c_k\}$. Given n training samples $x = \{x_1, x_2, \dots, x_n\}$ and their corresponding target class labels $y = \{l_1, l_2, \dots, l_n\}$, where each sample $x_i, i = 1, 2, \dots, n$ is described by a feature vector denoted as $x_i = (x_i^1, x_i^2, \dots, x_i^m)$ and $l_i \in C, i = 1, 2, \dots, n$, the FS is to rank the feature set $f_j, j = 1, 2, \dots, m$ based on the MI, $I(C, f_j)$, between each feature component, $f_j, i = 1, 2, \dots, m$ and the set of classes C .

The definition of MI is given as [2]

$$I(C; F_j) = \sum_{c \in C} \int p(c, x^j) \log \frac{p(c, x^j)}{p(c)p(x^j)} dx^j \tag{1}$$

where F_j is a random variable representing the j 'th feature and x^j is an observation of the feature F_j .

If the mutual information between two random variables is large (small), it means two variables are closely (not closely) related. The mutual information is often calculated through the entropy

$$I(C; F_j) = H(C) - H(C|F_j) \tag{2}$$

where $H(C)$ is the entropy of c and $H(C|F_j)$ is the conditional entropy of C given F_j .

$$H(C) = - \sum_{c \in C} P(c) \log P(c) \tag{3}$$

$$H(C|F_j) = - \sum_{c \in C} \int p(c, x^j) \log p(c|x^j) dx^j \tag{4}$$

where $P(c)$ is the probability of class C , $p(c|x^j)$ is the conditional probability density function (pdf) and $p(c, x^j)$ is the joint pdf.

It can be seen that MI calculation requires *pdfs* and their estimation will certainly influence the performance of the FSA. For discrete data, pdfs are easily to obtain. Continuous data is often discretized [16,1,10,6] or its pdf is modelled and estimated using parameterised functions [14,15,4], such as finite mixture models [15,4] and Parzen Window [14].

If the FSA is to select a compact subset feature out of F , the redundancy among features is needed in order to avoid highly correlated features being in the subset at same time. MI between two feature components has also been proved to be a good measurement of the redundancy [1,10,16]. Peng [16] recently proposed a min-Redundancy-Max-Relevance FS scheme (mRMR) that combines relevance and redundancy in order to find a significant and compact feature set. The mRMR measures both relevance and redundancy using MI and adopts incremental search strategy: each iteration maximizes relevance and minimizes redundancy for each newly added feature:

$$\max_{f \in (S - S_t)} \left(I(f; C) - \frac{1}{t-1} \sum_{s \in S_{t-1}} I(f; s) \right) \tag{5}$$

where S is the full feature set and S_t is the feature set selected in the t th iteration.

3 FS for Complex Patterns

In this section, we present an FSA to rank features for complex patterns. In particular, we propose an approach to calculating the MI between a feature component and complex pattern classes by using Gaussian mixture model.

Let $z = \{x_1, x_2, \dots, x_n\}$ be a complex pattern that is described by n simple patterns, each simple pattern is characterised by a feature vector as described in section 2. We assume that $n \geq \aleph$, where \aleph is an integer representing the minimum number of simple patterns or feature vectors. For instance, the minimum duration of voice to recognize a speaker is about 2 seconds. If $n < \aleph$, it is not possible to predict which class z belongs to. Our objective is to rank the features in F according to their power to discriminate z instead of x . In other word, the FSA has to be based on the relevance between the feature components and the targeted classes of z , NOT the targeted class of x . For, F_j , the j 'th component of F , we propose to measure the relevance using MI between C and F_j calculated with respect to z .

$$I(C; F_j) = \sum_{c \in C} \int p(c, z^j) \log \frac{p(c, z^j)}{p(c)p(z^j)} dz^j \tag{6}$$

where z^j is a set of observations of the feature F_j with respect to a complex pattern. Notice the difference of the MI with respect to x in Eq.(1) for simple patterns and the MI with respect to z in Eq. 6) for complex patterns.

Following the relationship between MI and entropy, we have

$$I(C; F_j) = H(C) - H(C|F_j) \tag{7}$$

where $H(C)$ is the entropy of C and $H(C|F_j)$ is the conditional entropy of C given F_j in the context of complex patterns.

$H(C)$ is easy to calculate using Eq.(3) and $H(C|F_j)$ is to be estimated from a set of training samples.

Let $\{z_1, z_2, \dots, z_N\}$ be the N training samples and their corresponding target class labels are $y = \{l_1, l_2, \dots, l_N\}$, where, $l_i \in c$. Sample z_i consists of n_i feature vectors, i.e. $z_i = \{x_1^i, x_2^i, \dots, x_{n_i}^i\}$. Then

$$\begin{aligned} H(C|F_j) &= \sum_{i=1}^N P(z_i^j) \sum_{c=1}^k H(c|z_i^j) \\ &= - \sum_{i=1}^N P(z_i^j) \sum_{c=1}^k P(c|z_i^j) \log P(c|z_i^j) \end{aligned} \tag{8}$$

where $P(\cdot)$ means mass probability.

Assuming that the probability of all training samples are equal, i.e. $P(z_i^j) = \frac{1}{N}$, Eq.(8) becomes

$$H(C|F_j) = - \sum_{i=1}^N \frac{1}{N} \sum_{c=1}^k P(c|z_i^j) \log P(c|z_i^j) \tag{9}$$

In Eq.(9), $P(c|z_i^j)$ is the probability of class c given the j 'th feature of i 'th sample, which is not easy to compute directly.

Using the concept of Parzen window for the estimation of pdf, let's assume that the conditional pdf $p(z^j|c)$ is known. Then according to Bayes' rule, the $P(c|z^j)$ can be estimated by

$$P(c|z^j) = \frac{p(c)p(z^j|c)}{\sum_{k=1}^k p(k)p(z^j|k)} \tag{10}$$

In traditional parzen window estimation of pdf where each training example is a simple pattern x , Gaussian window functions are often used for each example. In the case of complex pattern, z^j , we propose to use Gaussian mixture model to estimate $p(z^j|c)$ from the samples of class c .

3.1 Penalty Function for Noisy Samples

As $p(z^j|c)$ is represented as a finite Gaussian mixture model and the model is fitted to a set of training samples, the quality of the training samples is an important factor on the estimation of $H(C|F_j)$. For instance, the quality of voice signal is subject to the environmental noise and capturing devices. We have to consider the case that a training sample might not fit perfectly into the model. To deal with noisy or ambiguous samples, we introduce a penalty function t with respect to the number of false classified samples.

$$I(C; F_j) = H(C) - t \cdot H(C|F_j) \tag{11}$$

where t is the penalty function. Though it can be a non-linear function in relation to the number of false classified samples, we used a linear function

$$t(p) = 1 + \left(\frac{p}{n}\right) \tag{12}$$

where p is the number of false classified samples and n the total number of samples. All false classified samples are left out for MI calculation and increase t , as the number of false classified samples is an indicator for the relevance as well, which is extensively used in wrapper FS-algorithms. Therefore, the final relevance calculation consists of both feature's quality and feature's significance.

4 Experimental Results

4.1 Dataset

Voice data of ten (10) speakers from XM2VTS [13] database were used to verify the proposed algorithm. XM2VTS is a database for speaker recognition. It

contains speech samples of different speakers uttering different sequences (digits and sentences). For each speaker, there are 4 sessions, each session containing 6 samples, where 4 samples are spoken digits and 2 samples are sentences. Each sample has about 5-10 seconds of speech. Therefore, each speaker as a target class has totally 24 samples. In addition, the speakers' genders are random.

Silence was removed from the beginning and end of each speech sample before the feature MFCC (Mel frequency cepstral coefficients) [8] and SSC (spectral subband centroids) [8] were extracted. The feature set is composed of 12 MFCCs, 12 delta MFCCs, 10 SSCs, 10 delta SSCs and the energy of the signal. Features were extracted from 20ms time window with 10 ms overlap. This results a 45 dimensional feature vector for every 10ms. A sample will have about 500 to 1000 feature vectors.

4.2 $H(C; F_j)$ Calculation and Ranking Verification

$H(C; F_j)$ was calculated for ranking the features by fitting the samples to a Gaussian mixture model as an estimation of $p(z^j|c)$, where z^j is a set of observations of the feature F_j with respect to a complex pattern (i.e. a speaker) c . To demonstrate the correlation between the quality of the ranking and the number of Gaussians used, we fitted the GMMs with 1, 2, 10, 20, 50, 150 and 500 Gaussians.

To verify the ranking, Bayes classifier was employed using single feature by fitting GMM with larger number of Gaussians as suggested in [17]. The misclassification errors were used to indicate how good the ranking is consistent with the classification.

4.3 Results

Figure 1 shows the single feature based classification errors with respect to the rankings obtained by fitting different number of Gaussians to approximate the conditional pdf $p(z^j|c)$ for the calculation of $H(C; F_j)$. The horizontal axis represents the ranked features in a descending order of the relevance and the vertical axis represents the classification errors when the corresponding features were used for the classification. Therefore, if the ranking is consistent with classification, classification errors would monotonically increase from left to right in the graph as the relevance decreases. The trend of the classification errors against the ranking in each graph is indicated by the smooth curve.

It can be seen that the proposed FSA ranked the features very well in the cases of 500 and 150 Gaussians (Figure 1 (a) & (b)). Though there is slight degradation when the number of Gaussians decreases, the ranking obtained using two (even one) Gaussians is still reasonably good (Figure 1 (e)). In fact, we observed that the variation of the ranking was small with respect to the number of Gaussian used to approximate the $p(z^j|c)$. This demonstrates that the proposed FSA is stable and tolerant to the approximation accuracy of $p(z^j|c)$. It also shows that the proposed MI calculation well captures the relevance of the complex patterns to the feature component.

Figure 1(f) is the a feature ranking from the same training data through discretising the feature vectors and employing a simple pattern (i.e. x , not z) based MI calculation. The results did not show any correlation between the ranking order of the features and the classification errors. It is obvious that the traditional FSA for simple pattern worked poorly in this case.

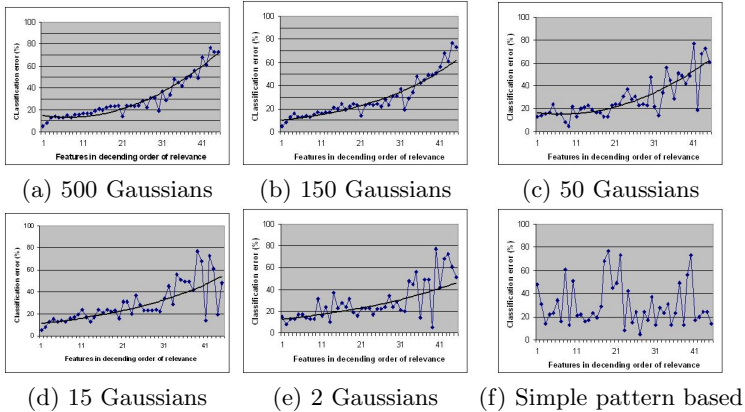


Fig. 1. Plots of the feature ranking against classification errors. The horizontal axis represents the ranked features in the descending order of the relevance and the vertical axis represents the classification errors when the corresponding features were used for the classification. The smooth curve in each graph indicate the trends of the classification errors against the ranking and the number of Gaussians indicated in graph (a)-(e) is the Gaussians used to estimate the conditional pdf $p(z^j|c)$. (f) is simple pattern based ranking when the features is discretised.

5 Conclusions

In this paper, we introduced the concepts of simple patterns and complex patterns and identified the need for FSAs for complex patterns. A hybrid FSA is proposed for complex patterns with a focus on continuous features and discrete target classes. The proposed FSA divides feature relevance into a quality(wrapper) part and a significance(MI) part, by using a light wrapper to select valid samples for MI calculation. Application of the algorithm to speaker recognition has verified its performance.

Acknowledgment

Peter Schenkel would thank Telecommunications and Information Technology Research Institute (TITR) of University of Wollongong (UOW) for the summer scholarship. Wanqing Li's work is supported by the Research Council of UOW (URC 2005).

References

1. R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. on Neural Networks*, 5(4):537–550, 1994.
2. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.
3. M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1:131–156, 1997.
4. T. Eriksson, S. Kim, H.-G. Kang, and C. Lee. An information theoretic perspective on feature selection in speaker recognition. *IEEE Signal Processing Letters*, 12(7):500–503, 2005.
5. A. A. Alzadeh et al. Distinct types of diffuse large b-cell lymphoma identified by gen expression profiling. *Nature*, 403:503–511, 2000.
6. A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review*, 33(2):1134–1140, 1986.
7. I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
8. X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing*. Prentice Hall, 2001.
9. R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):23-324.1997, 97(1-2):23–34, 1997.
10. N. Kwak and C.-H. Choi. Input feature selection for classification problems. *IEEE Trans. on Neural Networks*, 13(1):143–159, 2002.
11. H. Liu. Evolving feature selection. *IEEE Intelligent Systems*, pages 64–76, Nov/Dec 2005.
12. H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. on Knowledge and Data Engineering*, 17(4):491–502, 2005.
13. J. Luttin. Evaluation protocol for the xm2fdb database (lausanne protocol). Technical Report Communication 98-05, IDIAP, Martigny, Switzerland, 1998.
14. Kwak N and C.-H. Choi. Input feature selection by mutual information based on parzen window. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(12):1667–1671, 2002.
15. M. Nilsson, H. Gusafsson, S. V. Andersen, and W. B. Kleijn. Gaussian mixture model based mutual information estimation between frequency bands in speech. In *ICASSP*, volume 1, pages 525–528, 2002.
16. H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.
17. D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:19–41, 2000.
18. A. Tsymbal and M. Pechenizkiy and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information fusion*, 6:83–98, 2005.

Naïve Bayesian Tree Pruning by Local Accuracy Estimation

Zhipeng Xie

Department of Computing and Information Technology
Fudan University, Shanghai, P.R. China, 200433
xiezp@fudan.edu.cn

Abstract. Naïve Bayesian Tree is a high-accuracy classification method by combining decision tree and naïve Bayes together. It uses averaged global accuracy as the measurement of goodness in the induction process of the tree structure, and chooses the local classifier that is most specific for the target instance to make the decision. This paper mainly introduces a pruning strategy based on local accuracy estimation. Instead of directly using the most specific local classifier (mostly the classifier in a leaf node) to making classification in NBTree, our pruning strategy uses the measurement of local accuracy to guide the selection of local classifier for decision. Experimental results manifest that this pruning strategy is effective, especially for the NBTree with relatively more nodes.

1 Introduction

Ensemble has been popular for many years. Traditional ensemble techniques either use multiple classification methods on the same training set, or inject disturbance into the data set, in order to get a diverse set of member classifiers. Recently, a new idea and its several materialized algorithms have emerged. The main thrust is to learn multiple local classifiers in various subspaces of the global instance space, which has already been proven effective and efficient by several algorithms designed by some researchers. Such algorithms include: naïve Bayesian tree (NBTree) [4], lazy Bayesian rule (LBR) [8], selective neighborhood based naïve Bayes (SNNB) [6, 7], and concept lattice based naïve Bayes (CLNB) [5].

Once one ensemble (or a set of classifiers) is constructed, the simplest way is to choose the best classifier and to use it to make classification on behalf of the whole ensemble. Normally, the best classifier refers to the classifier with the highest estimated accuracy. How to estimate the accuracy of a classifier? Some existing techniques, such as k-fold cross validation and leave-one-out, can serve this job. The accuracy estimated by such techniques is a global accuracy averaged over the whole training set (or the whole instance space). Howbeit, a classifier may perform quite differently in different regions (or subspaces) of the instance space. Therefore, it will be better if we could estimate the local accuracy of a classifier in a local region. That is the rationale driving the work in this paper.

Estimated global accuracy is adopted by NBTree to direct the whole induction process. NBTree also directly uses the local classifier of the most specific subspace to which the target instance belongs to make the decision, without considering whether it really has the best performance at the point of the target instance. To take this into consideration is the motivation of this work. In this paper, instead of directly using the most specific classifier, local accuracy estimation is adopted as the measurement of goodness in choosing the most suitable classifier for the target instance. As a result, for some target instances, we will choose the non-terminal nodes and use the associated local classifier to make decision. Similarly, classical decision tree (like C4.5) also classify a target instance as the majority class of the leaf node that is the most specific node containing the target instance, and post-pruning the decision tree means some leaf nodes would be pruned and thus some target instances may be classified as the majority class of non-leaf nodes. That is why we call our approach a pruning technique.

This paper is organized as follows: section 2 analyzes NBTree algorithm in detail and points out its weakness. To make up this weakness, in section 3, a post-pruning strategy is developed and used to substitute the naïve decision process of original NBTree. Experimental results goes in Section 4, and manifests the effectiveness of this pruning strategy. At last, Section 5 points out possible future work.

2 Naïve Bayesian Tree and Its Analysis

2.1 Basic Notations and Accuracy Estimation

Consider a domain where instances are represented as instantiations of a vector $A=\{a_1, a_2, \dots, a_m\}$ of m nominal variables. Here, each instance x takes a value $a_j(x)$ from $domain(a_j)$ on each $a_j \in A$. Further, an example (or instance) x is also described by a class label $c(x)$ from $domain(c)$. Let $D=\{(x_i, c(x_i)) \mid 1 \leq i \leq n\}$ denote the training dataset of size n . The task of classification is to construct a model (or classifier) from the training set D , which is a function that assigns a class label to a new unlabelled example. What is used to accomplish this task is called a classification method (or classification algorithm). Naïve Bayes (NB) is a typical one, and the classifier constructed from the training set D using the naïve Bayes method is denoted by $NB(D)$. This classifier, also represented by $NB(x, D)$, assigns a value from $domain(c)$ to an example x .

The existing classification algorithms are too numerous to be counted. For a domain, we have to determine which one is more accurate, and thus to use it. There are a variety of techniques to estimate the accuracy of a classifier. Leave-One-Out is a popular one of them.

The accuracy of a classifier $NB(D_1)$ trained on the data set D_1 can be measured with the leave-out-one technique, as follows:

$$ACC_G(NB(D_1))=|\{x \in D_1 \mid NB(x, D_1 - \{x\})=c(x)\}|/|D_1| \tag{1}$$

The accuracy estimated by the above formula is called the estimated global accuracy. It is the accuracy globally averaged on the training set of the classifier.

Similarly, for a subset $D_2 \subset D_1$, the local accuracy of $NB(D_1)$ on D_2 is

$$ACC_L(NB(D_1), D_2)=|\{x \in D_2 \mid NB(x, D_1 - \{x\})=c(x)\}|/|D_2| \tag{2}$$

To check whether $\mathbf{NB}(x, D_1 - \{x\}) = c(x)$ holds, we must first leave the example x out of D_1 to get $\mathbf{NB}(D_1 - \{x\})$ in time $O(m)$, then use $\mathbf{NB}(D_1 - \{x\})$ to predict the class label of x in time $O(m)$, and finally put x back to restore $\mathbf{NB}(D_1)$ in time $O(m)$. Thereby, the complexity of global accuracy estimation is $O(|D_1| \times m)$.

2.2 Naïve Bayesian Tree Induction

By using decision tree structure, NBTree partitions the global instance space (corresponding to the whole training dataset) into multiple local subspaces (corresponding to multiple training subsets), and constructs one local naïve Bayes classifier for each of such subspaces. This kind of partitioning is recursively conducted until the estimated accuracy cannot be improved significantly any more.

Each node N in the constructed tree represents an instance subspace, and each edge " $N_1 \rightarrow N_2$ " represents a restriction " $a=v$ ", where the attribute a is the best attribute for splitting at node N_1 and $N_2 = \{x \in N_1 | a(x) = v\}$ is a subspace of N_1 . Any instance belongs to a set of subspaces along one path started from the root in the tree. Here, the root is the most general subspace (global space). The subspaces (or nodes) along the path are monotonically decreasing with their increasing depth in the tree. For a given unlabelled instance, the local classifier corresponding to the most specific subspace on the path is selected to make the decision.

Example 1. A simple tree structure has been depicted in Fig. 1. There are totally 8 nodes. Each node is drawn as a rectangle consisting of two cells, where top cell indicates the local training subset, while the bottom cell indicates the splitting attribute used at this node.

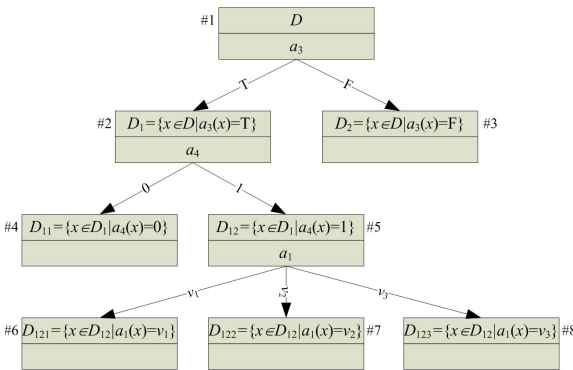


Fig. 1. A simple tree structure

A local naïve Bayes classifier, not shown in figure, can be easily trained on the local training subset at each node.

Now, if an instance x satisfies $a_3(x) = T$, $a_4(x) = 1$, and $a_1(x) = v_2$, it will belong to all the nodes on the path #1 → #2 → #5 → #7. Here, #7 is the most specific subspace that x belongs to, and thus NBTree

will use the associated local classifier $\mathbf{NB}(D_{122})$ to make the classification, with $\mathbf{NB}(x, D_{122})$ as the classification result.

However, does the local classifier corresponding to the most specific subspace make the best decision? The answer is sometimes yes, but not always. For any node N in naïve Bayesian tree, let D_N be the associated training subset. Assume that attribute a

partitions D_N into several disjoint subsets: $D_{N,a=v}=\{x \in D_N | a(x)=v\}$, $v \in domain(a)$. The utility of this attribute a at node N is calculated as:

$$Utility(D_N, a) = \sum_{v \in domain(a)} \frac{|D_{N,a=v}|}{|D_N|} \times ACC_G(NB(D_{N,a=v})) \tag{3}$$

The best attribute a_{best}

$$a_{best} = \arg \max_{a \in A} (Utility(a)) \tag{4}$$

will be chosen to fork the node N , and thus be used to partition the training subset D_N , if the relative reduction in error of this splitting is no less than 5%, that is, if $\frac{Utility(D_N, a_{best}) - ACC_G(NB(D_N))}{1 - ACC_G(NB(D_N))} \geq 0.05$.

Equation (3) is actually the weighted average accuracy of the child nodes after splitting, where the weight is proportional to the number of instances in that child node. Thus, in resulting naïve Bayesian tree, each partition will yield a higher estimated accuracy.

Clearly, the attribute selection criterion used in NBTree only guarantee the improvements of the averaged accuracy. It does not mean that each local classifier performs better than the classifier of its parent node. There are two different situations:

- (1) The estimated global accuracy of a parent node is not necessarily lower than that of each child node.
- (2) Even if a parent node were less accurate than each of its child nodes, it does not mean that it has lower estimated local accuracy on its child nodes than the estimated global accuracy of its child nodes.

The performance of NBTree could be improved if we overcome its weakness in the above two situations, which will be done in the next section. At the closing of the analysis, let us have a look at a simple example and get an intuitive feeling of these two situations.

Example 2. Let the local classifier $NB(D_N)$ associated with node N has the estimated global accuracy as 90%, where D_N is the corresponding training subset of size 200. Assume attribute a be a binary attribute with domain $\{T, F\}$. It divides D_N into two subsets, $D_{N,a=T}$ and $D_{N,a=F}$, each containing 100 instances.

One possible situation: the estimated global accuracy of $NB(D_{N,a=T})$ is 98%, while that of $NB(D_{N,a=F})$ is 88%. Here, the weighted average accuracy after splitting is 93%. The relative reduction in error is $3\%/10\%=0.30 > 0.05$. There is a great reduction in the estimated error. But the branch of “ $a=F$ ” leads to the node with even lower estimated global accuracy than the parent node.

Another situation: Even if the estimated global accuracy of $NB(D_{N,a=F})$ is raised to 92%, it still cannot guarantee that the local accuracy of $NB(D_N)$ on $D_{N,a=F}$ is lower than 92%. One possibility is that $ACC_L(NB(D_N), D_{N,a=F})=95\%$ while $ACC_L(NB(D_N), D_{N,a=T})=85\%$, where the classifier $NB(D_N)$ is preferable to the local classifier $NB(D_{N,a=F})$ when to classify a target instance in $D_{N,a=F}$.

3 Post-pruning Naïve Bayesian Tree

In the decision phase, NBTree directly assigns the most specific local classifier to the target new instance, regardless of its effect in the local neighborhood of the target instance. Let D_1 and D_2 be two instance subspaces satisfying $D_2 \subseteq D_1$ (here we call D_2 is a subspace of D_1). The assumption that NBTree took is that: the local classifier induced on D_2 should have better local performance on D_2 than the classifier learnt on a larger subspace D_1 which is more general than D_2 . It holds in many cases, but not in all cases.

In the previous section, through analysis and a simple example, it has been shown that the local classifier is possibly locally less accurate in the local area of some target instance than the preceding classifiers that are induced on more general subspaces.

The pruning of naïve Bayesian tree consists of three steps:

- (1) Identifying the deepest node that the target instance belongs to, and let it be N .
- (2) For each training example in N , computing its distance to the target instance, and then selecting the $k=30$ nearest training examples out to form the evaluation set.
- (3) For each local classifiers on the path activated by the target instance, measuring its local accuracy on the evaluation set. After that, the local classifier with the highest estimated local accuracy is chosen to make the decision.

Please note in Step 2: not all the training examples in the most specific (or deepest) node, only k nearest examples among them are selected out and used as the evaluation set. These k nearest examples also belong to the path activated by the target example. Each local classifier will be evaluated on the evaluation set to get its estimated local accuracy.

It is evident that this post-pruning technique will have larger possibility to be successful for the naïve Bayes trees with more nodes. If a naïve Bayes tree is of depth 1 (that is, the tree consists of only one root node which has the global naïve Bayes classifier as its local classifier), it will perform equally well as the naïve Bayes classification method. From the experimental results in the next section, it is shown that the naïve Bayesian trees with more nodes have got more reduction on classification error than those with fewer nodes.

4 Experimental Results

To evaluate the pruning strategy presented in this paper, we selected 22 datasets from UCI machine learning repository [1] with the constraint that each dataset should contain at least 300 examples. Table 1 lists all the datasets used and the related characteristics. Ten-fold cross validation was executed on each dataset to obtain the results. For comparison of these algorithms, we made sure that the same cross-validation folds were used for all the different learning algorithms involved in the comparison. All the algorithms were coded in Visual C++ 6.0 with the help of standard template library.

Our version of NBTree uses leave-one-out technique to measure the accuracy of the local NB classifiers. To simplify the implementation, our NBTree can deal with only nominal attributes and treat missing value as a known one. Since the current implementation of our algorithms can only deal with nominal attributes, the entropy-based discretization algorithm [2] were employed to discretize the numeric attributes in the training set for a given fold, as pre-processing.

Table 1. Datasets Information

DATA SET	#EXMP	#ATT	#ATT2	#CLS	DATA SET	#EXMP	#ATT	#ATT2	#CLS
ADULT	48842	14	13	2	PIMA	768	8	5.33	2
ANNEAL	898	38	37.33	5	SATIMAGE	6435	36	36	6
AUSTRALIAN	690	14	13.33	2	SEGMENT	2310	19	17	7
CHESS	3196	36	36	2	SHUTTLE-SMALL	5800	9	7	6
GERMAN	1000	20	14.67	2	SICK	2800	29	26	2
HYP0	3163	25	22.67	2	SOLAR	323	12	9	6
LED7	3200	7	7	10	SOYBEAN-LARGE	683	35	35	19
LETTER	20000	16	15	26	TIC-TAC-TOE	958	9	9	2
MUSHROOM	8124	22	22	2	VEHICLE	846	18	18	4
NURSERY	12960	8	8	5	VOTE	435	16	16	2
PENDIGITS	10992	16	16	10	WAVEFORM	5000	21	19	3

#EXMP: the total number of examples;

#ATT: The number of conditional attributes before preprocessing;

#ATT2: The averaged number of conditional attributes after preprocessing over the three folds;

#CLS: the number of class labels.

Table 2. The mean number of nodes averaged on all folds for each dataset

DATA SET	AVE # NODES	DATA SET	AVE # NODES	DATA SET	AVE # NODES
ADULT	135	MUSHROOM	7.4	SOLAR	3.5
ANNEAL	11.6	NURSERY	175.7	SOYBEAN-LARGE	6.6
AUSTRALIAN	3.4	PENDIGITS	113.1	TIC-TAC-TOE	27.1
CHESS	27.7	PIMA	1.9	VEHICLE	24.8
GERMAN	1	SATIMAGE	107.1	VOTE	8.8
HYP0	13.2	SEGMENT	36.7	WAVEFORM	19.8
LED7	2	SHUTTLE-SMALL	18.4		
LETTER	330.8	SICK	17.4		

For each dataset, Table 2 gives out the mean number of nodes averaged on all folds. Among all the 22 datasets, only 5 datasets have their mean number of nodes more than 100. For the other 17 datasets, the highest mean number of nodes is just 36.7. As discussed in section 3, we separated the 22 datasets into 2 categories, and studied the effect of the pruning strategy on them separately. Table 3 shows the results for those datasets with more than 100 nodes, while table 4 lists out the results for the other 17 datasets

From table 3, we can observe that, for the 5 datasets with more than 100 nodes, NBTREE-PRUNE outperforms NBTREE on 4 of them, while they have a tie on the other dataset (NURSERY). With the pruning strategy, the mean error rate averaged over these 5 datasets drops from 8.98% to 8.60%. In addition, with the significance level set at 95%, NBTREE-PRUNE statistically wins on 3 of them (LETTER, ADULT, SATIMAGE), and loses on no one.

Table 3. Error Rates of the datasets whose naive Bayesian tree has more than 100 nodes

DATA SET	AVE # NODES	NBTREE	NBTREE-PRUNE
LETTER	330.8	11.47%	11.06%
NURSERY	175.7	1.54%	1.54%
ADULT	135	14.13%	14.07%
PENDIGITS	113.1	4.14%	3.76%
SATIMAGE	107.1	13.63%	12.59%
MEAN		8.98%	8.60%

Table 4. Error rates of the datasets whose naive Bayesian tree has less than 100 nodes

DATA SET	AVE # NODES	NBTREE	NBTREE-PRUNE
SEGMENT	36.7	4.42%	4.33%
CHESS	27.7	0.91%	0.88%
TIC-TAC-TOE	27.1	19.42%	19.53%
VEHICLE	24.8	30.02%	30.50%
WAVEFORM	19.8	17.22%	16.76%
SHUTTLE-SMALL	18.4	0.28%	0.29%
SICK	17.4	2.54%	2.39%
HYP0	13.2	1.27%	1.11%
ANNEAL	11.6	0.78%	0.89%
VOTE	8.8	5.98%	4.83%
MUSHROOM	7.4	0.01%	0.01%
SOYBEAN-LARGE	6.6	5.56%	5.42%
SOLAR	3.5	30.93%	30.03%
AUSTRALIAN	3.4	14.35%	14.49%
LED7	2	27.13%	27.06%
PIMA	1.9	25.39%	25.78%
GERMAN	1	25.20%	25.20%
MEAN		12.44%	12.32%

For the other 17 datasets with small-size tree, the performance improvement by the pruning strategy is not so much. The error rate averaged over the 17 datasets goes from 12.44% for NBTree down to 12.32% for NBTree-Prune. Again, NBTREE-PRUNE statistically wins on 1 of them (VOTE), and loses on no one.

5 Related and Future Work

Local accuracy estimation was employed in this paper to post-prune the learnt naïve Bayesian tree. Experimental results proved its effectiveness, especially for the naïve Bayesian trees with relatively large number of nodes.

Lazy Bayesian Rule [8] is another classification method making use of local learning. The pruning strategy in this paper should be also applicable to it, which will be implemented and experimented as the future work.

SNNB [6] is yet another classification method in this category of local learning. It was shown in [7] the effectiveness of local accuracy estimation. It has another advantage in that it is not based on the heuristic search for the best local space. As we know, both NBTree and LBR are in fact in search for a local subspace where naïve Bayes has great performance. The search strategies adopted by NBTree and LBR are something like hill climbing, which suffer from local convergence (or prematurity). One possible way is to integrate the SNNB into the leaf node of NBTree.

In addition, since this work is to use local accuracy estimation based post-pruning to make up the weakness of NBTree's induction, how about using estimated local accuracy to guide the induction process of NBTree?

Furthermore, is it possible to use the measurement of the ranking information 3 for judging the goodness of a local classifier? How about the effect?

Acknowledgements

This work was funded in part by National Natural Science Foundation of China under grant number 60503025, and the Science & Technology Commission of Shanghai Municipality under grant number 03ZR14014.

References

1. Blake, C. L., & Merz, C. J.: UCI repository of machine learning databases. University of California, Irvine, CA (1998)<http://www.ics.uci.edu/~mllearn/MLRepository.html>
2. Fayyad, U. M., & Irani, K. B.: Multi-interval discretization of continuous-valued attributes for classification learning. Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann (1993) 1022-1027
3. Jiang, L., & Guo, Y.: Learning Lazy Naïve Bayesian Classifiers for Ranking. Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence, IEEE Computer Society (2005) 412-416
4. Kohavi, R.: Scaling up the accuracy of naïve-Bayes classifiers: a decision-tree hybrid. Proceedings of the Second International Conference on Knowledge Discovery & Data Mining, Cambridge/Menlo Park: AAAI Press/MIT press (1996) 202-207
5. Xie, Z., Hsu, W., Liu, Z., & Lee, M.-L.: Concept Lattice based Composite Classifiers for High Predictability, Journal of Experimental and Theoretical Artificial Intelligence, 2002, vol. 14(2-3): pp. 143-156
6. Xie, Z., Hsu, W., Liu, Z., & Lee, M.-L.: SNNB: a selective neighborhood-based naïve Bayes for lazy classification. Lecture Notes in Computer Science 2336 (PAKDD 2002), Springer-Verlag (2002) 104-114
7. Xie, Z., Zhang, Q., Hsu, W., & Lee, M.-L.: Enhancing SNNB with Local Accuracy Estimation and Ensemble Techniques. Lecture Notes in Computer Science 3453 (DASFAA 2005), Springer-Verlag, pp.523-535
8. Zheng, Z., & Webb, G. I.: Lazy learning of Bayesian rules. Machine Learning, 41 (2000) 53-84

A New Visualization Method for Patent Map: Application to Ubiquitous Computing Technology

Jong Hwan Suh¹ and Sang Chan Park^{1,2}

¹Department of Industrial Engineering and ²Graduate School of Culture Technology
Korea Advanced Institute of Science and Technology (KAIST),
Guseong-dong, Yuseong-gu, Daejeon, Republic of Korea
{SuhJongHwan, sangchanpark}@major.kaist.ac.kr

Abstract. As technologies develop in faster and more complicated ways, it is getting more important to expect the direction of technological progresses. So many methods are being proposed all around world and one of them is to use patent information. Moreover, with efforts of governments in many countries, many patent analysis methods have been exploited and suggested usually on the basis of patent documents. However, current patent analysis methods have some limitations. In this paper, we suggest a new visualization method for a patent map, which represents patent analysis results with considering both structured and unstructured items of each patent document. And by the adoption of the k-means clustering algorithm and semantic networks, we suggest concrete steps to make a patent map which gives a clear and instinctive insight on the targeted technology. In application, we built up a patent map for the ubiquitous computing technology and discussed an overall view of its progresses.

1 Introduction

With technologies developing in more complicated ways, it's becoming more important to understand how technologies are going on. Analyzing patent information is one of methods to recognize those progresses. And the patent map is the visualized expression of total patent analysis results to understand complex and various patents' information easily and effectively [1]. To build up the patent map, we usually utilize patent documents which contain dozens of items for analysis. In patent documents, structured items mean they are uniform in semantics and in format across patents such as patent number, filing date, or investors. On the other hand, the unstructured ones represent free texts that are quite different in length and content for each patent such as claims, abstracts, or descriptions of the invention. The visualized analysis results of the former items are called patent graphs and those of the later are called patent maps, although loosely patent maps may refer to both cases [2]. Patent documents are often lengthy and rich in technical and legal terminology and are thus hard to read and analyze for non-specialists [3]. Therefore visualization methods to analyze patent information and represent analysis results are more attractive. However, visualization methods for patent maps are limited on their representation. They do not summarize

overall information effectively and consider only one aspect between structured and unstructured items of each patent document. Hence more integrated and balanced visualization approaches are required.

In this paper, we suggest a new visualization method for a patent map, which represents patent analysis results with considering both structured and unstructured items of each patent document. Thereby we can keep the balance of analysis features by using a filing date as a structured item, and a keyword as an unstructured item. The rest of the paper is structured as follows. Section 2 begins by introducing related works and Section 3 gives an overview of our approach. And in Section 4 we apply this visualization method to develop a patent map for the ubiquitous computing technology, and discuss implications of the patent map. In Section 5 finally we conclude the paper with a discussion of patent map's implications in the ubiquitous computing technology.

2 Related Work

Current technological development necessitates conducting searches of patent information to avoid unnecessary investment as well as gaining the seeds for technological development and the applicable fields contained in the parent information. The Japan Patent Office has been producing and providing more than 50 types of expressions and more than 200 maps for several technology fields since 1997 [4]. Many other countries such as Korea and United States also provide patent maps [5, 6]. Researches on intelligent patent analysis have been made as well. The neural methods for mapping scientific and technical information (articles, patents) and for assisting a user in carrying out the complex process of analyzing large quantities of such information are concerned [7]. Machine learning technology is applied to text classification on United States patents to automatically differentiate between patents relating the biotech industry and those unrelated [6].

3 Visualization Method for Patent Map

Steps to implement a new visualization method for a patent map which we propose in this paper are as follows. Firstly, we target a domain technology interested in. And we select keywords to search related patent documents. After that, we redefine a list of keywords for further analysis. And now with a set of patent documents and a list of keywords, we check existence of each keyword within texts of each patent document. To record this step's result, we need to form a matrix with a column index of keywords (1...n) and a row index of patent documents (1...m). So, if the j^{th} keyword exists within texts of the i^{th} patent document, then an element of (i, j) is filled with '1'. But if it does not, then the element of (i, j) is filled with '0' (see Figure 1). The next step is to cluster patent documents by the k-means clustering algorithm using the completed matrix. Here each keyword's value between '0' or '1' plays as a feature's value for

each patent document. So the keywords' values are used to classify patent documents into 'k' groups (see Figure 2). Now with clustered patent documents, we investigate what keyword each group has. For example, let's assume that patent documents 'A' and 'B' belong to the group 1. According to the matrix in Figure 1, patent document 'A' has keywords of 'a' and 'c' and patent document 'B' has keywords of 'b' and 'c'. Then, the group 1 consists of three keywords of 'a', 'b', and 'c'. Like this way, we investigate keywords for each group (see Figure 3).

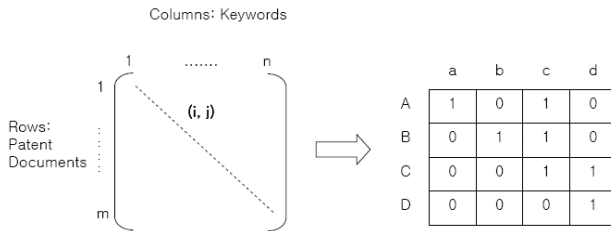


Fig. 1. A matrix of which elements are filled with '0' or '1' according to whether a keyword exists within texts of a patent document

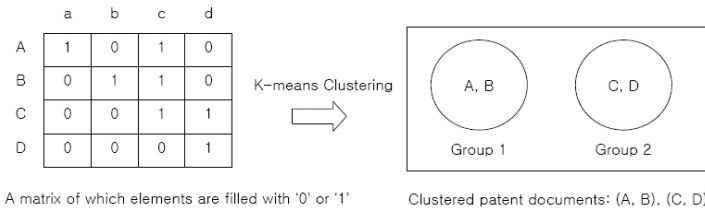


Fig. 2. Clustering patent documents by the k-means clustering algorithm using the completed matrix

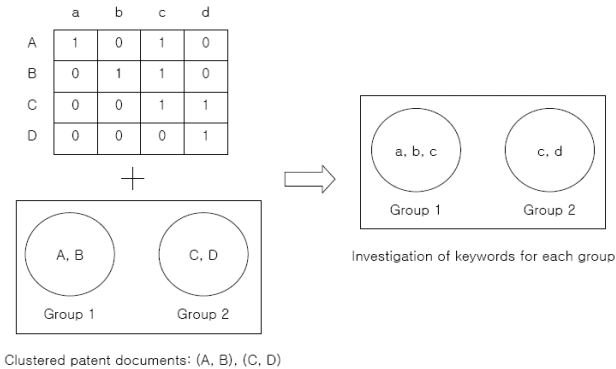


Fig. 3. Investigating keywords for each group after clustering patent documents

And using the list of keywords for each group, we make a semantic network. In the figure, group 1 has keywords of ‘a’, ‘b’, and ‘c’. On the other hand group 2 has keywords of ‘c’ and ‘d’. Then two groups share ‘b’ and therefore relationship between two groups can be represented by three nodes: (a, b), (c), and (d). Here the shared node is higher than the others, so arrows are drawn from (c) to (a, b) and (d). Like this way, we make a semantic network which consists of nodes with a keyword or more than two keywords (see Figure 4).

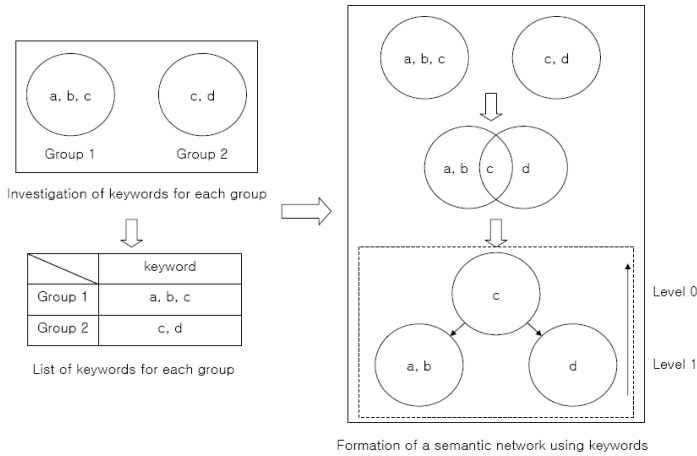


Fig. 4. Forming semantic networks using lists of keywords for each group

Actually, the semantic network is based on the previous steps such as ‘clustering patent documents with the k-means clustering algorithm’ and ‘investigating key words for clustered patent documents’. Therefore, the semantic network is dependent on the number of groups which is set temporarily by the k-means clustering algorithms, so there are many semantic networks. There are many executable programs which can perform the k-means clustering algorithm. Using any of them, easily we can repeat the clustering with increasing the number of groups. And for each time based on the clustering result, we repeat both steps of ‘investigation of keywords for each group’ and ‘formation of a semantic network’. Finally, we get ‘n’ semantic networks after ‘n’ repetitions but here we do not consider the case when the number of groups is just one. And then we have to choose one of many semantic networks. Usually we select one which explains the most of the relations of keywords. Actually, this is a manual operation. But usually as the number of groups in the chosen semantic network increases, it gets better to explain the relations of keywords by the semantic network. However, too big number makes it worse to form a semantic network therefore we have to find a point of comprise.

Since now, we have explained how to form a semantic network of keywords, unstructured items, from patent documents related to the target technology. From now on, we explain how to make use of structured items to complete a patent map on the basis of semantic networks. Let’s assume that finally we reached the semantic network as shown in Figure 5. Firstly, we have to investigate a filing date of each node in

the semantic network. The filing date of each node is the earliest filing date among patent documents which have keywords of the node. For example, in the figure, node 2 consists of keywords of 'a' and 'b'. And if 'a' belongs to patent documents of 'A', and 'b' belongs to patent documents 'B', then the filing date which node 1 has is the earliest filing date among document 'A' and 'B'. Therefore, the filing date of node 2 is '1997-11-27'. Similarly, the filing date of node 3 is '2000-08-18'. Like this way, we accomplish a semantic network of keywords of which each node has its filing date. Now the semantic network has both aspects of structured and unstructured items within patent documents. Finally, we move on to the stage for building up a patent map using the accomplished semantic network. Nodes of the semantic network are rearranged according to their filing dates. Figure 6 shows an example of the proposed patent map.

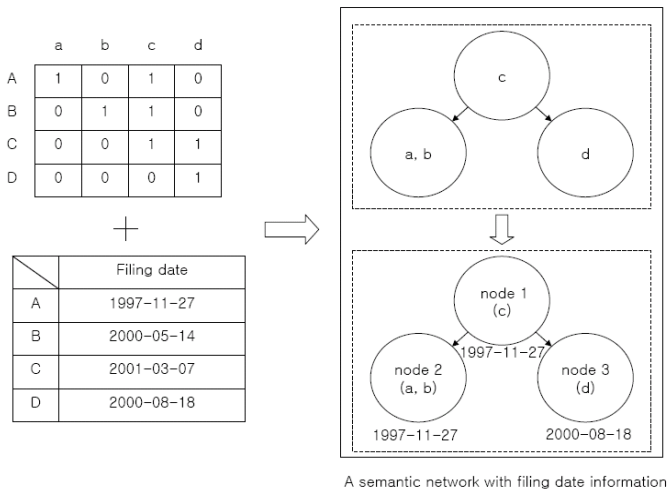


Fig. 5. Forming a semantic network with filing date information

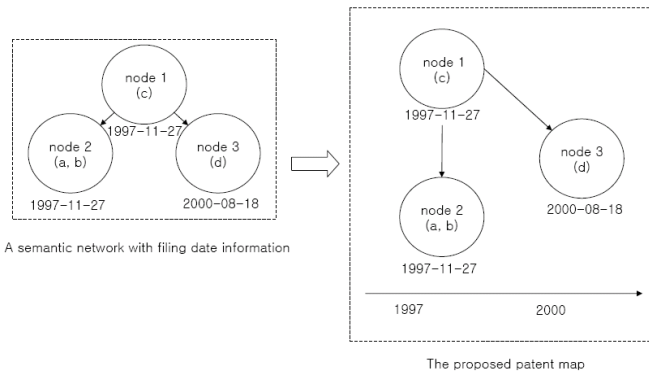


Fig. 6. Forming a patent map on the basis of a semantic network with filing date information

4 Application to Ubiquitous Computing Technology

In the long term, Ubiquitous Computing is expected to take on great economic significance. So a lot of patents related to ubiquitous computing technology are being invented all around world. According to these circumstances, it's been important to analyze those patents. Therefore in this paper, we targeted the ubiquitous computing technology for the application of our visualization method. With steps in Section 3, we ended up with a patent map for ubiquitous computing technology. Those steps for the application are described as follows. Firstly we searched keywords related to the ubiquitous computing technology. And then we searched patent documents related to the ubiquitous computing technology using those keywords. Totally 96 patent documents were searched and we used them for patent information analysis. We rebuilt up the list of keywords based on the searched patent document and the final list of keywords is as shown in table 1. On the basis of keywords in table 1, we clustered 96 patent documents using Clementine™. And then we made semantic networks with increasing the number of groups, and selected a semantic network with 5 groups. The list of keywords for each group is shown in table 2.

Table 1. The list of keywords redefined from searched patent documents, and the earliest filing dates of patent documents each keyword belongs to

Number	Keyword	Earliest Filing Date
1	RFID	1987-04-07
2	Universal PnP	2002-10-01
3	Trigger	2002-08-22
4	HAVI	2002-08-22
5	HTML↔VXML interchangeability	2003-04-02
6	Fabrication	2002-08-22
7	Shop floor	1987-08-18
8	Magnetic memory device	2003-04-30
9	Logistics	198704-07
10	Automatic identification	1987-04-07
11	PDA, mobile, handheld device	2002-05-21
12	Intelligent	2002-12-27
13	Remote Control System	2002-10-02
14	GPS	2002-05-21
15	Ubiquitous computing	2000-06-02
16	Sensor network	2001-01-31
17	Smart	2001-03-15
18	Identification	1987-04-07
19	Manufacturing	1997-08-22
20	Distribution	2000-06-02
21	Lifecycle	2002-08-22
22	Healthcare	1987-08-18
23	Blue tooth	2002-08-22
24	Tracking	1991-12-24
25	Context awareness	2002-07-29
26	Inventory	1987-04-07

Table 2. The list of keywords of each group of clustered patent documents with 5 groups

Group	Keyword
1	1, 2, 3, 9, 10, 11, 13, 14, 15, 16, 18, 19, 20, 21, 24, 25, 26
2	1, 3, 4, 6, 9, 15, 17, 18, 20, 21, 23, 25, 26
3	2, 4, 5, 8, 10, 11, 12, 13, 14, 15, 17, 19, 20, 23, 24, 25
4	1, 4, 7, 9, 10, 11, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26
5	1, 11, 12, 13, 15, 16, 19, 20, 23, 24, 26

Using the result of clustering, we completed the final semantic network as shown in Figure 7. And then based on the semantic network, the patent map for the ubiquitous computing technology was made as shown in Figure 8.

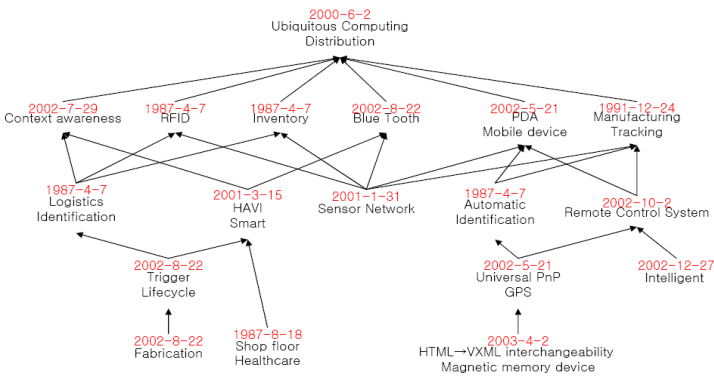


Fig. 7. A semantic network with nodes of keywords from clustered patent documents and filing information

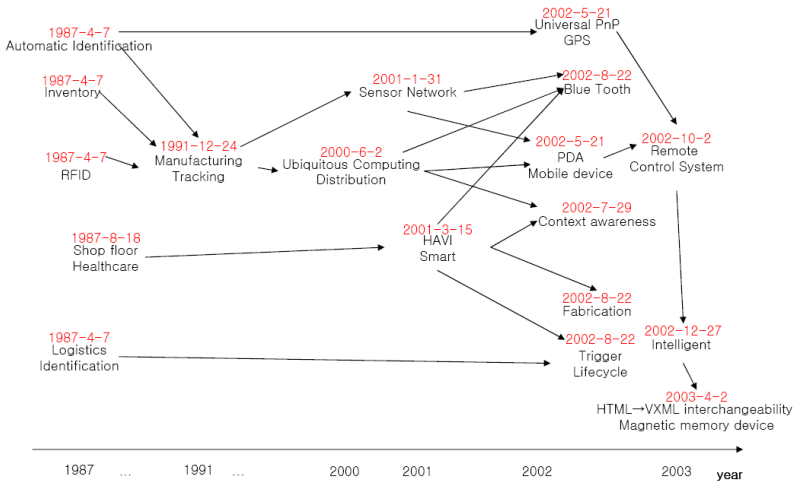


Fig. 8. A patent map based on a semantic network of Figure 7

5 Conclusion

In this paper, we proposed a new visualization method for a patent map and applied it to the ubiquitous computing technology. Comparing to the other methods in the literature of Section 2, our research considered both sides of structured items and the unstructured items of patent documents. Thereby it provided a balanced approach to analyze patent information. Moreover, we suggested concrete steps to form semantic networks by the k-means clustering algorithm with keywords and filing date, and finally a patent map as described concretely in Section 3. By doing so, we expect a patent map will turn out a more intelligent and sophisticated patent map. Also non expert also can make a patent map with more understandings because this paper explains how to make a patent map in clear and instinctive ways.

In addition, using the suggested framework of a visualization method for a patent map, we suggested a semantic network and a patent map for the ubiquitous computing technology. From the patent map, we can find what kinds of patents on the ubiquitous computing technology have appeared and how those patents are merged and divided as time passes. Figure 8 shows patents on the ubiquitous computing technology have progressed towards HTML→VXML interchangeability and magnetic memory devices in 2003 since the patents related to automatic identification, inventory, RFID, and logistics appeared in 1987. Like this, the proposed patent map gives a complete view of the development of patents which are related to the target technology. Also it helps the person concern on the target technology to have an insight to the next patents, thereby to avoid unnecessary investments and find the seeds for the next patent.

Acknowledgement. This research is supported by the KAIST Graduate School of Culture Technology.

References

1. WIPO: Patent Map with Exercises (related), URL: www.wipo.org/sme/en/activities/meetings/china_most_03/wipo_ip_bis_ge_03_16.pdf
2. S. J. Liu: A Route to a Strategic Intelligence of Industrial Competitiveness, The first Asia-Pacific Conference on Patent Maps, Taipei (2003), 2-13
3. Y.H. Tseng, Y.M. Wang, D.W. Juang, and C.J. Lin: Text Mining for Patent Map Analysis, IACIS Pacific 2005 Conference Proceedings (2005), 1109-1116
4. Japan Institute of Invention and Innovation: Guide Book for Practical Use of Patent Map for Each Technology Field, URL: www.apic.jiii.or.jp/p_f/text/text/5-04.pdf
5. J.H. Ryoo(KIPI), and I.G. Kim(KIPO): Workshop H-What patent analysis can tell about companies in Korea, Far East Meets West In Vienna 2005, URL: www.european-patent-office.org/epidos/conf/jpinfo/2005/_pdf/report_workshop_h.pdf
6. David B. and Peter C.: Machine Learning for Patent Classification, URL: www.stanford.edu/
7. J.C. Lamirel, S.A. Shehabi, M. Hoffmann and C. Francois: Intelligent patent analysis through the use of a neural network: experiment of multi-view point analysis with the MultiSom model (2002), URL: acl.ldc.upenn.edu/W/W03/W03-2002.pdf

Local Linear Logistic Discriminant Analysis with Partial Least Square Components^{*}

Jangsun Baek and Young Sook Son

Department of Statistics, Chonnam National University, Gwangju 500-757,
South Korea
{jbaek, ysson}@chonnam.ac.kr

Abstract. We propose a nonparametric local linear logistic approach based on local likelihood in multi-class discrimination. The combination of the local linear logistic discriminant analysis and partial least square components yields better prediction results than the conventional statistical classifiers in case where the class boundaries have curvature. We applied our method to both synthetic and real data sets.

1 Introduction

Many classification methods are used in data mining. The support vector machine or the neural network have been successfully applied to many fields (Furey et al. 2000, Tarca and Cooke 2005). However they only predict the class label of a new observation to be classified, not providing any estimate of the class probability. Classical statistical discriminant analysis techniques such as Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), logistic discriminant analysis can provide the estimates of the underlying probability. They are all parametric since they assume some parametric model such as normal distribution. The logistic discriminant analysis can be viewed as a partially parametric approach as only the ratio of the class densities are modeled. Non-parametric logistic approach based on local likelihood is possible in discriminant analysis when the parametric class distribution is unknown (Loader 1999). Fan and Gijbels (1996) investigated the local fitting of binary regression models. Zhu and Hastie (2004) proposed a penalized logistic regression method.

The main problem in using local fitting techniques is the curse of dimensionality (Hastie et al. 2001), where in high dimensions, large amount of data is needed to get accurate local estimates. When considering local logistic regression it needs to combine the localization with dimension reduction. There are two general approaches to cope with the curse of dimensionality. The first approach is the selection of relevant original features. Stepwise selection procedures such as Sequential Forward Selection or Sequential Backward Selection are commonly employed. The second approach is the feature transformation. In multivariate statistics Principal Component Analysis (PCA) is popular. Singular Value Decomposition (SVD) technique is used to reduce dimensionality

^{*} This study was financially supported by research fund of Chonnam National University in 2003.

(Fukunaga 1990, West et al. 2001). Nguyen and Rocke (2002) proposed using the method of Partial Least Squares (PLS) for microarray data. Baek and Kim (2004) used PLS as discriminating components for face recognition. In this research we propose a local linear logistic approach based on local likelihood in multi-class discrimination by combining PLS to reduce the predictor dimension and avoiding multicollinearity.

In Section 2 we describe the local linear logistic discriminant analysis with PLS components. Experiments with three synthetic and a real data sets are carried out in Section 3 to show that the classification rates of the local linear logistic discriminant analysis are higher than those of conventional statistical discrimination methods.

2 Local Linear Logistic Discriminant Analysis

Suppose there are K classes (C_1, C_2, \dots, C_K) to be classified, and we have the training data set of size $n \{(\mathbf{y}_i, \mathbf{x}_i); i = 1, 2, \dots, n\}$, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})'$ is the p -dimensional random feature vector and $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{i(k-1)})'$ is the $(K - 1)$ -dimensional vector of zero-one Bernoulli variables indicating the class of origin of \mathbf{x}_i . That is, $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{i(k-1)})'$ with $y_{il} = 1$ and $y_{ij} = 0$ for all $j \neq l$ if \mathbf{x}_i belongs to the l th class C_l , and $\mathbf{y}_i = (0, 0, \dots, 0)'$ if \mathbf{x}_i belongs to the K th class C_K .

Let $\pi_l(\mathbf{x}) = P(C_l|\mathbf{x})$ be the probability of an observation with feature \mathbf{x} belonging to class C_l . Corresponding to the conventional but arbitrary choice of C_K as the base class, the logistic regression model assumes that the log-odds $\log(\pi_l(\mathbf{x})/\pi_K(\mathbf{x}))$ is represented as $\eta_l(\mathbf{x})$, a function of feature variables x_1, x_2, \dots, x_p . Then the probability that the observation with \mathbf{x} belongs to group C_l and C_K is, respectively

$$\begin{aligned} \pi_l(\mathbf{x}) &= \exp(\eta_l(\mathbf{x})) / \{1 + \sum_{k=1}^{K-1} \exp(\eta_k(\mathbf{x}))\}, \quad l = 1, 2, \dots, K - 1, \\ \pi_K(\mathbf{x}) &= 1 / \{1 + \sum_{k=1}^{K-1} \exp(\eta_k(\mathbf{x}))\}. \end{aligned} \tag{1}$$

The ordinary logistic regression assumes $\eta_l(\mathbf{x})$ is linear, that is,

$$\eta_l(\mathbf{x}) = \beta_{l0} + \beta_{l1}x_1 + \dots + \beta_{lp}x_p, \quad l = 1, 2, \dots, K - 1.$$

The function $\eta_l(\mathbf{x})$ is called the link function in GLM (Generalized Linear Model: McCullagh and Nelder 1989).

Since \mathbf{y} follows the multinomial distribution, the log-likelihood function l is

$$l = \sum_{i=1}^n \left(\sum_{k=1}^{K-1} y_{ik} \log\{\pi_k(\mathbf{x}_i)\} + \left(1 - \sum_{k=1}^{K-1} y_{ik}\right) \log\left\{1 - \sum_{k=1}^{K-1} \pi_k(\mathbf{x}_i)\right\} \right).$$

The parameter $\beta_l' = (\beta_{l0}, \beta_{l1}, \dots, \beta_{lp})$ is estimated by maximizing the log-likelihood l . The class probability $\pi_l(\mathbf{x})$ is then estimated as $\hat{\pi}_l(\mathbf{x}; \hat{\beta}_l)$ by replacing β_l with its maximum likelihood estimate $\hat{\beta}_l$, and we classify a new observation into the class which has the largest probability estimate. This classification procedure is called parametric (linear) logistic discrimination.

The advantage of the parametric (linear) logistic discrimination lies in the easy interpretation of the model parameter β_l . However, when the link function has some curvature different from the linearity, the linear logistic model cannot fit the data very well. A simple nonparametric alternative to the parametric logistic model which allows for such curvature, and yet retains the easy interpretation of parameters is the approximation of the unknown smooth link function $\eta_l(\mathbf{x})$ by a local polynomial of order ν . Suppose that the first derivative of $\eta_l(\mathbf{x})$ at the point \mathbf{x}_0 exists. We then approximate the unknown smooth link function $\eta_l(\mathbf{x})$ locally by a polynomial of order $\nu = 1$. Suppose we have an observation with the feature \mathbf{x}_0 to be classified. A Taylor expansion gives, for \mathbf{x} in a neighborhood of \mathbf{x}_0 ,

$$\begin{aligned} \eta_l(\mathbf{x}) &\approx \eta_l(\mathbf{x}_0) + \frac{\partial \eta_l(\mathbf{x}_0)}{\partial \mathbf{x}'}(\mathbf{x} - \mathbf{x}_0) \\ &= \beta_{l0} + \beta_{l1}(x_1 - x_{01}) + \beta_{l2}(x_2 - x_{02}) + \dots + \beta_{lp}(x_p - x_{0p}) \\ &= \beta_l' \mathbf{z}, \quad l = 1, 2, \dots, K - 1, \end{aligned}$$

where $\mathbf{z}' = (1, x_1 - x_{01}, \dots, x_p - x_{0p})$ and $\beta_l' = (\beta_{l0}, \beta_{l1}, \dots, \beta_{lp})$ with $\beta_{l0} = \eta_l(\mathbf{x}_0)$, $\beta_{lj} = \partial \eta_l(\mathbf{x}_0) / \partial x_j$, $j = 1, 2, \dots, p$. These polynomials are fitted for $\beta_T = (\beta_1', \beta_2', \dots, \beta_{K-1}')'$ by maximizing the local log-likelihood function $l^*(\beta_T)$;

$$l^*(\beta_T) = \sum_{i=1}^n K_B(\mathbf{z}_i) \left(\sum_{k=1}^{K-1} y_{ik} \log\{\pi_k(\mathbf{z}_i)\} + \left(1 - \sum_{k=1}^{K-1} y_{ik}\right) \log\{\pi_k(\mathbf{z}_i)\} \right), \quad (2)$$

where $K_B(\mathbf{u}) = K(B^{-1}\mathbf{u})/|B|$, $K(\cdot)$ is a p -variate nonnegative kernel function, B is a nonsingular $p \times p$ bandwidth matrix, and $|B|$ denotes its determinant. For simplicity we assume that $K(\cdot)$ is a multivariate probability density function with $\int K(\mathbf{u})d\mathbf{u} = 1$ and $\int \mathbf{u}K(\mathbf{u})d\mathbf{u} = 0$. $K_B(\cdot)$'s are the weights for the log-densities in the local likelihood.

Define the local score statistic as

$$U(\beta_T) = \frac{\partial l^*(\beta_T)}{\partial \beta_T}.$$

The parameters are estimated by solving the local score equation $U(\beta_T) = 0$, and the estimates are obtained by the iterative Fisher scoring algorithm of the form

$$\hat{\beta}_T^{(s+1)} = \hat{\beta}_T^{(s)} + I(\hat{\beta}_T^{(s)})^{-1}U(\hat{\beta}_T^{(s)}), \quad s = 0, 1, 2, \dots,$$

with the Fisher's information matrix $I(\beta_T)$ which will be introduced later. Then the value of the link function is estimated as $\hat{\eta}_l(\mathbf{x}_0) = \hat{\beta}_{l0}$, and we classify the observation with feature \mathbf{x}_0 into the class with the largest $\hat{\pi}_l(\mathbf{x}_0)$ which is calculated by substituting $\hat{\eta}_l(\mathbf{x}_0) = \hat{\beta}_{l0}$ for $\eta_l(\mathbf{x}_0)$ in equation (1). We call this procedure the local linear logistic discriminant analysis.

Let $\mathbf{z}_i' = (1, x_{i1} - x_{01}, x_{i2} - x_{02}, \dots, x_{ip} - x_{0p})$. It is easy to show that

$$\begin{aligned} \partial \pi_l / \partial \beta_l &= \pi_l(1 - \pi_l)\mathbf{z} \\ \partial \pi_k / \partial \beta_l &= -\pi_k \pi_l \mathbf{z}, \quad k \neq l, \quad k = 1, 2, \dots, K. \end{aligned}$$

Then the l th element of the score statistic $U_l(\beta_l) = \partial l^*(\beta_T)/\partial \beta_l$ in $U(\beta_T) = (U_1(\beta_1)', U_2(\beta_2)', \dots, U_{K-1}(\beta_{K-1})')'$, is obtained by taking the partial derivative of (2) as follows:

$$\begin{aligned} U_l(\beta_l) &= \sum_{i=1}^n K_B(\mathbf{z}_i) \left(\sum_{k=1}^{K-1} \frac{y_{ik}}{\pi_k(\mathbf{z}_i)} \frac{\partial \pi_k(\mathbf{z}_i)}{\partial \beta_l} + \frac{(1 - \sum_{k=1}^{K-1} y_{ik})}{\pi_K(\mathbf{z}_i)} \frac{\partial \pi_K(\mathbf{z}_i)}{\partial \beta_l} \right) \\ &= \sum_{i=1}^n K_B(\mathbf{z}_i) \left(\sum_{k=1, k \neq l}^{K-1} y_{ik} \{-\pi_l(\mathbf{z}_i)\} \mathbf{z}_i + y_{il} \{1 - \pi_l(\mathbf{z}_i)\} \mathbf{z}_i \right. \\ &\quad \left. + (1 - \sum_{k=1}^{K-1} y_{ik}) \{-\pi_l(\mathbf{z}_i)\} \mathbf{z}_i \right) \\ &= \sum_{i=1}^n K_B(\mathbf{z}_i) \{y_{il} - \pi_l(\mathbf{z}_i)\} \mathbf{z}_i. \end{aligned}$$

Therefore the local score statistic is $U(\beta_T) = (U_1(\beta_1)', \dots, U_{K-1}(\beta_{K-1})')'$, with $U_l(\beta_l) = \sum_{i=1}^n K_B(\mathbf{z}_i)(y_{il} - \pi_l(\mathbf{z}_i))\mathbf{z}_i$.

The Fisher's information matrix which is defined as

$$I(\beta_T) = E \left(-\frac{\partial^2 l^*(\beta_T)}{\partial \beta_T \partial \beta_T'} \right)$$

tells us how much information about the true parameter is given by the data. Since the parameter β_T is a $(p + 1)(K - 1) \times 1$ vector, the Fisher's information matrix $I(\beta_T)$ is a $(p + 1)(K - 1) \times (p + 1)(K - 1)$ matrix. Let $I_{jk}(\beta_T) = E(-\partial U_j(\beta_j)/\partial \beta_k)$. Then it can be shown that

$$\begin{aligned} I_{jk}(\beta_T) &= \sum_{i=1}^n K_B(\mathbf{z}_i) \pi_j(\mathbf{z}_i) \{1 - \pi_j(\mathbf{z}_i)\} \mathbf{z}_i \mathbf{z}_i', \quad j = k, \\ &= -\sum_{i=1}^n K_B(\mathbf{z}_i) \pi_j(\mathbf{z}_i) \pi_k(\mathbf{z}_i) \mathbf{z}_i \mathbf{z}_i', \quad j \neq k. \end{aligned} \tag{3}$$

The Fisher's information matrix consists of these $(p + 1) \times (p + 1)$ $I_{jk}(\beta_T)$ matrices as its (j, k) th element. That is $I(\beta_T) = (I_{jk}(\beta_T))$, where (\cdot) is the matrix notation. Assume \mathbf{Z} is the design matrix of the feature data, that is

$$\mathbf{Z} = \begin{pmatrix} 1 & x_{11} - x_{01} & x_{12} - x_{02} & \cdots & x_{1p} - x_{0p} \\ 1 & x_{21} - x_{01} & x_{22} - x_{02} & \cdots & x_{2p} - x_{0p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} - x_{01} & x_{n2} - x_{02} & \cdots & x_{np} - x_{0p} \end{pmatrix} = \begin{pmatrix} \mathbf{z}_1' \\ \mathbf{z}_2' \\ \vdots \\ \mathbf{z}_n' \end{pmatrix}.$$

Let $\mathbf{W}_j = \text{diag}(w_{ji})$, where $w_{ji} = K_B(\mathbf{z}_i) \pi_j(\mathbf{z}_i) \{1 - \pi_j(\mathbf{z}_i)\}$ is the i th diagonal element of the diagonal matrix \mathbf{W}_j . Then we can express $I_{jj}(\beta_T)$ in (3) with the matrix product as

$$I_{jj}(\beta_T) = \mathbf{Z}' \mathbf{W}_j \mathbf{Z}, \quad j = 1, 2, \dots, K - 1.$$

Since $K_B(\mathbf{z}_i) > 0$, $0 < \pi_j(\mathbf{z}_i) < 1$ and \mathbf{W}_j is a diagonal matrix, \mathbf{W}_j is invertible. Thus $\text{rank}(\mathbf{W}_j \mathbf{Z}) = \text{rank}(\mathbf{Z})$ and

$$\text{rank}(I_{jj}(\beta_T)) = \text{rank}(\mathbf{Z}' \mathbf{W}_j \mathbf{Z}) \leq \min(\text{rank}(\mathbf{Z}'), \text{rank}(\mathbf{W}_j \mathbf{Z})) = \text{rank}(\mathbf{Z}).$$

Therefore if \mathbf{Z} is not of full rank, the j th information matrix $I_{jj}(\boldsymbol{\beta}_{\mathcal{T}})$ does not have the inverse because it is singular, and neither does the Fisher's information matrix $I(\boldsymbol{\beta}_{\mathcal{T}})$. This makes the procedure of estimating parameters by the Fisher scoring algorithm unstable in case of less than full rank. The situation that the rank of the design matrix \mathbf{Z} is less than full rank of $(p + 1)$ is caused by multicollinearity among the feature variables. The singular problem caused by multicollinearity can be overcome by using PLS components. The presence of singular information matrix may be bypassed by using penalized logistic regression model which needs additional regularization parameter (Zhu and Hastie 2004).

As the dimension of the feature vector increases, larger sample is needed for accurate nonparametric local estimation (curse of dimensionality). Thus we try to reduce the dimension of feature vector. One of the most popular approaches to the reduction of the feature dimension is based on PCA. PCA is used to reduce the high dimensional feature to only a few feature components which explain as much of the observed total feature variation as possible. This is achieved without regard to the variation of the observation's class or group. In contrast to PCA, PLS chooses components so that the sample covariance between the group variable and a linear combination of the original feature vector is maximum. Their orthogonality also confirms the non-singularity of information matrix. The PLS method is well suited for the prediction of regression models with many predictor variables (Garthwaite 1994). Recently it was applied to biometric data classification (Nguyen and Rocke 2002) and face recognition (Baek and Kim 2004). We will use PLS components as predictors in the local linear logistic discriminant analysis for both nonsingular information matrix and dimension reduction.

The objective criterion for constructing PLS components is to sequentially maximize the covariance between the class variable and a linear combination of the features. Suppose \mathbf{X} is the standardized $n \times p$ feature data matrix, and $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_n)'$ is the $n \times (K - 1)$ multinomial class indicator matrix. PLS is to find the weight vector \mathbf{b}_i such that

$$\mathbf{b}_i = \arg \max_{\mathbf{b}'\mathbf{b}=1, \mathbf{c}'\mathbf{c}=1} Cov^2(\mathbf{X}\mathbf{b}, \mathbf{Y}\mathbf{c}), \quad i = 1, 2, \dots, q,$$

subject to the orthogonality constraint

$$\mathbf{b}_i' \mathbf{S} \mathbf{b}_j = 0 \quad \text{for all } 1 \leq i < j,$$

where \mathbf{b} , \mathbf{c} are unit vectors, and $\mathbf{S} = \mathbf{X}'\mathbf{X}$. The procedure is called the multivariate PLS. The i th PLS component is the linear combination of the original features, $\mathbf{X}\mathbf{b}_i$. PLS extracts the components to maximize the correlation between the component and the class variable. PLS component scores can be calculated by standard statistical packages, for example, SAS.

3 Experimental Results

In synthetic data sets we have 2-dimensional feature vector $\mathbf{x} = (x_1, x_2)'$ and there are two classes to be classified. We assume in the first data set that the feature

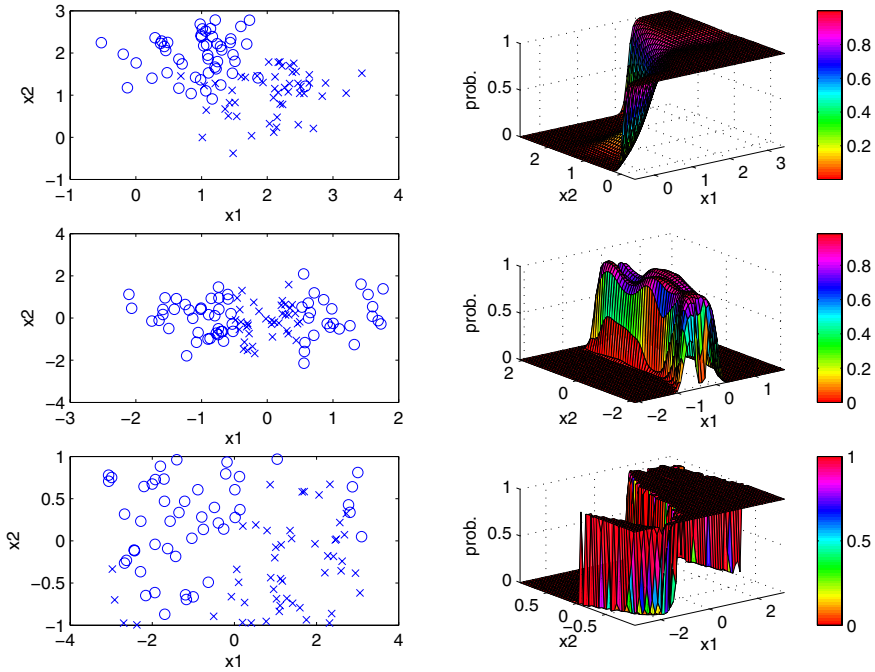


Fig. 1. The scatter plots and the class probability estimates $\hat{\pi}_1(\mathbf{x})$ of three synthetic data sets

vector for each class follows bivariate normal distribution with different mean vector and the same covariance matrix. More specifically, the first and second class feature distribution is $N_2(\boldsymbol{\mu}_1, \Sigma), N_2(\boldsymbol{\mu}_2, \Sigma)$, with $\boldsymbol{\mu}_1 = (1, 2)'$, $\boldsymbol{\mu}_2 = (2, 1)'$, $\Sigma = \begin{pmatrix} 1/3 & 1/2 \\ 1/2 & 1/3 \end{pmatrix}$, respectively. We generated 50 feature vectors randomly from each distribution, and selected 15 observations randomly from each class to form the test set. Thus the training set consists of the rest of 70 observations. The second data set consists of 100 random observations from $N_2(\boldsymbol{\mu} = (0, 0)', \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix})$, and is divided into the first class sample of observations with $-0.5 < x_1 < 0.5$ and the other second class sample. For the third data set 100 (x_1, x_2) 's are generated from $x_1 \sim Uniform(-\pi, \pi)$, $x_2 \sim Uniform(-1, 1)$. If $x_2 \leq \sin(x_1)$, the observation belongs to the first class, and to the second class otherwise. The 30% of total sample is drawn randomly as the test set from the second and the third data set. Each row of Fig. 1 is the scatter plot of the randomly generated observations with its class identification, and the surface plot of the class probability estimates $\hat{\pi}_1(\mathbf{x})$ using the local linear logistic discriminant analysis for three synthetic data sets. We used the standard bivariate normal density as the kernel function, and $B = \begin{pmatrix} (\lambda s_1)^2 & 0 \\ 0 & (\lambda s_2)^2 \end{pmatrix}$ as the bandwidth matrix with the constant λ for the optimal bandwidth, which produced best classification rate among 10 λ s between 0.2 and 2.0. s_1 and s_2 is the standard deviation of x_1 and x_2 respectively.

Table 1. Mean classification rates of 100 replications for each data set

Data set	LDA	QDA	Logistic	Local logistic
1	0.9147 (0.0484)	0.9123 (0.0523)	0.9137 (0.0551)	0.9240 (0.0526)
2	0.5127 (0.0997)	0.9372 (0.0469)	0.6051 (0.0828)	0.9475 (0.0418)
3	0.8224 (0.0779)	0.8261 (0.0807)	0.8177 (0.0780)	0.9425 (0.0518)

Table 2. Classification rates for heart disease data

# of PLS	LDA	QDA	Logistic	Local logistic
2	0.8480	0.8446	0.8446	0.8514 ($\lambda : 0.7$)
3	0.8480	0.8412	0.8446	0.8514 ($\lambda : 1.6$)
4	0.8446	0.8480	0.8345	0.8480 ($\lambda : 1.6$)

The first two rows of Fig.1 shows that the local linear logistic classifier finds the correct boundary as LDA (and logistic classifier) and QDA do in cases where the boundary between two classes is either linear or quadratic. On the other hand, the proposed classifier outperforms the others in case where the boundary has more curvature (the third row of Fig. 1). We repeated the data generation 100 times for each synthetic case and showed the mean and standard deviation of the classification rates for each classifier in Table 1. While there is no difference in performance between LDA (and logistic classifier) or QDA and the local linear logistic classifier for data set 1 and 2, the mean classification rate of the proposed classifier is significantly higher than those of others in data set 3 where the boundary is neither linear nor quadratic.

The real data in our experiments is the heart disease data (Li and Biswas 2002). It contains a mixture of eight nominal and five numeric features from two classes: people with no heart disease and people with heart disease. Li and Biswas (2002) selected five most influential features (two numeric-valued features: maximum heart rate during stress test, EGC ST depression, three nominal-valued features: chest pain type, chest pain during stress test, indication of calcification of major cardiac arteries) from the original 13 features. We used these five features with no missing values, observed from 296 individuals. We extracted PLS components from these five most influential original features, and plugged them into the classifiers. The multivariate standard normal density was used as the kernel, and the constant λ for optimal bandwidth was chosen by the leave-one-out cross validation for local linear logistic classifier. Table 2 contains the classification rates with the number of PLS components being used for each classifier. The local linear logistic classifier attains higher classification rate in all cases. Its rate is slightly lower when four PLS components are used because of the lack of locality in high dimensions.

References

- Baek, J., Kim, M.: Face recognition using partial least squares components. *Pattern Recognition* **37** (2004) 1303-1306
- Fan, J., Gijbels, I.: *Local polynomial modeling and its applications*. London: Chapman & Hall (1996)
- Fukunaga, K.: *Introduction to statistical pattern recognition*. San Diego CA: Academic Press (1990)
- Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16** (2000) 906-914
- Garthwaite, P. M.: An interpretation of partial least squares. *J. Am. Stat. Assoc.* **89** (1994) 122-127
- Hastie, T., Tibshirani, R., Friedman, J.: *The elements of statistical learning*. New York: Springer (2001)
- Li, C., Biswas, G.: Unsupervised learning with mixed numeric and nominal data. *IEEE Transactions on knowledge and data engineering* **14** (2002) 673-690
- Loader, C.: *Local regression and likelihood*. New York: Springer (1999)
- McCullagh, P., Nelder, J. A.: *Generalized linear models*, 2nd ed. London: Chapman and Hall (1989)
- Nguyen, D., Rocke, D.: Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **18** (2002) 39-50
- Tarca, A. L., Cooke, J. E. K.: A robust neural networks approach for spatial and intensity-dependent normalization of cDNA microarray data. *Bioinformatics* **21** (2005) 2674-2683
- West, M., Blanchette, C., Dressman, H., Huang, F., Ishida, S., Spang, R., Zuzan, H., Olason, J., Marks, I., Nevins, J.: Predicting the clinical status of human breast cancer by using gene expression profiles. *PNAS* **98** (2001) 11462-11467
- Zhu, J., Hastie, T.: Classification of gene microarrays by penalized logistic regression. *Biostatistics* **5** (2004) 427-443

Activity Mining: Challenges and Prospects*

Longbing Cao

Faculty of Information Technology, University of Technology Sydney, Australia
lbcao@it.uts.edu.au

Abstract. Activity data accumulated in real life, e.g. in terrorist activities and fraudulent customer contacts, presents special structural and semantic complexities. However, it may lead to or be associated with significant business impacts. For instance, a series of terrorist activities may trigger a disaster to the society, large amounts of fraudulent activities in social security program may result in huge government customer debt. Mining such data challenges the existing KDD research in aspects such as unbalanced data distribution and impact-targeted pattern mining. This paper investigates the characteristics and challenges of activity data, and the methodologies and tasks of activity mining. Activity mining aims to discover impact-targeted activity patterns in huge volumes of unbalanced activity transactions. Activity patterns identified can prevent disastrous events or improve business decision making and processes. We illustrate issues and prospects in mining governmental customer contacts.

Keywords: Activity data, activity mining, impact-targeted mining, unbalanced data.

1 Introduction

Activities can be widely seen in many areas, and may lead to or be associated with impact to the world. For instance, terrorists undertake a series of terrorist activities which finally lead to a disaster to our society [8]. In social security network, a large proportion of separated fraudulent activities can result in huge volumes of governmental customer debt [3]. In addition, activity data may be found in business world with frequent customer contacts [10], business intervention and events, and business outcome oriented processes, as well as event data [12], national and homeland security activities [8] and criminal activities [7]. Such activities are recorded and accumulated in relevant enterprise activity transactional files. Activity data hides rich information about the relations between activities, between activities and operators, and about the impacts of activities or activity sequences on business outcomes. Activity data may enclose unexpected and interesting knowledge about the optimum decision making and processes which may result in low risk of negative impact. Therefore, it is significant to study activity patterns and impact-targeted activity behavior.

* This work is sponsored by Australian Research Council Discovery Grant (DP0667060), China Overseas Outstanding Talent Research Program of Chinese Academy of Sciences (06S3011S01), and UTS ECRG and Chancellor grants.

Activity data embodies organizational, information and application constraints, and impact-targeted multi-dimensional complexities which combine those from temporal, spatial, syntactic and semantic perspectives. As a result, activity data presents special structure and semantic complexities. For instance, variant characteristics such as sequential, concurrent and causal relationships may exist between activities. Activity data usually presents unbalanced distribution. As a result, many existing techniques cannot be used directly, which rarely cares for the impact of mined objects.

Therefore, new data mining methodology and techniques need to be developed to preprocess and explore activity data. This leads to *activity mining*. This paper discusses the challenges and prospects in building up effective methodologies and techniques to mine interesting activity patterns. *Activity mining* aims to discover rare but significant impact-targeted activity patterns in unbalanced activity data, such as frequent activity patterns, sequential activity patterns, impact-oriented activity patterns, impact-contrasted activity patterns, and impact-reversed activity patterns. The identified activity patterns may inform risk-based decision making in terms of predicting and preventing the happenings of targeted activity impact, maintaining business goals, and optimizing business rules and processes, etc.

The remainder of this paper is organized as follows. Section 2 presents the scenario and characteristics of activity transactional data and its challenges to the existing KDD. Section 3 discusses possible activity mining methodologies. Activity mining tasks are discussed in Section 4. Finally, Section 5 concludes this paper.

2 Activity Data

This section introduces an example and the characteristics of activity data, and further build up an activity model representing and defining activity data. Challenges of activity data on the existing KDD are discussed further.

2.1 An Example

Here we illustrate the activities in social security network. In the process of delivering Government social security service to the population, large volumes of customers contact governmental service agencies [3]. For instance, over 6 million Australians access Centrelink's services at one point in time. Every single contact, e.g., a circumstance change, may trigger a sequence of activities running serially or in parallel. Among them, some are associated with fraudulent actions and result in government customer debt. For example, Table 1 lists an excerpt of activity transactions [2] relevant to a scenario of changing customer address in Centrelink. When a Newstart (NSA) benefit recipient i reports his/her circumstance $C_{i,1}$ to Centrelink, an officer conducts activity $A_{i,2}$ and $A_{i,3}$ to check and update i 's entitlement and details. In parallel, the officer also conduct activity $A_{pi,3}$ and consequently activities $A_{pi,4}$ and $A_{pi,5}$ to inspect i 's partner j 's details and possible debts. Concurrently, customer i 's task $T_{i,1,1}$ triggers $A_{k,2}$ on i 's NSA co-tenant k . $A_{k,2}$ further triggers $A_{k,3}$, $A_{k,4}$ and $A_{k,5}$ on k to reassess and update k 's rent details and possible debts.

Table 1. Activity transactional data

Customer $i = \text{Newstart (NSA) recipient}^1$	Partner j^2	Customer $k = \text{NSA recipient}$
Circumstance $C_{i,1} = \text{Change of Address}$		
$A_{i,2} = \text{Accelerated Client Matching System review (triggered by } T_{i,1,1})$		$A_{k,2} = \text{Accelerated Client Matching System review (by } T_{i,1,1} \text{ on customer } j)$
$T_{i,2,1} = \text{Letter to Customer}$		$T_{k,2,1} = \text{Letter to Customer}$
$A_{i,3} = \text{Customer Details Update}$	Parallel Activity $A_{pi,3} = \text{Customer Details Update}$	$A_{k,3} = \text{Customer Details Update}$
$T_{i,3,1} = \text{Change Rent Details};$	$T_{pi,3,1} = \text{Change Rent Details};$	$T_{k,3,1} = \text{Change Rent Details};$
$T_{i,3,2} = \text{Change Home-ownership Details.}$	$T_{pi,3,2} = \text{Change Home-ownership Details.}$	$T_{k,3,2} = \text{Change Home-ownership Details.}$
$T_{i,3,3} = \text{NSA Reassessment}$	$T_{pi,3,3} = \text{PPP Reassessment}$	$T_{k,3,3} = \text{NSA Reassessment}$
$E_{i,3,1}$ (from $T_{i,3,1}$ & $T_{i,3,2}$) = NSA Reassessment	$T_{pi,3,4} = \text{FTB Reassessment}$	$E_{k,3,1}$ (from $T_{k,3,1}$ & $T_{k,3,2}$) = NSA Reassessment
	$E_{pi,3,1}$ (from $T_{pi,3,1}$) = FTB Reassessment	$E_{k,3,2}$ (from $T_{k,3,1}$) = Transfer NSA rate variation data to Debt Management System
	$E_{pi,3,2}$ (from $T_{pi,3,1}$) = Transfer FTB rate variation data to Debt Management System	
	$A_{pi,4}$ (from $T_{pi,3,1}$ of $A_{pi,3}$) = New Debt	$A_{k,4}$ (from $T_{k,3,1}$ of $A_{k,3}$) = New Debt
	$T_{pi,4,1} = \text{Raise debt}$	$T_{k,4,1} = \text{Raise debt}$
	$E_{pi,4,1} = \text{Profiling Assessment}$	$E_{k,4,1} = \text{Profiling Assessment}$
	$A_{pi,5}$ (from $T_{pi,4,1}$ of $A_{pi,4}$) = Debt	$A_{k,5}$ (from $T_{k,4,1}$ of $A_{k,4}$) = Debt
	$T_{pi,5,1} = \text{Withholdings}$	$T_{k,5,1} = \text{Withholdings}$

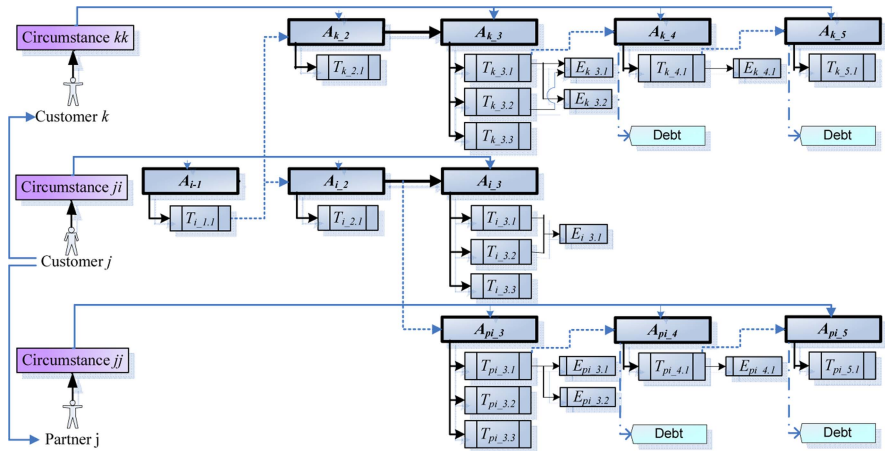


Fig. 1. Activity scenario diagram

¹ **Note:** In this case Rent Assistance will be paid as part of the partner’s FTB in Centrelink business.

² = Parenting Payment – Partnered (PPP) & Family Tax Benefit (FTB) recipient.

In this example, an activity may further trigger one to many tasks. A task may trigger another activity on the same or different customer or some follow-up events (e.g., $E_{i,3,1}$ from tasks $T_{i,3,1}$ & $T_{i,3,2}$). With respect to time frame, parallel or serial activities may run dependently or independently. For instance, parallel activity $A_{pi,3}$ depends on $A_{i,3}$ while parallel activity $A_{pi,4}$ and concurrent activity $A_{k,5}$ run independently even though the activities on customer i are completed. Another interesting point is that some activities may generate impacts on business outcomes such as raising debts (e.g., $A_{pi,5}$ and $A_{k,5}$). Such debt-oriented activities are worthy of further identification so that debt can be better prevented and predicted.

2.2 Activity Model

The term “Activity” is an informative entity embracing both business and technical meanings. In business situations, an activity is a business unit of work. It corresponds to one to many activity operators to conduct certain business arrangement forming a workflow or process. It directly or indirectly satisfies certain organizational constraints and business rules. Technically, an activity refers to one to several transactions recording information related to a business unit of work. Therefore, an activity may undertake certain business actions, embody business processes, and trigger some impact on business. Moreover, activity transactions embed much richer information about business environment, causes, effects and dynamics of activities and potential impact on business, as well as hidden information about the dynamics and impact of activities on debts and activity operator circumstances. In general, an activity records information about *who* (maybe multiple operators) processes *what types* of activities (say change of address) from *where* (say customer service centres) and for *what reasons* (say the action of receipt of source documents) at *what time* (date and time), as well as resulting in *what outcomes* (say raising or recovering debt).

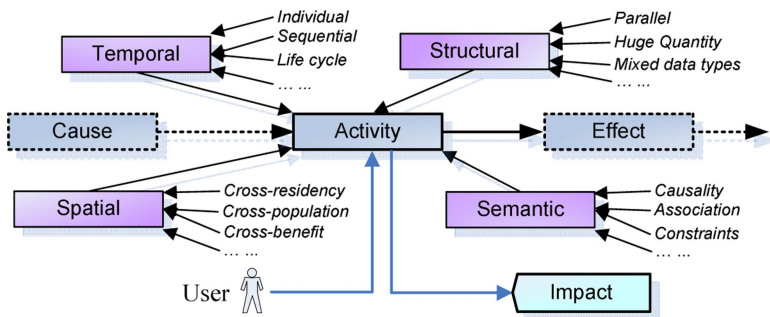


Fig. 2. Activity model

Based on the understanding of the structural and semantic relationships existing in activity transactions, we generate an abstract *activity model* as shown in Figure 2. An activity is a multi-element entity $A = (C, E, U, I, F)$, where C and E are *cause* triggering and *effect* triggered by the activity A , respectively. An activity either is operated by or act on one or multiple *users* U . It may directly or indirectly generate

impact I on business outcomes such as leading to debt or costing. In particular, an activity and its sequence present complex features in terms of *temporal*, *spatial*, *structural* and *semantic* dimensions. For each of the four dimensions, activity presents various observations which make activity mining very complicated. For instance, each activity has a life cycle starting from registration by a user and ending via completion on the same day or some time later on. During its evolution period, it may be triggered, restarted, held, frozen, deleted or amended for some reason.

2.3 Challenges

Activity data proposes the following challenges to existing KDD approaches.

- Activities of interest to business needs are *impact-oriented*. Impact-oriented activities refer to those directly or indirectly lead to or are associated with certain impact on business situations, say fraudulent social security activities resulting in government customer debt. Therefore, *activity mining aims to discover specific activities of high or low risk associated with business impact*, which we call *impact-targeted activity pattern mining*. While the existing KDD research rarely deal with impact-targeted activity pattern mining.
- Impact-oriented activities are usually a very small portion of the whole activity population. For instance, fraudulent activities in Centrelink only account for 4% of all activities. This leads to an *unbalanced class distribution* [15] of activity data, which means positive target-related activity class is only a very small fraction of the whole data set. Unbalanced class distribution of activity data proposes challenges to impact-targeted activity pattern mining in aspects such as activity sequence construction, pattern mining algorithms and interestingness design and evaluation.
- Among activities, some occur more often than others. This indicates an *unbalanced item distribution* of activity item set. Unbalanced item distribution also affects activity sequence construction, pattern mining algorithms and interestingness evaluation.
- In analyzing impact-targeted activities, *positive* and *negative* impact-targeted activity patterns can be considered, which correspond to positive and negative activity patterns. Other forms of activity patterns include *sequential activity patterns*, activity patterns representing the contrast of impact (called *contrast activity patterns*) and the reversal of impact (named *reverse activity patterns*), etc.
- In constructing, modeling and evaluating activity patterns, *constraints* from aspects such as targeted impact, distributed data sources and business rules must be considered. Real-world *constraint-based* activity pattern mining is more or less domain-driven [1]. Constraint-based mining and domain-driven data mining should be taken into account in mining impact-targeted activity patterns.
- Activities present *spatial-temporal* features such as sequential, parallel, iterative and cyclic aspects, as well as crossing benefits, residencies and regions. For instance, an activity may trigger one to many serial or parallel tasks and certain corresponding events, and generate complex action sequences.

The complexities of activity data differentiate it from normal data sets such as those in event [4], process [13] and workflow [5] mining, where data is much flat and

simple. Those mainly studies process modeling and has nothing to do with complex activity structure and business impacts of activities. Therefore, new methodologies, techniques and algorithms must be developed.

3 Activity Mining Methodologies

3.1 Activity Mining Framework

In developing activity mining methodologies, we first focus on understanding activity data and designing a framework for activity mining.

In business world, activities are driven by or associated with business rules [2]. For instance, the activity sequences triggered by changing address (see Figure 1) present interesting internal structure. Activity A_{i-1} triggers A_{i-2} and A_{k-2} in parallel, while two series of activities A_k and A_{pi} go ahead after the completion of original activity sequence A_i . This example shows that there may exist meta-patterns in activity transactions, which are helpful for further understanding and supervising activity pattern mining. For instance, fundamental activity meta-patterns such as *serial* ($x \rightarrow y$), *parallel* ($x \parallel y$), *cyclic* ($x \rightarrow x$ or $x \rightarrow z \rightarrow x$) and *causal* ($x \Rightarrow z$) may exist between activities x , y and z . These meta-patterns if identified can supervise further activity pattern learning. Temporal logic-based ontology specifications can be developed to represent and transform activity metapatterns.

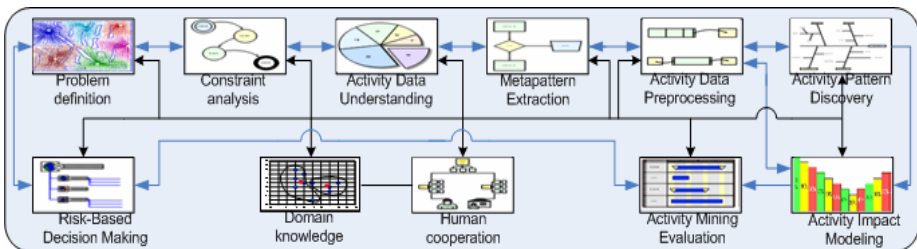


Fig. 3. An activity transaction mining framework

Based on the above activity data understanding, we can study a proper framework for activity mining. Figure 3 illustrates a high-level process of activity mining. It starts from understanding activity constraints, data and meta-patterns. The results are used for preprocessing activity data by developing activity preparation techniques. Then we design effective techniques and algorithms to discover interesting activity processing patterns and model activity impacts on debts. Further work is to evaluate the performance of activity analysis. Finally, we integrate the above results into an activity mining system, and deploy them into strengthening risk-based decision making in debt prevention. It is worth noting that there may be back and forth among some of the above steps. Additionally, domain analysts and knowledge are essential for iterative refinement and improving the workable capability of mining results.

3.2 Activity Mining Approaches

Due to the closely coupled relationship between activities, activity users and impact on business, we need to combine them to undertake systematic analysis of activity data. This is different from normal data mining which usually only focus on some aspect of the problem, e.g., process mining focuses on business event and workflow analysis. Driven by business rules and impact, we can highlight some key aspect and undertake activity mining in terms of *activity-centric analysis*, *impact-centric analysis* and *customer-centric analysis*. We introduce them individually with regard to the example of analysing activities in governmental customer debt prevention.

Activity-centric analysis. Activity-centric analysis focuses on analysing activity patterns, namely the relationships between activities. For instance, we can conduct activity centric debt modelling in terms of the following aspects: (1) Pattern analyses of activities which have or haven not led to debts, (2) Activity process modelling, and (3) Activity monitoring.

Impact-centric analysis. Impact-centric analysis attempts to analyse the impact of activities and activity sequences on business such as debts, as well as optimizes activities and processes to reduce the negative impact of activities/processes on business. The major research includes (1) Analysing the impacts of a type of notifiable events or a class of relevant activity sequence against debt outcomes, (2) Risk/cost modelling of activities which may or may not lead to debts, and (3) Activity/process optimization.

Customer-centric analysis. Customer-centric analysis studies the patterns of activity operators' circumstances and circumstance changes which may or may not lead to debt in aspects such as (1) Circumstance profiling, (2) Officer behaviour analysis and (3) Customer behaviour analysis.

We further discuss these approaches by illustrating potential business problems in governmental customer debt prevention in Section 4.

4 Activity Mining Tasks

The major challenge of mining activity transactions come from the following processes in mining activity transactions: (1) activity preprocessing, (2) activity pattern mining, (3) activity impact modeling, and (4) activity mining evaluation.

4.1 Activity Preprocessing

The characteristics of activity transactional data make activity preprocessing very essential and challenging. The tasks include developing proper techniques to (1) improve data quality, (2) handle mixed data types, (3) deal with unbalanced data, (4) perform activity aggregation and sequence construction, etc.

Unbalanced data. As shown in Figure 4, activity data presents unbalanced class distribution (e.g., the whole set $|A|$ is divided into $|S|$ as debt-related activity set while $|\bar{S}|$ as non-debt set) and unbalanced item distribution. Unbalanced data mainly affect the performance and evaluation of traditional KDD approaches. Therefore, in activity preprocessing, effective methods and strategies must be considered to balance the affection of data imbalance. For balancing the impact of unbalanced class distribution,

techniques such as equal sampling in separated data sets, redefining interestingness measures such as replacing global support with local support in individual sets can be used. With respect to the imbalance of activity items, domain knowledge and domain experts must be involved to determine what strategies should be taken to balance the impact of some high proportional items. Their impact may be balanced by deleting or aggregating some items and designing interestingness measures.

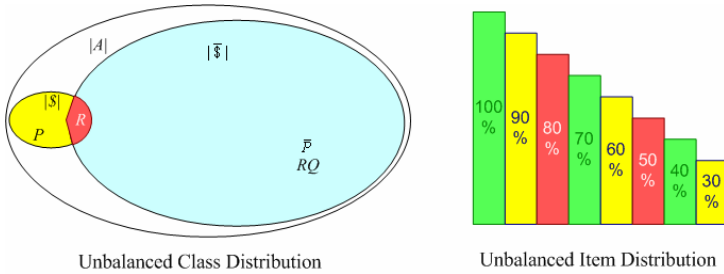


Fig. 4. Unbalanced activity data

Activity sequence construction. It is challenging to construct reliable activity sequences. The performance of activity sequences greatly affects the performance of activity modeling and evaluation. Different sliding window strategies can be used and correspondingly generate varied activity sequences. For instance, the activity series in Figure 5 could be constructed or rewritten into varied activity sequences, say the sequence for d_2 -related activities could be $S_1: \{a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, d_2\}$, $S_2: \{a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, d_2\}$, $S_3: \{a_{11}, a_{12}, a_{13}, d_2, a_{14}, a_{15}\}$, etc. The design of sliding window strategies must be based on domain problems, business rules and discussion with domain experts. S_1 considers a fixed window, S_2 may cover the whole debt period, while S_3 account for the further effect of d_2 on activities. Domain knowledge plays an important role in determining which one of the three strategies makes sense.

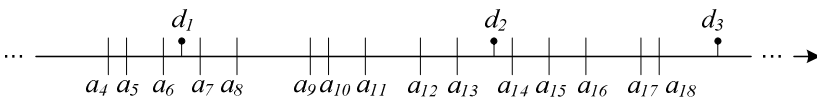


Fig. 5. Constructing activity sequences

4.2 Activity Pattern Mining

Impact-targeted activities are usually mixed with customer circumstances and business impact. Therefore, activity pattern mining is a process of mining interesting activity processing and user behavior patterns based on different focuses such as *activity-centric*, *impact-centric* and *customer-centric* analyses. As shown in Table 2, activity pattern mining aims to identify *risk factors* and *risk groups* highly or seldom related to concerned business impact by linking activity, impact and customer files together.

Table 2. Impact-targeted activity pattern mining

Risk level	Risk factor		Risk group	
High	Activity features	Customer circumstances	Activity processing patterns	Customer behavior patterns
Low				

Impact-targeted risk factors include major activity features and customer circumstances at high or low risk of leading to or being related to targeted business impact. For instance, in social security network, the activity “reassessing benefit” is found highly correlated with leading to debt. Impact-targeted risk groups target identifying activity processing patterns or customer behavior patterns more or less resulting in targeted impact. For example, frequent activity sequences are likely associated with government customer debt. In the following, we illustrate some novel impact-targeted activity patterns.

Customer-centric activity analysis mainly investigates user decision-making behavior and profiling as well as the impact of a user’s or a class of users’ actions on related stakeholders. This identifies officer/customer’s demographics and profiling leading to debts, e.g., studying the impact of staff proactive actions on debt compared with passive and customer-triggered activities, or the impact of face-to-face dealings vs. technology-based contacts. We can develop classification methods for debt-related customer segmentation. Classification methods based on logistic regression tree and temporal decision tree can be studied via considering temporal factors in learning debt/no-debt, low-debt/high-debt and debt reason patterns. The results of frequent, sequential and causal activity patterns can benefit the analysis of customer demographics and circumstances leading to debts.

Activity-centric analysis focuses on inspecting relations between debt-related activities. This includes mining activity patterns such as frequent, sequential [6] and causal ones in constrained scenarios. To mine frequent patterns, association rule method can be expanded to discover temporally associated [14] activities by recoding activity records and developing new measures and negative association rules. Those frequent activity patterns can be identified leading to debt, no debt or debt completion. Negative associations such as “if activity *a* and *b* but not *c* then debt” can be studied. Further, based on the constructed sequences of activities, sequential activity patterns in or crossing activity sequences can be investigated by considering temporal relations between activities. We can test various sequence combinations based on different sliding window strategies, and incorporate the identified meta-patterns and frequent patterns into sequential activity mining. In addition, there exists certain causal relation [9] in activity sequences. Causal pattern mining aims to find and explain contiguity relations between activities and between activities and debt reasons/state changes. Determinant underlying causal pattern, relational casual pattern and probabilistic causal patterns can be analyzed by considering aspects like activity forming, contiguity and interaction between activities and spatial-temporal features of activities and debt-related activity sequences.

Table 3. Impact-targeted activity patterns

Activity pattern		Explanation
Frequent activity patterns	Positive associations	Activity associations P related to impact $\$: P \rightarrow \$$
	Negative associations	P related to non-impact $\bar{\$}: P \rightarrow \bar{\$}$
	Positive sequences	Activity sequences P related to impact $\$: P \rightarrow \$$
	Negative sequences	Activity sequences P related to non-impact $\bar{\$}: P \rightarrow \bar{\$}$
Contrast activity patterns	Contrast associations	P related to impact $\$: P \rightarrow \$$ in impact data set; P also associated with non-impact $\bar{\$}: P \rightarrow \bar{\$}$ in non-impact data set
	Contrast sequences	
Reverse activity patterns	Reverse associations	P related to impact $\$: P \rightarrow \$$ in impact data set, while $\{P, Q\}$ associated with non-impact $\bar{\$}: P, Q \rightarrow \bar{\$}$ in non-impact data set
	Reverse sequences	

4.3 Activity Mining Evaluation

It is essential to specify proper mechanisms [11] for evaluating the workable capability of identified activity patterns and risk models. Technically, we implement *impact-centric mining* which develops interestingness measures *tech_int()* in terms of particular activity mining methods. The debt preventable capability of the identified findings can also be assessed by checking the existing administrative/legal business rules and domain experts.

For technical evaluation, the existing interestingness can be testified and expanded or new measures may be designed to satisfy activity mining demand. In pilot analysis, the measures of some existing KDD methods are found invalid when deployed into activity mining, e.g., *support* may be too low to measure frequency. For newly developed activity mining methods, we can design specific measures by considering technical factors such as activity statistics, debt ratios and customer circumstance changes. On the other hand, the identified patterns and models can also be examined in terms of rigor and relevance to business factors such as business goals, significance, efficiency, risk to debts and cost-effectiveness. Additionally, measures themselves need be evaluated in terms of interpretability and actionable capability.

Further evaluation may be necessary by using significance test, cross-validation and ensemble from both business and technical perspectives. Under certain condition, it is useful to present an overall measurement of identified patterns by integrating interest from both technical and business perspectives. To this end, fuzzy set-based fuzzy aggregation and ranking may be useful to generate overall examination of the identified patterns and models. In addition, multiple measures may apply to one method. We can aggregate these various concerns to create an integrated measure for global assessment.

4.4 Matching List Between Activity Analysis Goals and Business Problems

The following Table 4 further explains them by illustrates some relevant business problems through observing the example of governmental customer debt prevention.

Table 4. Activity mining tasks

	Activity analysis goals	Business problems
Activity pre-processing	(1) Activity data quality	How to identify wrongly coded activities and debts led by them?
	(2) Mixed data types	How to systematically analyze data mixing continuous, categorical and qualitative types?
	(3) Activity aggregation & sequence construction	How to aggregate sequence A_i with its partnered one A_{pi} into an integrated sequence in Fig. 1?
Activity-centric analysis	(4) Activity meta-pattern analysis (e.g., parallel, causal, cyclic metapatterns)	Can we find relations such as $x \rightarrow y$, $x \parallel y$, $x \rightarrow x$ or $x \rightarrow z \rightarrow x$ and $x \Rightarrow z$ between activities x , y and z ?
	(5) Activity pattern analysis (e.g., frequent, sequential, causal patterns)	Can we find rules like “if activity A_1 then A_2 but no A_3 in the following 3 weeks \rightarrow debt”? or “if A_1 triggers A_2 , A_2 triggers $A_3 \rightarrow$ no debt”?
	(6) Activity process simulation and modelling (e.g., reconstruct processes)	Can we reconstruct some processes (activity series) based on activity transactions which may or may not lead to debts?
	(7) Activity replay and monitoring (e.g., generating recommendation or alerts)	Can we find knowledge like “If the activity A_3 is triggered, then generating an alert to remind the likely risk of this activity?”
Impact-centric analysis	(8) Activity impact analysis (e.g., the impact of an activity sequence on debt)	Is there correlation between activity types and debt types indicting what activity types more likely lead to certain types of debts?
	(9) Risk/cost modeling of activities (e.g., leading to debt or operational costs)	To what extent a certain activity/activity type/activity class will lead to certain type of debt?
	(10) Activity or process optimization	If activity A_3 is triggered, then recommending activity A_6 rather than A_4 then A_5 , which will lead to low/no debts or the ending of debt?
Customer-centric analysis	(11) Operator circumstance profiling	What are the demographics of those customers who more likely lead to debt?
	(12) Officer behavior analysis	What are the impacts of those activities triggered by staff proactively compared with other passive activities and customer-triggered activities?
	(13) Customer behavior analysis	Whether face-to-face dealings lead to low debts compared with technology-based contacts such as by Internet/email?
Activity mining evaluation	(14) Technical objective & subjective measures	Are the existing technical objective measures ok when they are deployed to mine activity data?
	(15) Business objective & subjective measures	How to evaluate the impact of activity patterns on business?
	(16) Integrated evaluation of activity patterns	If technical interestingness clashes with business one, how to assess them?

5 Conclusions and Future Work

Rare but significant impact-targeted activities differentiate from traditional mining objects in aspects such as closely targeting certain business impact. For instance, a series of dispersed terrorist activities may finally lead to a serious disaster to our society. Impact-targeted activity data presents special structure complexities such as

unbalanced class and item distribution. Mining rare but significant impact-targeted activity patterns in unbalanced data is very challenging. This paper analyzes the challenges and prospects of activity mining. We present an example to illustrate the complexities of activity data, and summarize possible impact-targeted activity pattern mining methodologies and tasks based on our practice in identifying fraudulent social security activities associated with government customer debt. In practice, activity mining can play an important role in many applications and business problems such as counter-terrorism, national and homeland security, distributed fraudulent and criminal mining. Techniques coming from impact-targeted activity mining can prevent disastrous events or improve business decision making and processes.

Acknowledgement

Thanks are given to Dr Jie Chen, Mr Yanchang Zhao and Prof Chengqi Zhang at UTS for their technical discussion, as well as to Ms Yvonne Borrow, Mr Peter Newbigin and Mr Rick Schurmann at Centrelink Australia for their domain knowledge.

References

1. Cao L, Zhang C. Domain-driven data mining: a practical methodology, *Int. J. of Data Warehousing and Mining*, 2006.
2. Centrelink. *Integrated activity management developer guide*, 1999.
3. Centrelink. *Centrelink annual report 2004-05*.
4. Guralnik V, Srivastava J. Event Detection from Time Series Data, *KDD-99*, 33-42.
5. Hammori M, Herbst J, Kleiner N. Interactive workflow mining—requirements, concepts and implementation, *Data & Knowledge Engineering*, 56 (2006) 41–63.
6. Han J., Pei J. and Yan X. Sequential Pattern Mining by Pattern-Growth: Principles and Extensions, in *Recent Advances in Data Mining and Granular Computing*, Springer Verlag, 2005.
7. Mena, J. *Investigative Data Mining for Security and Criminal Detection*, First Edition, Butterworth-Heinemann, 2003.
8. National Research Council, *Making the Nation Safer: The Role of Science and Technology in Countering Terrorism*, Nat'l Academy Press, 2002.
9. Pazzani M. A Computational Theory of Learning Causal Relationships, *Cognitive Science*, 15:401-424 1991.
10. Potts, W. *Survival Data Mining: Modeling Customer Event Histories*, 2006
11. Silberschatz A, Tuzhilin A. What makes patterns interesting in knowledge discovery systems, *IEEE TKDE*, 8(6):970-974, 1996.
12. Skop, M. Survival analysis and event history analysis. © Michal Škop, 2005.
13. Van der Aalst W.M.P., Weijters A.J.M.M. Process mining: a research agenda, *Computers in Industry*, 53 (2004) 231–244.
14. Williams G, et al. Temporal Event Mining of Linked Medical Claims Data. *PAKDD03*.
15. Zhang, J., Bloedorn, E.; Rosen, L.; Venese, D. **Learning rules from highly unbalanced data sets**, *2004 ICDM Proceedings*, pp571 – 574.

Finding the Optimal Cardinality Value for Information Bottleneck Method

Gang Li¹, Dong Liu², Yiqing Tu¹, and Yangdong Ye²

¹ School of Information Technology, Deakin University,
221 Burwood Highway, Vic 3125, Australia

² School of Information Engineering, Zhengzhou University, Zhengzhou, China
gang.li@deakin.edu.au, ielsld1@gs.zzu.edu.cn, ytu@deakin.edu.au,
yeyd@zzu.edu.cn

Abstract. *Information Bottleneck* method can be used as a dimensionality reduction approach by grouping “similar” features together [1]. In application, a natural question is how many “features groups” will be appropriate. The dependency on prior knowledge restricts the applications of many *Information Bottleneck* algorithms. In this paper we alleviate this dependency by formulating the parameter determination as a model selection problem, and solve it using the minimum message length principle. An efficient encoding scheme is designed to describe the information bottleneck solutions and the original data, then the minimum message length principle is incorporated to automatically determine the optimal cardinality value. Empirical results in the documentation clustering scenario indicates that the proposed method works well for the determination of the optimal parameter value for information bottleneck method.

1 Introduction

In many data mining tasks the data is represented by a large number of features, among which maybe only some features are relevant for the data mining task. The presence of large number of redundant, similar or weakly relevant features can usually lead machine learning algorithms to become computationally intractable, and this usually attributed to “*the curse of dimensionality*”. Therefore the task of choosing a small subset of features or some “feature groups” that capture the relevant properties of the data, is crucial for efficient learning.

A number of methods have been proposed for dimensionality reduction in the literature. Recently, the *Information Bottleneck* method has been applied for dimensionality reduction by grouping “similar” features together [1]. It is argued that Shannon’s theory provides a compelling mechanism for quantifying the fundamental tradeoff between complexity and accuracy, by unifying the source and channel coding theorems into one principle called the “*Information Bottleneck*” (IB), and this unified view of the coding theorems sheds new light on the question of “relevant data representation” and suggests new algorithms for extracting such representations from co-occurrence statistics [1]. In the past several years, the *Information Bottleneck* method has been successfully employed in a wide

range of applications including image classification [2,3], document categorization [4,5,6], analysis of neural codes [7], text classification [8], gene expression analysis [9], etc.

One crucial stage in *Information Bottleneck* method is to determine the appropriate number of groups. Most existing approaches to this problem are based on a pre-assumed user knowledge, as in [10]. However, in the real world applications, it is infeasible to assume those kinds of prior knowledge. It would be attractive if an IB algorithm has a build-in mechanism which could automatically determine this parameter. This is in general a question of model selection. In this paper, we attempt to solve this problem using the “*Minimum Message Length*” (MML) principle, and propose an algorithm to automatically determine the optimal IB parameter.

The paper is organized as follows: in section 2, we briefly recap the *Information Bottleneck* method, and In Section 3, we formulate our model selection methods from the Minimum Message Length principle, and introduce the algorithm to determine an appropriate parameter for IB method. In Section 4, we evaluate our method on existing benchmark data sets. Finally, we conclude in Section 5.

2 Background

The *Information Bottleneck* method originated from Shannon’s information theory [11], and it aims to extract a compact, yet meaningful representation T of a much larger original data D , which preserves high mutual information about a target variable Y . Throughout this paper, we will use the following notations: y refers to relevant variables such as words, x refers to data instances such as documents, and t refers to the compact representation such as document categories. Upper case letters X , Y and T are used to denote the corresponding random variables.

2.1 Information Bottleneck Principle

Consider a simple coding scheme for a random variable X that follows a distribution $p(x)$. The encoding scheme involves representing the random variable X by a compressed variable T . Given a distortion function $d(x, t)$, we want to encode the values of X such that the average error is less than a given number \mathcal{D} . Shannon’s rate distortion theorem states that the infimum of all rates (the logarithm of $|T|$) under a given constraint on the average distortion \mathcal{D} is given by the following function:

$$R(\mathcal{D}) = \min_{\{p(t|x): E(d(x,t)) \leq \mathcal{D}\}} I(X; T) \quad (1)$$

where the $I(X; T)$ is the mutual information between X and T , and the $E(d(x, t))$ is the expected distortion induced by $p(t|x)$.

Unlike the rate distortion theory, the IB method avoids the arbitrary choice of a distortion function $d(x, t)$ [1,3]. The motivation comes from the fact that in

many cases, defining a “target” variable Y with respect to X is much simpler than defining a distortion function. Given a joint probability distribution $p(x, y)$ on variables X and Y , IB methods considers the following distortion function

$$d(x, t) = D_{KL}(p(y|x)||p(y|t)) \tag{2}$$

where $D_{KL}(f||g)$ is the Kullback-Leibler divergence between distributions $f(\cdot)$ and $g(\cdot)$. What is interesting is that the $p(y|t) = \sum_x p(y|x)p(x|t)$ is the function of $p(t|x)$. Hence, the distortion function $d(x, t)$ is not predetermined here, instead as it searches for the best representation $p(t|x)$ it also searches for the most suitable distortion function $d(x, t)$.

As the expected distortion $E(d(x, t))$ can be written as

$$\begin{aligned} E(d(x, t)) &= E[D_{KL}(p(y|x)||p(y|t))] \\ &= \sum_{x,t} p(x, t) \sum_y p(y|x) \log \frac{p(y|x)}{p(y|t)} \\ &= \sum_{x,t,y} p(x, t, y) \log \frac{p(y|x)}{p(y|t)} \\ &= \sum_{x,t,y} p(x, t, y) \log \frac{p(x,y)}{p(x)} - \sum_{x,t,y} p(x, t, y) \log \frac{p(y|t)}{p(y)} \\ &= I(X; Y) - I(T; Y) \end{aligned} \tag{3}$$

Substituting formula (3) in the rate distortion function (1) we can get:

$$R(\mathcal{D}) = \min_{\{p(t|x): I(X;Y) - I(T;Y) \leq \mathcal{D}\}} I(X; T) \tag{4}$$

in which $I(X; Y)$ is a constant, the rate distortion function is usually written as

$$R(\mathcal{D}) = \min_{\{p(t|x): I(T;Y) \geq \mathcal{D}\}} I(X; T) \tag{5}$$

The equation shows that IB method minimizes $I(X; T)$ when ensuring $I(T; Y)$ is no less than an infimum \mathcal{D} . In a sense, this objective function implements a “bottleneck” for the dependency between X and Y through T , i.e., one is trying to squeeze the information X provides about Y through a compact “bottleneck” formed by the compressed representation T . The objective of IB method is then to *minimize*

$$I(X; T) - \alpha I(T; Y) \tag{6}$$

which can be considered as a compromise between two mutual information. If we multiply equation (6) with $-\frac{1}{\alpha}$, we can get the dual optimization function which maximizes

$$I(T; Y) - \beta I(T; X) \tag{7}$$

for some prespecified $\beta > 0$, which balances the tradeoff between the compression and preservation.

2.2 Information Bottleneck Algorithms

Without any assumption on the joint distribution $p(x, y)$, Tishby et al showed that the IB optimization function has an exact optimal solution as follows [1]:

$$p(t|x) = \frac{p(t)}{Z(x, \beta)} \exp(-\beta D_{KL}(p(y|x)||p(y|t)))$$

$$p(y|t) = \frac{1}{p(t)} \sum_x p(y|x)p(t|x)p(x)$$

$$p(t) = \sum_x p(t|x)p(x)$$

where $Z(x, \beta)$ is a normalization function.

However, how to construct the optimal solution remains an NP-hard problem, and several algorithms were developed to approximate the optimal solution (see [12] for detailed review and comparison). In general, any IB algorithm will need two parameters:

- one is the tradeoff parameter β ;
- the other one is the cardinality value of the compressed representation T . In applications, this is also referred to as the number of “feature groups”, etc.

For applications in which the motivation is to group the data, such as image clustering, document clustering, and the general dimensionality reduction, it is tempting to set β to be ∞ . In this case, the cost equation (7) becomes the maximization of $I(T; Y)$, and the optimal IB solution tends to become crisp, i.e., the $p(t|x)$ approaches zero or one almost everywhere. If the cardinality value of T is given, there exist several information bottleneck algorithms which aim to produce this crisp representation T , as shown in [12] and [13].

Accordingly in these IB applications the user still need to provide the value for the second parameter, although the prior knowledge on the cardinality value of T is usually unavailable. This problem has restricted the application of IB method. It will be attractive if an automatical mechanism can be incorporated into IB algorithm to determine the appropriate cardinality value for T . In general, this is a *model selection* problem, and we are going to approach it using the minimum message length principle

3 Choosing Optimal Cardinality Value Using MML Principle

A fundamental problem in model selection is the evaluation of different models for fitting data. In 1987, Wallace and Freeman [14] extended their earlier method [15] for model selection based on *Minimum Message Length* (MML) coding. The basic idea of the MML principle is to find the hypothesis H which leads to the shortest message. Its motivation is that this method can automatically embrace the selection of an hypothesis of appropriate complexity as well as leading to good estimates of any free parameters of the hypothesis.

In the case of *determining the optimal cardinality value* for IB method, it requires the estimate of the optimal encoding code for the description of the IB solution and the original data D . The whole message consists of 2 parts:

1. The message costed to describe the IB solution. This part of message can be further divided into two sub-messages.

- (a) message encoding the cardinality value k .
 - (b) message encoding the IB solution T_k .
2. The message costed to describe the original data set D , under the assumption that the IB solution was the optimal one.

According to the MML principle, the shorter the encoding message length is, the better the corresponding model is. From *Information Theory*, the total message length can be approximated using the following formula¹,

$$MsgLen = MsgLen(k) + MsgLen(T_k) + MsgLen(D|k, T_k) \tag{8}$$

Where k is the cardinality value of the representation T , T_k is IB solution when the cardinality value of T is set to k .

Let us suppose that we are interested in encoding the data set D , and would like to do so as efficiently as possible. We can also assume that the IB solution corresponding to k consists of k groups of objects², with none of them empty. With these assumptions, we can design an encoding scheme as in the following sections.

3.1 Encoding IB Model

Given a cardinality value k , the IB solution returned from any IB algorithm (aIB or sIB) contains k groups of objects. The encoding of this model shall consist of the following two terms:

1. Encoding of the cardinality value k : The encoding length of this description can be estimated as $\log^*(k)$, where \log^* is the universal code length for integers [16]. When the range of values for k is not known as prior knowledge, the universal code length is the optimal length, and it can be shown that

$$msgLen(k) = \log^*(k) \approx \log_2(k) + \log_2 \log_2(k) + \dots$$

2. Encoding of the IB solution T_k : The encoding of T_k is an important part for whole encoding scheme. Suppose the number of objects in each group are known, and we sort them such that

$$m_1 \geq m_2 \geq \dots \geq m_k \geq 1$$

Accordingly, the encoding of T_k can be further divided into:

- the encoding of m_1, m_2, \dots, m_k : We can compute

$$\bar{m}_i = \sum_{t=i}^k m_t - k + i \quad \text{with } i = 1, \dots, k - 1$$

¹ For convenience, we use 2-based logarithm through the paper, and calculate all message length in *bits*. If the natural logarithm is used, the length will be calculated in *nits*.

² Here, the ‘object’ can be feature or instance, depending on the scenarios of IB applications.

then the encoding length of m_1, m_2, \dots, m_k can be estimated by

$$msgLen(m_1, \dots, m_k) = \sum_{i=1}^{k-1} \log \bar{m}_i$$

- the encoding of the partition which assigns n objects into k groups, with the number objects in each group to be m_1, m_2, \dots, m_k . As $\sum_{i=1}^k m_k = n$, and we then have the encoding length

$$\begin{aligned} msgLen(T_k|m_1, \dots, m_k) &= \log \frac{n!}{m_1! \dots m_k!} \\ &= \log n! - \sum_{i=1}^k \log m_i! \\ &\approx n \log n - \sum_{i=1}^k m_i \log m_i \\ &= -n \sum_{i=1}^k \frac{m_i}{n} \log \frac{m_i}{n} \end{aligned} \tag{9}$$

For a candidate cardinality value k , the encoding length of k and the T_k can then be estimated as:

$$MsgLen(k) + MsgLen(T_k) = \log^*(k) + \sum_{i=1}^{k-1} \log \bar{m}_i - n \sum_{i=1}^k \frac{m_i}{n} \log \frac{m_i}{n} \tag{10}$$

where k is the cardinality value, and n is the number of objects, it equals to the number of features or the number of instances, depending on different IB application scenario.

3.2 Encoding Data Assuming IB Model

We just described how we can encode the IB solution, now we will see how to encode the data set D assuming the IB solution is right. Considering the fact that in IB applications, the data set D is actually a collection of x instances. From information theory [11], we know that *Conditional Entropy* $H(X|T)$ is the ambiguity of X after knowing T . In other words, it's the minimum additional number of bits needed on average to represent a source letter in X once T is known.

In the case of encoding the original data set D when the IB solution T_k is known, the encoding length can be estimated as n multiplies the $H(X|T_k)$, i.e.

$$\begin{aligned} MsgLen(D|k, T_k) &= n \times H(X|T_k) \\ &= n \times (H(X) - I(X; T_k)) \end{aligned} \tag{11}$$

where $I(X; T_k)$ is the mutual information between X and T_k , and $H(X)$ is the entropy of X .

Therefore, the total message length for the IB model and the original data set D can be estimated as

$$MsgLen = \log^*(k) + \sum_{i=1}^{k-1} \log \bar{m}_i - n \sum_{i=1}^k \frac{m_i}{n} \log \frac{m_i}{n} + n \times (H(X) - I(X; T_k)) \quad (12)$$

where k is the cardinality value, n is the number of objects in D .

3.3 Automatic SIB Algorithm — ASIB

Based on the encoding length formula 12, we can determine the optimal cardinality value k using an iterative algorithm as described in Algorithm 1: Starting with a range $[min_k, max_k]$ of possible cardinality values (by default, $min_k = 1$ and $max_k = n$), the algorithm runs through each possible k value, and a sequential IB algorithm is called to get the IB solution T_k , then the encoding length of the IB model is calculated. Finally the k which results in the minimum encoding message length will be returned as the optimal cardinality value.

Algorithm 1. Automatic SIB algorithm

Require: a joint distribution $p(x, y)$, the range of possible cardinality values $[min_k, max_k]$ (optional).

Ensure: the optimal cardinality value $bestK$, and the feature grouping result T_{bestK}
 $T \leftarrow$ an empty array with length $max_k - min_k + 1$

{ T will store the IB solutions corresponding to different k }

$MsgLen \leftarrow$ an empty array with length $max_k - min_k + 1$

{ $MsgLen$ will store the encoding length corresponding to different k }

for each $k \in [min_k, max_k]$ **do**

$T_k \leftarrow sIB(k)$

$MsgLen(k) \leftarrow \log^*(k) + \sum_{i=1}^{k-1} \log \bar{m}_i - n \sum_{i=1}^k \frac{m_i}{n} \log \frac{m_i}{n} + n \times (H(X) - I(X; T_k))$

if $msgLen(k) < msgLen(bestK)$ **then**

$bestK = k$

end if

end for{Iterate each candidate feature set to choose the optimal one}

returns $bestK$ and T_{bestK}

4 Experiment Design and Results Analysis

In this section, we evaluate the performance of the proposed *ASIB* algorithm under the document clustering scenario, as in the papers [8,10,4,12]. The difference between our method and previous IB experiments is that we don't assume a prior knowledge on the correct number of categories in the corpus, while in papers [10,4,12] a predetermined number of categories was provided to their algorithms.

Our method is to test whether the *ASIB* algorithm can correctly determine the number of categories in the document corpus. Intuitively, if the algorithm

works perfectly, it should reproduce exactly the correct number of documentation categorization. In practise, sampling errors might result in deviations from the original model, but algorithm which can discover a number similar to the original, must be considered to be performing better than those which do not.

4.1 Data Sets

Following [8,10,4,12], nine data sets in the *20NewsGroup* corpus [17] are re-examined in this paper. These nine data sets were got from news groups, and they have already been preprocessed by:

- Removing file headers, and left only the subject line and the body.
- Lowering all upper-case characters.
- Uniting all digits into a single“digit” symbol.
- Ignoring the non-alpha-numeric characters.
- Removing the stop-words and these words that appeared only once.
- Keeping only top 2000 words according to their contribution.

Each of these nine data sets consists of 500 documents randomly chosen from some discussion groups, and contains a sparse document-word count matrix: each row of this matrix represents a document $x \in X$, and each column corresponds to a word $y \in Y$, and the matrix value $m(x, y)$ represents the occurrence times of the word y in the document x . Thus, we can extract “document clusters” that capture most of the information about the words that occur. The details of these 9 data sets are tabulated in Table 1.

Table 1. Information of Nine Data Sets of *20NewsGroup* corpus

Data Sets	Associated Themes	Category Size	
Binary-1	talk.politics.mideast, talk.politics.misc	2	500
Binary-2	talk.politics.mideast, talk.politics.misc	2	500
Binary-3	talk.politics.mideast, talk.politics.misc	2	500
Multi5-1	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast	5	500
Multi5-2	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast	5	500
Multi5-3	comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast	5	500
Multi10-1	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns	10	500
Multi10-2	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns	10	500
Multi10-3	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns	10	500

4.2 Experiment Results and Analysis

In this experiment, we set $min_k = 2$ and $max_k=20$, and then the *ASIB* algorithm is then called to automatically determine the correct number of document categories, and cluster the documents.

In Table 2 we present the results returned by *ASIB*. From the table, it is evident that the *ASIB* algorithm successfully determined the correct number of categories in 6 out of 9 data sets, and located very close values for the other 3 data sets.

Table 2. Results returned by *ASIB* algorithm

Data Set	k	ASIB-II result error	
Binary-1	2	2	0
Binary-2	2	2	0
Binary-3	2	4	2
Multi5-1	5	5	0
Multi5-2	5	5	0
Multi5-3	5	7	2
Multi10-1	10	10	0
Multi10-2	10	9	1
Multi10-3	10	10	0

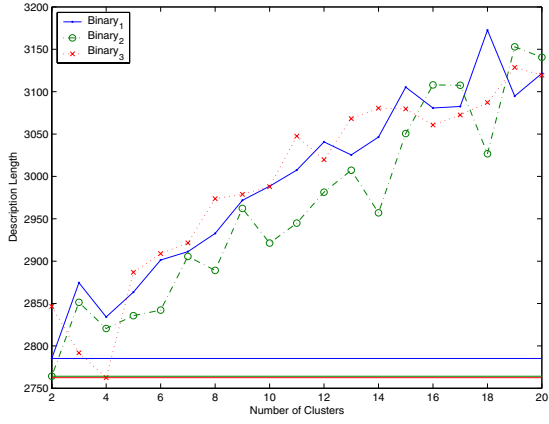
Figure 1 shows the general shape of the total encoding cost versus the number of possible categories. In Figure 1(a), it is interesting to note that the encoding length increases as the increasing of cluster number, and data sets *Binary-1* and *Binary-2* arrive at their minimum encoding length when the $k = 2$. Figure 1(b) and 1(c) present a similar results, in which we can find that the *ASIB* algorithm determined the correct number of clusters for *Multi5-1*, *Multi5-2*, *Multi10-1* and *Multi10-3*, and located very close results for data sets *Multi5-3* and *Multi10-2*.

These results are of special interest taking into account that no prior knowledge of the number of categories is provided to the *ASIB* algorithm. More over, there is strong empirical evidence suggesting that the proposed method could identify an appropriate parameter value for IB method automatically.

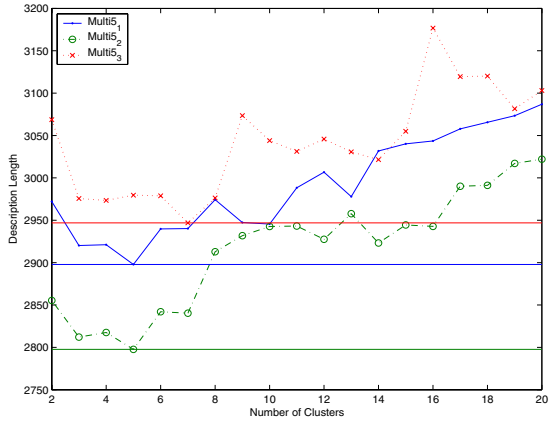
5 Conclusion

This paper removed this requirement by formulating the parameter determining as a model section problem, and solving it using the minimum message length principle. In this paper, we designed an encoding scheme for IB solutions corresponding to different cardinality values, and incorporated it into the *ASIB* algorithm to automatically determine the best cardinality value based on the data set itself.

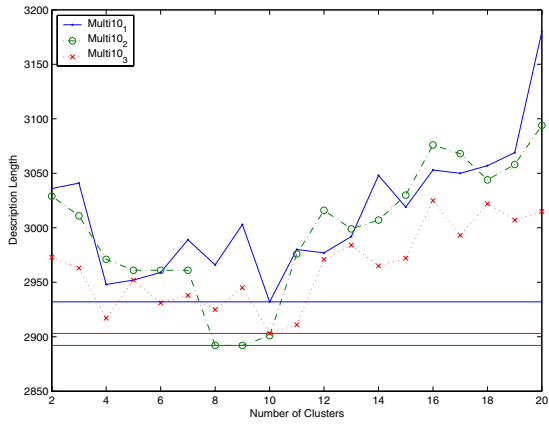
The experimental results on document clustering showed that the proposed *ASIB* algorithm is capable of recovering the true category number, and this



(a) Binary



(b) Multi5



(c) Multi10

Fig. 1. Cost versus number of Categories

indicates that the proposed parameter determining mechanism is correct. The main contributions of this paper can be summarized as follows:

1. This paper successfully solved the parameter determination problem for the Information Bottleneck method. No prior knowledge on parameter will be required, and the proposed algorithm can automatically determine the appropriate parameter value with a satisfactory accuracy;
2. We incorporated the proposed parameter determining mechanism into the IB algorithm, and proposed the *ASIB* algorithm, which can be considered as an improved version of *sIB*. We believe that the proposed method opens up many other directions of Information Bottleneck applications, such as data preprocessing, document classification, etc.

In our future work we are planning to extend the proposed method to the more general information bottleneck case in which the tradeoff parameter β is also unknown. We optimistically expect that these further work will promise to be of assistance to scientists wishing to analyze high dimensional data sets.

References

1. N. Tishby, F.P., Bialek, W.: The information bottleneck method. In: Proc. 37th Allerton Conference on Communication and Computation. (1999)
2. Shiri Gordon, Hayit Greenspan, J.G.: Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations. Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV) **2** (2003)
3. J. Goldberger, H.G., Gordon, S.: Unsupervised image clustering using the information bottleneck method. The annual symposium for Pattern Recognition of the DAGM02, Zurich (2002)
4. Slonim, N., Tishby, N.: Document clustering using word clusters via the information bottleneck method. In: Proc. of the 23rd Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval. (2000) 208–215
5. Verbeek, J.J.: An information theoretic approach to finding word groups for text classification. Masters thesis, The Institute for Logic, Language and Computation, University of Amsterdam, (2000)
6. Niu, Z.Y., Ji, D.H., Tan, C.L.: Document clustering based on cluster validation. In: CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, New York, NY, USA, ACM Press (2004) 501–506
7. E. Schneidman, N. Slonim, R.R.d.R.v.S.N.T., Bialek, W.: Analyzing neural codes using the information bottleneck method. Unpublished manuscript (2001)
8. Slonim, N., Tishby, N.: The power of word clusters for text classification. School of Computer Science and Engineering and The Interdisciplinary Center for Neural Computation The Hebrew University, Jerusalem, 91904, Israel. (2001)
9. N. Tishby, Slonim, N.: Data clustering by markovian relaxation and the information bottleneck method. Advances in Neural Information Processing Systems (NIPS) **13** (2000)

10. N. Slonim, N.F., Tishby, N.: Unsupervised document classification using sequential information maximization. In: Proc. of the 25th Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval. (2002)
11. THOMAS M.COVER, J.A.: Elements of Information Theory. City College of New York (1991)
12. Slonim, N.: The Information Bottleneck: Theory and Applications. PhD thesis, the Senate of the Hebrew University (2002)
13. Slonim, N., Tishby., N.: Agglomerative information bottleneck. Advances in Neural Information Processing Systems (NIPS) 12 (1999) 617–623
14. Wallace, C., Freeman, P.R.: Estimation and inference by compact coding. Journal of the Royal Statistical Society **49** (1987) 223–265
15. Wallace, C., Boulton, D.: An information measure for classification. Computer Journal **11** (1968) 185–194
16. Rissanen, J.: Universal Prior for Integers and Estimation by Minimum Description Length. Annals of Statistics **11** (1983) 416–431
17. Lang, K.: Learning to filter netnews. In Proc. of the 12th International Conf. on Machine Learning (1995) 331–339

A New Algorithm for Enumerating All Maximal Cliques in Complex Network*

Li Wan, Bin Wu, Nan Du, Qi Ye, and Ping Chen

Telecommunication Software Engineering Center,
School of Computer Science and Technology,
Beijing University of Posts and Telecommunications,
Beijing 100876, China
{wanli, dunan, yeqi, chenping}@tseg.org,
wubin@bupt.edu.cn

Abstract. In this paper, we consider the problem of enumerating all maximal cliques in a complex network $G = (V, E)$ with n vertices and m edges. We propose an algorithm for enumerating all maximal cliques based on researches of the complex network properties. A novel branch and bound strategy by considering the clustering coefficient of a vertex is proposed. Our algorithm runs with time $O(d^2 * N * S)$ delay and in $O(n + m)$ space. It requires $O(n * D^2)$ time as a preprocessing, where D, N, S, d denote the maximum degree of G , the number of maximal cliques, the size of the maximum clique, and the number of triangles of a vertex with degree D respectively. Finally, we apply our algorithm to the telecommunication customer-churn-prediction and the experimental results show that the application promotes the capabilities of the churn prediction system effectively.

1 Introduction

Graph-based data mining becomes a hot-spot research of KDD (Knowledge Discovery in Databases). Finding and enumerating sub-graphs of different structures in a single graph is one of the fundamental problems in combinatorics, which enjoys many important applications in fields of Graph Theory[1,2,3], Artificial Intelligence[4,5], Data Mining[6,7], Web Mining[8] as well as Bioinformatics[9]. Clique is one of the important sub-graph structures and a well-known NP-complete problem as well. Given an undirected graph G , the clique is a sub-graph with all pairs of its vertices adjacent. People often try to find the largest clique in G . This problem is termed as “maximum clique”[10,11,12]. Furthermore, we also want to solve “maximal clique” which is a clique that could not be contained as a sub-graph in any larger one, and our goal is to enumerate all maximal cliques in G . It is clear that the worst-case time complexity is $O(3^{(|V|/3)})$ in graph $G = \{V, E\}$ [13].

Since the famous small-world theory comes forth, people realize that the actual network in our daily life does not conform to a random graph-model in the traditional

* This work is supported by the National Science Foundation of China under grant number 60402011.

graph theory, but rather to a complex network with short average path-length, power law degree distribution as well as high clustering coefficient. With respect to real-world applications, more and more burgeoning researches focus on how to discover useful patterns and knowledge from complicated structural datasets involving rich link information. These researches are often referred as link mining analysis that emphasizes the interrelationship among entities and extends the traditional data mining technology by introducing new mining concepts, such as link prediction, frequent sub-graph mining and graph classification, which becomes an effective way for the knowledge-discovery from complicated structural datasets as a result [19].

Most existing algorithms of all-maximal-cliques enumeration is based on the random graph. With respect to real-world applications, this paper presents a novel triplet-based algorithm to enumerate all maximal cliques in a complex network depending on the key properties of the network. Our algorithm runs with time $O(d^2 * N * S)$ delay and in $O(n + m)$ space, requires $O(n * D^2)$ time as a preprocessing, where D , N , S denote the maximum degree of G , the number of maximal cliques, the size of the maximum clique respectively. d is the number of triplet containing a vertex with degree D and $d = C * T$ where C and T denote the clustering coefficient of each vertex with degree D and the maximum number of triplets that can be constructed from the vertices with degree D (i.e. $T = D * (D - 1) / 2$). Since vertices with large degree or d are few in complex network, the algorithm is more applicable to large scale datasets, especially when we bring the idea of enumerating all maximal cliques into the customer-churn analysis, the actual predictions and results are improved efficiently.

The remaining of the paper is structured as follows. Section 2 introduces the related work. Section 3 gives definitions and notations. Section 4 presents our algorithm in detail. We report the experiment results in Section 5. Finally, concluding remarks are stated in Section 6.

2 Related Works and Commercial Background

2.1 Complex Network

Generally speaking, a complex network [18] often takes on both small-world and scale-free features, which means short average path-length, power-law degree distribution and high clustering coefficient. Most networks in real world are complex network. This paper sets the branch-and-bound strategy based on the high clustering coefficient feature of the complex network. The ER model of random network proposed by Erdos and Renyi in 1959 is quite different from the complex network model. When N grows very large, the static characteristics of ER model could be summarized as low clustering coefficient, short average path length and Poisson degree distribution. By sufficient investigation and survey, maximal clique enumeration problem could be fully described by the clustering coefficient [18], so it is necessary to propose an algorithm applicable to the complex network to enumerate all maximal cliques.

2.2 Several Existing All-Maximal-Clique Enumeration Algorithms

The existing all-maximal-clique enumeration algorithms could be classified into two groups. One adopts the depth-first search algorithm [14, 15,16] and the other uses frequent pattern discovery algorithm based on the Brute Force [17] idea. In terms of the former, the main idea is to treat the discovery of all the maximal cliques as a search algorithm. All possible combinations of vertices in the network construct a search tree with each leaf node being correspondent to a maximal clique. As a result, the process of enumerating all the maximal cliques can be viewed as a depth-first traversal of this search tree. With respected to the latter, we can construct the maximal clique of size K based on the maximal clique of size $K-1$ and execute the same process recursively until all maximal cliques are found. Our algorithm belongs to the first group compared with the BK[14] algorithm.

Published in 1973, the base maximal clique enumeration algorithm [14], called herein Base BK, utilizes a recursive branching strategy. It uses three dynamically changing sets: COMPSUB, CANDIDATES and NOT. The algorithm is performed by a function called EXTEND, which operates as any recursive backtracking algorithm. This process can be viewed as a depth-first traversal of a search tree with each node being correspondent to a maximal clique.

The algorithm of [16] defines the parent-child relationship between two maximal cliques. Therefore, all maximal cliques form a search tree. This algorithm first finds a maximal clique K_0 of the maximum lexicographic order. Then constructed by the parent-child relationship and rooted as K_0 , the search tree is traversed to enumerate all the maximal cliques. In addition, given a lower boundary D , this algorithm divides the graph into two sub-graphs: V^* and V_2 . V^* consists of the vertices whose degree is larger than D and V_2 comprises of the left vertices. Firstly, we can find all maximal cliques in V^* and then find the maximal cliques in V_2 and eliminate those that appears in V^* and are contained in the maximal cliques in V_2 as sub-graphs. In short, The algorithm of [16] utilizes the parent-child relationship among all maximal cliques to construct a search tree and treat the vertices with large degree separately to improve the efficiency. However, there still exist two major drawbacks. The first one is that how to select the lower bound D does not have a common standard, which has a great impact on the performance of the algorithm. The second one is that when the child nodes computation involves too many expensive set operations. If the average size of all the maximal cliques is large, the performance will be deteriorated.

The algorithm of [17] belongs to the second group. It takes a very different approach than the recursive branching procedures previously described and depends on the fact that every clique of size k , where $k \geq 2$, is comprised of two cliques of size $k-1$ that share $k-2$ vertices. The algorithm takes as input an edge list with the edges (2-cliques) listed in non-repeating, canonical order and builds from it all possible 3-cliques. Any 2-clique that cannot become a component of a 3-clique is declared maximal and output. When all 3-cliques are generated, the 2-clique list is deleted. The algorithm then attempts to build 4-cliques from the previously constructed set of 3-cliques using the same procedure. This procedure of enumerating maximal cliques is repeated in the increasing order of clique size until it is no longer possible to build a larger clique. Unfortunately, the algorithm also has some less than appealing features. Firstly, it is evident that building cliques in this manner requires maintaining both the

(k-1)- and k-clique lists. This consumes an enormous amount of space. Secondly, the algorithm has a hidden cost. Every time a k-clique is formed, all (k-1)-cliques contained within the new clique must be marked as used; else they might be mistaken for maximal cliques.

2.3 Commercial Background

All-maximal-cliques enumeration enjoys pervasive applications and plays an important role in the research of Bioinformatics, Web-community discovery and Data Mining. This section mainly introduces the application of all-maximal-cliques enumeration in the customer-churn analysis.

Nowadays, as market is getting more and more competitive in telecoms industry, companies realize that customers are their major asset and that they should focus on how to keep and expand their customers. However, traditional customer-churn analysis only treats the attributes of a single user as the analysis entity while neglects the pervasive structural attributes. Recently, link mining becomes a hot-spot research field and draws lots of attentions and applications. By sufficient investigation, survey and analysis, we discover that the call-graph formed among all the telecom users conforms to a complex network model and the maximal clique describes the users who have a close relationship with each other. The total number and the size of the maximal cliques where any vertex in this call-graph reside reflect how this vertex (single user) relies on the network. By taking the number and size of the maximal cliques as the structural attributes of a single user plus link-based classification technology, the prediction of the customer-churn analysis could be improved efficiently.

3 Definitions and Notations

This section introduces some notions and notations of graphs used in the subsequent sections.

Let $G = (V, E)$ be a graph with a vertex set $V = \{v_1, \dots, v_n\}$ and an edge set $E = \{e_1, \dots, e_n\}$. $R(u)$ denotes the set of adjacent vertices of vertex u .

Definition 1. $S \subseteq V$, a vertex u in S , with two vertices v and h having larger indices than vertex u in S and (u, v) , (u, h) , (v, h) included in E , i.e. u, v, h construct a triangle, then S is a clique of graph G . If S is a clique of simple graph G , no vertex t in graph G satisfies that S unions $\{t\}$ constructs a clique, then S is a maximal clique of graph G .

Definition 2. If vertex v has d_v neighbors, then v and its neighbors can construct $m = d_v * (d_v - 1) / 2$ triangles at most, but the actual number of triangles constructed by v and the neighbors is H , so the clustering coefficient of v is $C_v = 2 * H / d_v * (d_v - 1)$.

Definition 3. If $P(u)$ is a set included in $R(u)$, then the Local Clustering Coefficient of u is $C(P(u))$, which has the same definition as Definition 3 but let the $R(u) = P(u)$.

Definition 4. n -Clique is a clique that contains n vertices, e.g. an edge of graph G is a 2-clique, a triangle in G is a 3-clique and so on.

D, N, S denote the maximum degree of G , the number of maximal cliques, and the size of the maximum clique respectively. d is the number of triangles containing a vertex with degree D , $d = C * T$ where C and T denote the clustering coefficient of the vertex with degree D and the maximum number of the constructed triangles that contain the vertex with degree D (i.e. $T = D * (D - 1) / 2$) respectively.

4 Algorithm for Enumerating All Maximal Cliques in Complex Network

Presently proposed algorithms to enumerate all maximal cliques mostly focus on generating all maximal cliques in random graphs. However in practical applications, we often need to enumerate all maximal cliques in complex networks. In this paper, we propose a CLIM algorithm that is efficient to enumerate all the maximal cliques in complex networks, by utilizing the large clustering coefficient property of the network.

4.1 Clustering Coefficient

As mentioned in section 2, the properties of complex network are dramatically different from those of random graph. After our researches we found that clustering coefficient is correlation to maximal clique more closely than other properties of complex network, so we put our work on the researching of clustering coefficient.

According to the second definition of clustering coefficient we can compute the clustering coefficient of vertex 1 in graph 1, the result is the same as the one computed by the definition 1 of the clustering coefficient. The difference between the definition 3 and 4 is that the formal one based on the basic 2-clique structure but the latter one based on the basic 3-clique structure. By utilizing the definition 2, we can generate all maximal clique more efficiently.

4.2 Maximal Clique

Similar to the definitions of clustering coefficient, the definition of maximal clique also has two forms which are mentioned in section 3. The first one uses basic 2-clique structure to describe the concept of maximal clique, the second one uses basic 3-clique structure to describe it, the form of definition actually compact the graph, it can promote the efficiency of the algorithms run on the graph, our examinations have proved it. A maximal clique by 4-cliques or n -cliques where n is larger than 4 can also be constructed, but the computation of 4-clique or n -clique also has some complexity. So we find 3-clique as the basic structure to construct maximal clique has a trade off. The definition of maximal clique constructed by 3-cliques has been described in section 3.

The existence algorithms of enumerating all maximal cliques are all mostly based on the definition 1 of maximal clique, such as algorithm [14][15][16] are all to generate the maximal cliques on the leaf nodes of the search tree by traverse it.

Although the algorithms mentioned above have the different strategy to cut off branches, but they all based on the 2-clique relationship of one candidate vertex with the nodes already on the search tree. But the algorithm we proposed has the cut off branches strategy based on the 3-clique relationship of every two vertices with the nodes already on the search tree. In next section we will describe our algorithm more detail.

4.3 Algorithm in Complex Network – CLIM Algorithm

Theorem1 and 2 are based on the relationship between clique and the clustering coefficient.

Theorem 1. set M contains vertices of a maximal clique in graph $G(V,E)$, if and only if there is a vertex u in M that satisfies the following two conditions

- a) $R(u)$ includes $M - \{u\}$
- b) u and $M - \{u\}$ has the local clustering coefficient $C(P(u)) = 1$

A vertex with clustering coefficient 1 can generate a clique with its adjacent vertices. Vice versa, a vertex having larger clustering coefficient can construct more and larger maximal cliques with its adjacent vertices than a vertex with the same degree. Our algorithm focuses on finding maximal cliques constructed by large clustering coefficient vertices efficiently.

Theorem 2. vertex v , $R(v)$ includes M , set M contains vertices of a maximal clique in graph $G(V,E)$, if and only if:

- a) vertex u is the smallest index vertex in M
- b) vertex t is any vertex includes in $M - \{u\}$
- c) $\langle v, u, t \rangle$ is a 3-clique structure in graph G
- d) $M - \{u\}$ satisfies conditions a) – c)

According to the discussion above, our algorithm has $O(d^2 * N * S)$ time delay in $O(m + n)$ space, where $O(n * D^2)$ time is required as a preprocessing to generate all 3-cliques in the graph.

Formally, our algorithm can be described as follows

Algorithm: FindAllMaxCliques

Input: Graph G

Output: All the maximal cliques in Graph G

step1: Read in Graph

step2: Compute all the triangles of vertices in

G .

step3: Call AllChildren (CandClique, set_Tri23).

step4: Output all cliques found.

Procedure: AllChildren (CandClique, set_Tri23)

step1:for each (Tri2, set_Tri3) in set_Tri23{

step2: for each vertex in set_Tri3 {

if vertex i in set_Tri3 is adjacent to
vertex j in set_Tri3{

```

        if(vertex i is not find in new_set_Tr23){
            new_Tri2 = vertex i
            input vertex j into new_set_Tri3
            input(new_Tri2,new_set_Tri3)into
            new_set_Tri23}
        else{
            input vertex j into new_set_Tri3
            input(new_Tri2,new_set_Tri3)into
            new_set_Tri23
        }}} //end of for in setp2
    step3:if new_set_Tri3 is not empty {
        new_CandClique = CandClique + Tri2
        Call AllChildren ( new_CandClique,
        new_set_Tri23)
    }else if new_set_Tri3 is empty {
        for each vertex i in set_Tri3{
            clique = CandClique + Tri2 + i
            check whether clique is a maximal
    clique
                if clique is a maximal clique{
                    put clique in the result set}}}}//end
                of for in step1
    step4: return.

```

Our algorithm can be viewed as a depth first traverse algorithm, and dynamically changes set *CandClique* and *set_Tri23* to generate all the maximal cliques. Set *CandClique* containing the vertices constructs a candidate clique; every element in set *set_Tri23* has a structure of $(Tri2, set_Tri3)$, where *Tri2* is a vertex and *set_Tri3* is a set of vertices. The correspondent vertices coming from *CandClique*, *Tri2* and *set_Tri3* can induct a sub-graph of 3-clique in graph G. Initially the *CandClique* is empty and *set_Tri23* is the adjacent list of graph G. *new_CandClique*, *new_set_Tri23* and *new_Tri2* are similar to *CandClique*, *set_Tri23* and *Tri2* respectively.

Now we explain the procedure of *AllChildren* in more detail. Step1 of *AllChildren* ensures that every element of *set_Tri23* can be traversed after the recursive call of *AllChildren*. Purpose of step2 is to compute the 3-clique constructed by the vertices in $(Tri2, set_Tri3)$, which is traversed in step1. If there exists a 3-clique that is constructed by *Tri2* with another two vertices in *set_Tri3*, we put it into *new_set_Tri23*, and let $new_CandClique = CandClique + Tri2$. If *new_set_Tri23* is empty, it indicates that we have traversed to a leaf node of the search tree. *CandClique*, *Tri2* and any vertex in *set_Tri3* construct a clique. If the clique is checked as a maximal one, then we put it in the result set. If the *new_set_Tri23* is not empty, it indicates that there are 3-cliques in $(Tri2, set_Tri3)$ and *new_CandClique* is surely not a maximal clique, so recursively call *AllChildren* with *new_CandClique* and *new_set_Tri23* as the parameters.

The step of checking whether a clique is a maximal one in step3 has the time delay of $O(d*S)$, so step3 has the $O(N*d*S)$ overall time delay. If we compute all the triangles in $O(n*D^2)$ time delay as preprocessing, step2 of *AllChildren* has $O(d)$ time delay. The algorithm has $O(d^2*N*S)$ time delay totally.

There are some algorithms that considered the problem of enumerating all maximal cliques in real world data (but not in complex network) as well, such as the algorithm

of [16]. Algorithm [16] processes the vertices with degree larger than a lower bound D recursively. However, it introduces a new problem concerning how to choose the exact value of D . We can see that different graphs have different values of D . Another less appealing feature of Algorithm [16] is that it generates the maximal cliques by the parent-child relationships between them, but when the relationship is becoming complicated between maximal cliques, it will cost too much time.

5 Computational Experiments and Application

5.1 Experiments of Maximal Clique Enumerating Algorithms

At the end of section 4, we compare our algorithm with algorithm [16] theoretically. To evaluate the performance of our algorithm, we implement our algorithm and the improved BK algorithm of [14]. Our codes are written in C++, and the program run in a PC of Pentium4 3.0GHz with 1.0GB memory, whose OS is WindowServer2003. We examine these algorithms by using graphs that are generated randomly and taken from telecommunication carrier’s call pair data of a month in a city.

Our random graphs are generated as follows. For given r and n , we construct a graph with n vertices such that v_i and v_j are adjacent with probability $1/2$ if $i + n - j(mod n) \leq r$ or $j + n - i(mod n) \leq r$. We examine the case of $r = 10$ and $n = 2000, 4000, 8000, 16000, 32000$. The complex networks are the calling records of a telecommunication carrier. Table1 represents the results of generating all maximal cliques in random graphs. Table2 represents the results of generating all maximal cliques in complex networks. Table3 represents the results of generating all maximal cliques in big complex networks, where only CLIM algorithm can generate all maximal cliques in acceptable time and the improved BK algorithm can not obtain the result in 5 hours so we do not list the time result of it. All the computational time in the tables is expressed in seconds.

By comparing Tab. 1 and Tab. 2, we can see that CLIM algorithm performs more efficiently than the improved BK algorithm on complex network. Tab. 1 shows that if the number of vertices of random graph is large, the CLIM performs better than improved BK, because when the random graph has more vertices, there will be more vertices with larger clustering coefficient as well. It proves that CLIM performs efficiently on the graph having large clustering coefficient. Tab. 3 shows that CLIM algorithm has time delay $O(d^2 * S)$ per maximal clique and CLIM can solve the problem of enumerating all maximal cliques in real world graphs efficiently.

Table 1. On Random graph

datasets	1	2	3	4	5
vertex	2000	4000	8000	16000	32000
edge	8938	17987	36223	71904	144350
MCN	4396	8894	18112	35897	68794
MCS	6	6	6	6	6
Improved BK (Sec)	11.672	150.656	2618.969	20912.109	>5 hours
CLIM (Sec)	130.766	327.000	1262.313	1796.375	4416.937

Table 2. On Complex network

Datasets	1	2	3	4	5	6
vertex	2752	3277	3831	4391	3318	5519
edge	2500	3000	3500	4000	4500	5000
Maximal	45	72	73	87	32	97
Maximum	5	5	5	5	5	5
Improved BK(Sec)	18.07	29.53	45.76	66.21	30.78	132.32
CLIM(Sec)	0.219	0.328	0.360	0.422	0.281	0.563

Table 3. On Call Pair of a month

Datasets	1	2	3	4	5
vertex	512024	503275	540342	539299	543856
Edge	1021861	900329	1030489	1014800	1034291
Maximal	153362	118353	143259	139040	145569
Maximum	14	13	14	15	17
Time(Sec)	1283	530	892	1053	2542

5.2 Application of Maximal Cliques in Customer Churn Prediction

Customer churn analysis is an interesting problem in telecommunication industry. Effective prediction means more retained customers and profits. The process to predict the customer churn is to make customer segmentation.

Maximal cliques reflect the closeness of one customer with other customers in the call graph. By analyzing these maximal cliques, we can get the structure information of a cluster of customers. Clique is a better static structure to indicate the possibility of customer churn than other static structures such as smallest path, expand and so on[18],the link based classification technology is used in this link based churn prediction [19].

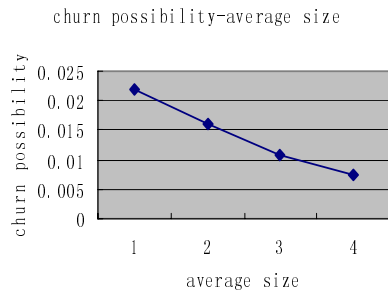
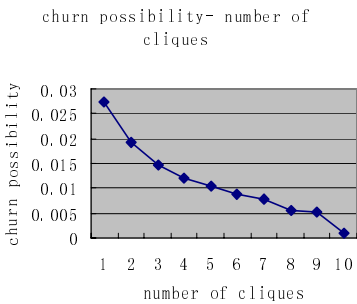


Fig. 1. 2004-1churn possibility-clique number **Fig. 2.** 2004-1churn possibility-clique avg-size

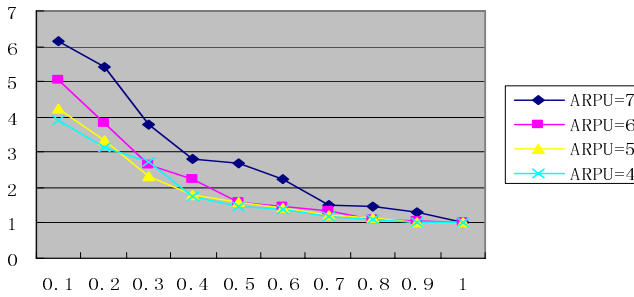


Fig. 3. Lift value in different customer group with different ARPU

The Figs above come from the statistical results of the customer records in 2004.1 and 2004.2 of a telecommunication carrier in a city. The total churn possibility of these customers is about 3%. Fig1 and 2 above shows the statistical relationship of churn possibility with the number and the avg-size of maximal cliques a customer in respectively. The Y axes is the possibility of customer churn, X axes is the number of maximal cliques and maximal clique's average size respectively. From the Figs above we can see that the possibility is negative along with the number of maximal cliques and the average size of maximal cliques.

Compared with the traditional way of churn prediction, by introducing the maximal clique, degree, and some other static structures as the properties of customers, we improve the churn prediction. The prediction result is especially satisfactory towards the VIP customers who are the most important customers to the telecommunication carrier. Fig 3 shows the lifted value of the churn prediction method on customers of different value. The ARPU indicates the consumption of a customer per month.

6 Conclusion and Future Work

Unlike the existing algorithms that are fit for the maximal clique problem in random graphs, we propose an efficient algorithm to enumerate all maximal cliques in complex network. Since these two graphs have very different properties, the algorithm that performs efficiently on random graph may not perform well on complex network. CLIM algorithm utilizes the large clustering coefficient property to promote the performance of enumerating all the maximal cliques in a complex network. We also apply our algorithm to the customer churn prediction and improve the prediction of the VIP customer especially.

Future work is mainly focus on applying the maximal clique to solve more problems, for example, we can combine it with the link-based cluster technology in link mining. The research of the evolution of the collaboration networks is also interesting domains.

Acknowledgement

We thank the members of Telecommunication Software Engineering Center at Beijing University of Posts and Telecommunications for their suggestions and support. We also acknowledge the suggestions made by two anonymous referees that helped to improve the paper presentation.

References

1. D.S. Johnson, M. Yanakakis and C.H. Papadimitriou On generating all maximal independent sets, *Info. Proc. Lett.*, 27 (1988) 119 - 123
2. R. C. Read and R. E. Tarjan, Bounds on backtrack algorithms for listing cycles, paths, and spanning trees, *Networks*, 5(1975) 237-252
3. S. Tsukiyama, M. Ide, H. Ariyoshi and I. Shirakawa, A new algorithm for generating all the maximal independent sets, *SIAM J. Comput.*, 6(1977) 505-517
4. T. Eiter and K. Makino, On computing all abductive explanations, *Proc. AAAI '02*, AAAI Press, pp.62 – 67, 2002
5. B. Selman and H. J. Levesque, Support set selection for abductive and default reasoning, *Artif. Int.*, 82(1996) 259 - 272
6. R. Agrawal and R. Srikant, Fast algorithms for mining association rules in large databases, *Proc. VLDB '94*, pp.487 – 499, 1994.
7. R. Agrowal, H. Mannila, R. Srikant, H. Toivonen and A.I. Verkamo, Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, MIT Press, pp. 307 – 328, 1996.
8. S. R. Kumar, P. Raghavan, S.Rajagopalan, and A. Tomkins, Trawling the web for emerging cyber-communities, *Proc. the Eighth International World Wide Web Conference*, Toronto, Canada, 1999.
9. Faisal N. Abu-Khzam, Nicole E. Baldwin, Michael A. Langston and Nagiza F. Samatova On the relative efficiency of maximal clique enumeration algorithms, with application to High-Throughput computational biology. *Proceedings, International Conference on Research Trends in Science and Technology*, Beirut, Lebanon, 2005
10. I. Bomze, M. Budinich, P. Pardalos, and M. Pelillo, "The maximum clique problem," in *Handbook of Combinatorial Optimization*, vol. 4, D.-Z. Du and P. M. Pardalos, Eds.: Kluwer Academic Publishers, 1999).
11. Patric R. J. A fast algorithm for the maximum clique problem, *Discrete Applied Mathematics* 120 (2002) 197–207
12. Etsuji Tomita and Tomokazu Seki An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique *DMTCS* 2003, pp. 278–289, 2003.
13. Etsuji Tomita, Akira Tanaka, Haruhisa Takahashi, The worst-case time complexity for generating all maximal cliques, *COCOON 2004*, LNCS 3106, pp.161 – 170, 2004.
14. C. Bron and J. Kerbosch, Algorithm 457: Finding all cliques of an undirected graph, *Proceedings of the ACM*, vol. 16(9), 1973, 575-577.
15. S.Tsukiyama,H.Ariyoshi, I.Shirakawa, A New Algorithm for Generating all the Maximal Independent sets, *SIAM J. COMPUT.* Vol. 6. No.3, September 1977.
16. Kazuhisa Makino, Takeaki Uno, New Algorithms for Enumerating All Maximal Cliques, *SWAT 2004*, pp.260 – 272 ,2004.

17. F. Kose, W. Weckwerth, T. Linke, and O. Fiehn, Visualizing plant metabolomic correlation networks using clique–metabolite matrices, *Bioinformatics*, vol. 17, 2001, 1198-1208.
18. Wang Yanhui, Wu Bin, Wang Bai, Research on Static Measures of Telecom Society Network (in Chinese), *COMPLEX SYSTEMS AND COMPLEXITY SCIENCE* 2005 Vol.2 No.2
19. Lise Getoor, Christopher P.Diehl ,Link Mining:A Survey, *Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, December 2005. Volume 7, Issue 2.

Modeling and Mining the Rule Evolution*

Ding Pan

Management School, Jinan University, Guangzhou 510632, China
Department of Computer Science, Xi'an Jiaotong University, Xi'an 710049, China
dingpan.cn@gmail.com

Abstract. Temporal data mining attempts to provide accurate information about an evolving business domain. A framework is proposed to discover continuously temporal knowledge based on a session model. The main concepts and properties in temporal rule induction are defined and proved in a formal way, using first-order linear temporal logic. The measures of first-order rule are used to discover evolutionary regularity about the rule. The mining process consists of four stages: planning, session mining, merge mining, and post-processing. Various session mining for temporal data generates a measure sequence of first-order rule. The parameter estimation method applicable to the measure sequence with a small-sample is presented, based on the principle of information diffusion. Experiment shows the validity and simplicity of the method.

1 Introduction

With the explosive growth of available data, there is an urgent need to develop continuous data mining which reduces manual interaction evidently. The temporal data mining (TDM) has the ability to mine the behavioral aspects of objects as opposed to simply mining rules that describe their states at a point in time.

Agrawal et al. [1] showed an active data mining, where the mining algorithm was applied to each of the partitioned data set and rules were induced. The problem of finding an effective mining algorithm has received much attention and has made great progress so far [2,3]. In contrast, the problem of presenting a mining formal work has received far less attention. Cotofrei et al. [4] investigated the form of temporal rules of time series, and presented a formalism of main notions, based on the first-order temporal logic. However, the work is limited by the fundamental condition, while a particular strategy has been adopted to construct the train set in order to infer temporal rules using classification tree. Higher order mining attempts to discover the interesting, more understandable higher semantic rules, namely, higher order rule [4,5].

Huang [6] proposes principle of information diffusion, to estimate the statistical parameters of small-sample. As a soft computing technique, the theory has been developed in surveying and mapping, risk analysis, biomedical engineering, etc.

As the basis of the continuous data mining, we attempt to update and expand the formalism theory of [4], present definition of the concepts used in temporal rule

* Supported by the National Natural Science Foundation of China under Grant No. 70372024.

induction, and discuss the valuation of the formula and measure definition, based on first-order linear temporal logic. From the application viewpoint, we propose a novel continuous data mining process model based on a session model, and parameter estimation for the measures based on principle of information diffusion.

This paper is organized as follows: Section 2 describes the background. Section 3 presents the formalism of temporal rule induction. Section 4 briefly discusses continuous data mining process. Section 5 proposes the estimation method of rule measures. Finally, Section 6 concludes the paper.

2 Background

We consider a linearly ordered temporal domain T . For simplicity, we assume that the elements of T are strictly increasing, and $t_{i+1}-t_i=\Delta$ is a positive constant. So, given the T , a non-empty attribute set $\{A_1, \dots, A_k\}$ and involved domain D_{A_i} on A_i , then a temporal sequence is a ordered item list $X=\{X_1, X_2, \dots, X_m\}$, where X_i is a $k+1$ -tuple (t, a_1, \dots, a_k) , $t \in T$, $a_i \in D_{A_i}$. There are three categories of sequence: 1) X is a time series, when $D_{A_i} \subseteq R$. 2) X is an event sequence, when $D_{A_i} \subseteq \Sigma$, Σ denotes to an alphabet. 3) X is a transaction sequence, when $D_{A_i} \subseteq R^+$ or $D_{A_i} \subseteq \{0, 1\}$. A sequence space W_X consists of all instances of a sequence category. Attribute set $\{A_1, \dots, A_k\}$ denotes to each attribute of finite entity set Q (such as customer, stock, etc.). Mined data set of TDM w_X is a subset of $Q \times W_X$.

Data representation and pre-processing are important while dealing with temporal sequence, since it is extremely difficult to manipulate directly the high dimensionality, high feature correlation in an efficient way. Some possible solutions are sequence discretization and clustering transformation. After the application of data representation and pre-processing, we have transformed the sequence X in w_X into linear ordered sequence of events $S=(S_1, \dots, S_k)$ consisted of some basic, interesting shapes or strings.

Given a finite symbol set D_e of the basic shapes or strings, the feature function set $\{f_1, \dots, f_p\}$ ($p \geq 0$) and each corresponding domain D_{f_i} , then the event set of w_X is $E_s = \{(e, b_1, \dots, b_p) \mid e \in D_e, b_i = f_i, b_i \in D_{f_i}\}$.

For classification rule, there is a finite class symbol set D_g . The mapping $w_X \rightarrow D_g$ must be specified prior to the induction, where w_X is called as *train set* for supervised learning, while the induction process generates $w_X \rightarrow D_g$ for unsupervised learning.

Example. Given a database containing daily price variations of the stocks, specified basic symbol set $D_e = \{\text{top, bottom, rise, drop, flat}\}$. Each event has form (e, b_1, b_2) , where e is a element of the D_e , and b_1, b_2 represent the means respectively, the standard error. We assume that the price sequence of IBM stock is transformed into the event sequence $((\text{flat}, 3, 1.5), (\text{rise}, 10, 2.4), \dots, (\text{top}, 8, 1.4))$. For classification task, there is a class symbol set $D_g = \{\text{grow, stabile, wave, risk}\}$.

3 Formalism of Temporal Rule Induction

A first-order logic language contains usually constant, function, predicate, and general symbols like connectives, etc. For the requirement of formalism we consider a restricted first-order temporal logic language L , which contains only constant symbols, function symbols, predicate symbols, relational symbol set $\{=, <, \leq, >, \geq\}$, a logical connective $\{\wedge\}$ and a temporal connective $X_k, k \in \mathbb{Z}$, where $k > 0$ denotes next k time instants, $k < 0$ denotes last k time instants, $k = 0$ denotes now [4].

3.1 Syntax and Semantics

The syntax of L defines the set of terms, atomic formulae (or atom) and formulae, according to the general form. A quasi-Horn clause is a formula of form: $A_1 \wedge A_2 \wedge \dots \wedge A_k \Rightarrow A_{k+1}$, if and only if it is syntactically equivalent with the formula $A_1 \wedge A_2 \wedge \dots \wedge A_k \wedge A_{k+1}$, where each A_i is a positive atom.

Definition 1. An event is an atom formed by a $p+1$ -ary predicate $E(e, f_1, \dots, f_p) (p \geq 0)$, where e is a constant symbol representing the name of the event, and f_1, \dots, f_p are the function symbols.

Definition 2. A constraint formula for the event $E(e, f_1, \dots, f_p)$ is a conjunctive formula, $C_1 \wedge C_2 \wedge \dots \wedge C_m$, where C_i is a relational atom. It contains a $t_e = e$ and some $t_j \rho b$, where $\rho \in \{=, <, \leq, >, \geq\}$ and $0 \leq j \leq p$. The t_e, t_j are variable symbols representing the name of event and corresponding f_i respectively. The b is a constant symbol. A temporal constraint formula H_k denotes $X_k(C_1 \wedge C_2 \wedge \dots \wedge C_m), k \in \mathbb{Z}$.

Definition 3. A sequence pattern (also called pattern) is a conjunctive formula of several ordered temporal constraint formula $H_k, H_{i_1} \wedge H_{i_2} \wedge \dots \wedge H_{i_m}, i_1 < i_2 < \dots < i_m$.

Definition 4. A temporal rule is a formula of the form $H_{i_1} \wedge H_{i_2} \wedge \dots \wedge H_{i_{m-1}} \Rightarrow H_{i_m}$, where H_{i_m} contains a relational atom $t_e = e$, and $i_1 < i_2 < \dots < i_m$.

The above $H_{i_1} \wedge \dots \wedge H_{i_{m-1}}$ is rule body. For the global properties, a class is an atom formed by the predicate $G(g)$, where g is an element of the class symbol set D_g .

Definition 5. A classification rule is a formula of the form $H_{i_1} \wedge H_{i_2} \wedge \dots \wedge H_{i_m} \Rightarrow G(g)$, where $i_1 < i_2 < \dots < i_m$.

The temporal rule, classification rule and pattern are called first-order rule R_F , by a joint name. $|i_m - i_1|$ is the time interval of R_F . When only order of the event is considered, the temporal connective X_i may be omitted in R_F .

The semantics of formulae of L is provided by an interpretation. For structure $U = (D, \{a^i\}, \{f^j\}, \{R^i\})$, where D is domain, a^i, f^j and R^i represent constant, total function and predicate respectively on D , the constant, function and predicate symbols of L are mapped to U , respectively. Moreover, the individuals in D are assigned to interpreted freedom variables. The interpretation and assignation are called valuation jointly. For a formula p , the meaning of truth under valuation V is denoted for $\forall \models p$.

In the TDM, given $D=w_X \cup D_e \cup D_f \cup D_g$, the feature function set $\{f_1, \dots, f_p\}$ is defined on D . To define a first-order linear temporal logic based on L , we need a structure having a temporal dimension and capable of valuating the relationship between a specific moment and valuation V .

Definition 6. Given L and a domain D , a linear state structure is a quintuple $M=(U, E_s, Tr, \Omega, V)$, where $U=(D, \{a^i\}, \{f^j\}, R^i)$, E_s is the event set, $Tr:w_X \rightarrow N \times E_s$ is a function that maps sequence X into a state sequence $(S_{(1)}, \dots, S_{(i)}, \dots)$, Ω is the set of state sequence, V is a function that associates with each $S_{(i)}$ an valuation V_s of symbols of L .

Given a linear state structure M and $\Omega=\{S^1, \dots, S^k, \dots, S^n\}$, we denote the $V|=p$ of a sequence S^k at a state $S_{(i)}$ (or X_i) by $(M, S^k, i)|=p$ or simply $i_j|=p$. We can also define: $i|=p \wedge q$ if and only if $i|=p$ and $i|=q$; $i|=X_k p$ if and only if $i+k|=p$. So, $i|=E(e, f_1, \dots, f_p)$ denotes that an event with the name e and the features $V(f_1), \dots, V(f_p)$ occurs at state $S_{(i)}$. Analogously, a temporal constraint formula H_k is true at state $S_{(i)}$ if and only if all $i|=C_j$; a pattern is true at state $S_{(i)}$ if and only if all $i|=H_{i_k}$; a temporal rule is true at state $S_{(i)}$ if and only if $i|=H_{i_1} \wedge H_{i_2} \wedge \dots \wedge H_{i_{m-1}} \wedge H_{i_m}$. Moreover, A classification rule is true at state $S_{(i)}$ if and only if $i|=H_{i_1} \wedge \dots \wedge H_{i_m}$ and $i|=G(g)$.

We can establish some measures about ranged degree of truth of a formula $V|=p$, considering usually data is incomplete or missing in the TDM. Now assuming that for each formula p , there is an algorithm to calculate the value of $V(p)$ for every state on mined dataset, in a finite number of steps.

Definition 7. Given L and a linear state structure M , for every formula p , on the set of state sequence Ω , a real set function $P(p)=|A|/n$, where $n=|\Omega|$ and $A=\{k \in \{1, \dots, n\} | (M, S^k, i)|=p\}$.

Theorem 1. Given L and a linear state structure M , for a formula p , the real set function $P(p)$ is a probability of $V|=p$ on the Ω .

Proof. By applying M , suppose $\Omega=\{S^1, \dots, S^k, \dots, S^n\}$ as sample space. Let $F=2^\Omega$, then $\Omega \in F$ and $Q \in F$, where Q is an arbitrary subset of Ω . Therefore, F is a σ -algebra.

For a formula p , let $Q=\{S^k | (M, S^k, i)|=p\}$, $A=\{k | S^k \in Q\}$, then $|A| \geq 0$, $P(p) \geq 0$. When $Q=\Omega$, $|A|=n$, then $P(p)=1$. Again let $|A_j|=K_j (\leq n)$, then corresponding $P(p_j)=K_j/n$ ($j=1, 2, \dots, m$). If the A_j s are not intersectant, then for the corresponding p_j :

$$P(\sum_{j=1}^m p_j) = (\sum_{j=1}^m K_j) / n = \sum_{j=1}^m \frac{K_j}{n} = \sum_{j=1}^m P(p_j)$$

Therefore, (Ω, F, P) is a probability space.

Definition 8. Given L and a linear state structure M , a measure for the $V|=p$ of a formula p is a function $supp(p)=P(p)$. The measure is usually called the support of p .

There is another useful measure for a temporal rule.

Definition 9. Given L and a linear state structure M , for the $V|=p$ of a temporal rule p , a measure of p is a function $conf(p)=P(p)/P(p_b)$, where p_b is the rule body. The $conf(p)=0$ if $P(p_b)=0$. The measure is usually called the confidence of p .

3.2 Session Model

In practical application, the user has no access to the entire sequence, or the sequences mined have only finite time intervals. Therefore, the measures should be calculated in a finite linear state structure, i.e. a session model.

Definition 10. Given L and a linear state structure M , a session model for M is a structure $\tilde{M} = (\tilde{\Omega}, sl)$, where sl is length of the session, $\tilde{\Omega} = \{\tilde{S}^k = \{S_{i_1}^k, S_{i_2}^k, \dots, S_{i_{sl}}^k\} \mid \tilde{S}^k \subseteq S^k, 1 \leq k \leq n\}$.

Definition 11. Given L and a session model \tilde{M} for M , an estimator of $supp(p)$ of a formula p is $ES(p, \tilde{M}) = |A^e|/n$, where $|\tilde{\Omega}| = n, A^e = \{k \in \{1, \dots, n\} \mid (\tilde{M}, S^k, i) \models p\}$.

Definition 12. Given L and a session model \tilde{M} for M , an estimation of the $conf(p)$ of temporal rule p is $EC(p, \tilde{M}) = ES(p, \tilde{M}) / ES(p_b, \tilde{M})$, where p_b is a rule body, if $ES(p_b, \tilde{M}) = 0$, then $EC(p, \tilde{M}) = 0$.

Definition 13. Given L and a session model \tilde{M} for M , a session mining SM_F for \tilde{M} is a sextuple $(Task, T_{st}, w_X, Stat, DK, R)$, where $Task$ is a task of TDM, T_{st} is a start time of mining, w_X is a mined data set, $Stat$ is a threshold of the measure, DK is domain knowledge, $R = \{r \in R_F \mid Stat(r) \wedge DK(r)\}$.

According to the definition, the estimator sequence of the measures, called measure sequence, is generated across various sessions, for the formula p . For example, the support sequence, $ES_1, ES_2, \dots, ES_r, \dots$, and the confidence sequence, EC_1, \dots, EC_r, \dots .

Definition 14. Given L and a linear state structure M , a sequence S is consistent for a formula p , if the limit $\lim_{m \rightarrow \infty} \frac{|B|}{m}$ exists, where $B = \{i \in \{1, \dots, m\} \mid (M, S, i) \models p\}$. The set of state sequence Ω is a p consistent set if every sequence in Ω is consistent for the p .

Theorem 2. Given L and a session model \tilde{M} for M , if a formula p exists the consistent set Ω , then for the p , when sl of the \tilde{M} is long enough, $\lim_{r \rightarrow \infty} ES_r = P(p) = |A|/n$, where $|\Omega| = n, A = \{k \in \{1, \dots, n\} \mid (M, S^k, i) \models p\}$.

Proof. Let $\Omega = \{S^1, \dots, S^n\}$ is a p consistent set, then for a $S^k \in \Omega$, the limit $\lim_{m \rightarrow \infty} \frac{|B|}{m} = \alpha_k$ exists, where $B = \{i \in \{1, \dots, m\} \mid (M, i) \models p\}$. Therefore, there is $\alpha(p) = \{\alpha_1, \dots, \alpha_n\}$ in the Ω . Let $\alpha = \min(\{\alpha_j \in \alpha(p) \mid \alpha_j > 0\})$, $sl = \max(1/\alpha, \text{the time interval of } p)$.

Let $\tilde{M} = (\tilde{\Omega}, sl)$, the estimator sequence of support is $ES_1, ES_2, \dots, ES_r, \dots$. Obviously, $A^e \subseteq A$. If $j \in A = \{k \in \{1, \dots, n\} \mid (M, S^k, i) \models p\}$ for arbitrary S^j , then $\alpha_j > 0$. When r is large enough, $|\{i \in \{r_1, \dots, r_{sl}\} \mid (\tilde{M}, \tilde{S}^j, i) \models p\}| > 0$ for the subsequence $\tilde{S}^j = \{S_{i_1}^j, S_{i_2}^j, \dots, S_{i_{sl}}^j\}$, so $j \in A^e$, viz. $A \subseteq A^e$. Therefore,

$$\lim_{r \rightarrow \infty} ES_r = \lim_{r \rightarrow \infty} \frac{|A_r^e|}{n} = \frac{1}{n} \lim_{r \rightarrow \infty} |A_r^e| = \frac{|A|}{n} = P(p).$$

Generally, there are three categories of basic trend for the measure sequence of: ascend, descend and fluctuation. High order rule is used to describe the dynamic characteristic of the first-order rule. Its syntax is the same as the definition of the first-order rule above, but its domain involves the R_F and the measure sequence.

4 Prospect of Applications

The classical data mining process is limited in data preparation and mining operation according to the specific goal, whose background is mainly absolute mining task. In order to improve efficient and effective operation, it is needed to develop an autonomous, continuous mining process in the TDM environment. A C-KDD process and architecture for continuous TDM is proposed, based on the above session model [7].

The C-KDD process consists of four stages: planning, session mining, merge mining, and post-processing. During the planning stage, through interactive exploration and experimentation, the discovery goals, business data, and subsequent processes are identified and the specification of discovery task schedule is generated. The session mining stage performs select-transfer-premining and achieves partial data mining. It places emphasis on local and static rules induction, periodically repeating on incremental data. The merge mining, called high order mining too, attempts to discover evolutionary regularity among rules, based on the results of previous session mining runs. The post-processing stage evaluates the discovered rules, and filters useless ones. We will not discuss here in details for space limitation.

5 Parameter Estimation of Rule Measures

Several session minings generate the measure sequence, $S_R = \{ES_1, ES_2, \dots, ES_n\}$. The parameter estimation of the S_R is an important task during the merge mining.

5.1 Principle of Information Diffusion and Parameter Estimation

Let W be a sample drawn from population U . W is incomplete, if the probability density function $f(x)$ of U cannot be understood accurately from W . The principle of information diffusion is described informally as: Let $W = \{w_1, w_2, \dots, w_n\}$ be a sample, V be the universe of discourse, and v_j be an observation of w_j . If W is incomplete, there is a reasonable diffusion function $\mu(x)$ which can lead to the information obtained from v_j , diffuse to v according to $\mu(\varphi(v, v_j))$, and the diffusion estimate is nearer to the real distribution than non-diffusion estimate [6].

Definition 15[6]. Let $\mu(x)$ is a Borel measurable function in $(-\infty, \infty)$, $d > 0$ is a constant, n is the sample number. Then,

$$\hat{f}_n(v) = \frac{1}{nd} \sum_{j=1}^n \mu\left(\frac{v - v_j}{d}\right) \tag{1}$$

is called the diffusion estimation about $f(x)$, where d is called diffusion coefficient.

The measure sequence is an incomplete sample. Let $W = \{S_1, S_2, \dots, S_n\}$, $U=[0,1]$, $C=\{c_1, c_2, \dots, c_m\} \subseteq U$, where $0 \leq c_1 < c_2 < \dots < c_m \leq 1$ and $c_{i+1} - c_i = \Delta$ is a constant. The c_i is called a control point. Let μ be a normal diffusion function, then according to (1), the control point c_i receives total information from sample W as follow:

$$Q(c_i) = \hat{f}_n(c_i) = \frac{1}{nd} \sum_{j=1}^n \mu\left[\frac{c_i - ES_j}{d}\right] = \frac{1}{\sqrt{2\pi nh}} \sum_{j=1}^n \exp\left(-\frac{(c_i - ES_j)^2}{2h^2}\right), \tag{2}$$

where $h = \sigma d$, generally, $h = \alpha(ES_{max} - ES_{min}) / (n-1)$. If $n \geq 10$, $\alpha = 1.4208$ [6].

If the frequency of the sample, $P_i = Q(c_i) / \sum_{i=1}^m Q(c_i)$, is regarded as probability estimation of c_i , its expectation estimation is $\hat{\mu} = \sum_{i=1}^m c_i P_i$, viz. the expectation of measure estimator. Therefore, $\hat{\mu}$ can be regarded as diffusion estimation of support or confidence. This method is a deduction essentially, to estimate parameter by data driven, not to be given prior model structure of the sample.

In (2), suppose that the samples come from the same type of observations and have the same expectation $E(v_i) = \mu$. For fluctuating W , the μ is constant, but for ascending/descending W , μ is not a constant. To estimate the parameter of measure sequence in ascending/descending trend, the variation of μ may be regarded as the local excursion along with measure estimator generated. The samples form several clusters in the center of local μ along time axis, and the samples in each cluster have constant expectation μ . The algorithm of diffusion estimation is shown below. The algorithm needs the effort of $O(knI)$, where k is cluster number, $n = |W|$, and I is iteration number.

Diffusion_Estimation(*samp_data*, *k*, *control_no*)

- step 1. Obtains k center points of clusters, using k-means algorithm on W ;
- step 2. For each clustering $O_j (1 \leq j \leq k)$ do (Estimation in k center points)
 - step 2.1. Obtains state t_j of center point of cluster O_j ;
 - step 2.2. Compute $Q(c_i)$ according to (4);
 - step 2.3. $P_i = Q(c_i) / \sum_{i=1}^m Q(c_i)$, $\hat{\mu}_j = \sum_{i=1}^m c_i P_i$
- step 3. Let linear equation of μ be $\mu = a + bt$, where t is a variable, a, b are constants. The μ is fitted by $(t_j, \hat{\mu}_j)$ s obtained in step 2.

5.2 Experiments and Discussion

We performed experimental evaluations on a real dataset from a tobacco distributor since Oct. 2004 to Nov. 2005, involving the sale of 18 tobacco products. The experiment was performed on a Pentium 2.4GHz PC with 1GB main memory, Windows 2000 Server, Matlab 6.5, and compared our method with least square methods.

The experiment revealed comparable results in Fig. 1. The obvious errors occurred in least square estimation due to sample 8, 11 and 14. The diffusion estimation is more robust than the least square estimation. The diffusion estimation is non-bias, the same as the least square estimation, when the sample does not include mistaken and

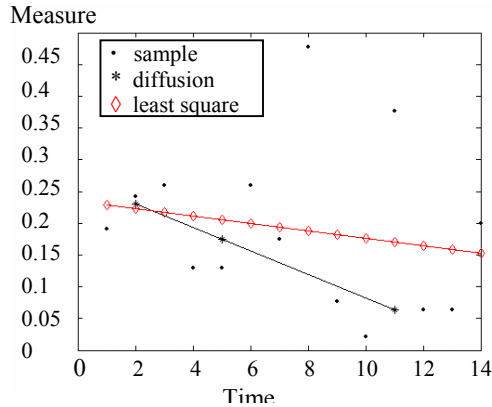


Fig. 1. Diffusion estimation vs. least square estimation

exception value; on the contrary, the diffusion estimation can be away from the effect of mistaken and exception value. This comparison confirms the inherent advantage of diffusion estimation over the least square estimation.

6 Conclusions

We have discussed a general framework for continuous temporal knowledge discovery, and presented the prospect of mining applications. It defines the valuation for the measure of a formula on a linear state structure, constructs the measure sequence based on a session model, and proves the consistency in the estimator of a formula satisfied consistent characteristic. A novel C-KDD process for continuous TDM is proposed. The parameter estimation of rule measures may obtain basic characteristics of dynamic evolution. Experiment shows the validity and simplicity of our method.

References

1. Agrawal, R., Psaila G.: Active data mining. In: Proc. of the KDD'95. AAAI Press, California (1995) 3-8
2. Roddick, J.F., Spiliopoulou M.: A survey of temporal knowledge discovery paradigms and methods. *IEEE Trans. on Knowledge and Data Engineering*, Vol. 14, No.4, (2002) 750-768
3. Keogh, E.J., Kasetty S.: On the Need for Time Series Data Mining Benchmarks. *Data Mining and Knowledge Discovery*, Vol. 7, No.4, (2003) 349-371
4. Cotofrei, P., Stoffel, K.: From Temporal Rules to Temporal Meta-rules. In: Kambayashi, Y. et al. (Eds.): Proc. of the DaWaK 2004, LNCS 3181. Springer, Zaragoza (2004) 169-178
5. Spiliopoulou, M., Roddick, J.F.: Higher Order Mining. In: Proc. 2nd Intl. Conf. on Data Mining Methods and Databases, Cambridge, UK (2000) 309-320
6. Huang, C.F., Shi, Y.: Towards Efficient Fuzzy Information Processing - Using the Principle of Information Diffusion, Physica-Verlag, Heidelberg, 2002.
7. Pan, D., Shen, J.: Ontology Service-based Architecture for Continuous Knowledge Discovery. In: Proc. of the 4th ICMLC 2005. IEEE Press, Guangzhou (2005) 2155-2160

Knowledge Reduction in Inconsistent Decision Tables

Qihe Liu, Leiting Chen, Jianzhong Zhang, and Fan Min

College of Computer Science and Engineering, University of Electronic Science and
Technology of China, Chengdu 610054, China

{qiheliu, richardchen, jianzhong, minfan}@uestc.edu.cn

Abstract. In this paper, we introduce a new type of reducts called the λ -Fuzzy-Reduct, where the fuzzy similarity relation is constructed by means of cosine-distances of decision vectors and the parameter λ is used to tune the similarity precision level. The λ -Fuzzy-Reduct can eliminate harsh requirements of the distribution reduct, and it is more flexible than the maximum distribution reduct, the traditional reduct, and the generalized decision reduct. Furthermore, we prove that the distribution reduct, the maximum distribution reduct, and the generalized decision reduct can be converted into the traditional reduct. Thus in practice the implementations of knowledge reductions for the three types of reducts can be unified into efficient heuristic algorithms for the traditional reduct. We illustrate concepts and methods proposed in this paper by an example.

1 Introduction

The rough set theory is a relatively new mathematical approach to uncertain and vague data analysis. Since firstly introduced by Pawlak in the 1980's, it has been applied in many fields such as machine learning, data mining, and expert systems[1].

The knowledge reduction, a basic concept in the rough set theory, is to remove superfluous attributes from information systems (information tables or decision tables) while preserving the consistency of classifications[1]. The knowledge reduction is performed in information systems by means of the notion of a reduct based on a specialization of the general notion of independence[2][3]. Many types of reducts have been proposed and researched in the rough set community[2][3][4][5][6][7], each of which aimed at some basic requirements. The traditional reduct which preserve positive regions was proposed by Pawlak [1]. the α -reduct and the β -reduct were proposed and studied in the literature [4] and [5] respectively. The α -reduct allows the occurrence of additional inconsistency controlled by the parameter α [4]. The β -reduct is studied in the variable precision rough set model and the parameter β is used to tune reduction precision[5]. In inconsistent decision tables, Kryszkiewicz[7] compared five notions of the knowledge reduction. In fact, only the generalized decision reduct and the distribution reduct are needed because the others are equivalent to one of them respectively. The distribution reduct requires decision vectors are

not changed after performing the knowledge reduction. Zhang[2][3] proposed the maximum distribution reduct which preserves all maximum decision rules.

In essence, in inconsistent decision tables, the distribution reduct preserves the degree of confidence in which every object belongs to each decision classes. It is a harsh requirement and easily affected by noises in data. The maximum distribution reduct and the generalized decision reduct eliminate this requirement to some extent. However, like the distribution reduct, they are not flexible. On the other hand, tuning parameters to find reducts with precision level is emphasized in the rough set theory (e.g. the α -reduct and the β -reduct). Thus, for some specific applications such as interactive data mining problems, it is necessary to find a new kind of reducts with parameters to tune the similarity precision level of decision vectors.

In this paper, the cosine distance between decision vectors is defined. And based on it, the fuzzy similarity relation and the corresponding fuzzy equivalent relation are obtained. A λ -level set of the fuzzy equivalent relation is used to define decision values of objects in the universe and the corresponding traditional reduct is called the λ -Fuzzy-Reduct. By tuning the parameter λ , one can obtain a satisfactory result and the harsh requirement of the distribution reduct is eliminated at the same time. It should be noted here that this concept is totally different from Fuzzy-Rough and Rough-Fuzzy models[8][9]. Rough approximation of fuzzy sets or fuzzy information systems are investigated in these models, but fuzzy relations proposed in this paper are used to introduce a new type of reducts.

In the literature [2], the judgement theorems and discernibility matrices for the distribution reduct, the maximum distribution reduct and the generalized decision reduct were proposed. But knowledge reductions for the three types of reducts based on discernibility matrices are not efficient when dataset is large. In this paper, we present a method which converts the three types of reducts into the traditional reduct. By utilizing efficient heuristic algorithms for the latter, the computational cost of the former can be efficiently reduced.

The rest of this paper is organized as follows. In Section 2 we enumerate relative concepts of the rough set theory and introduce some types of reducts. In Section 3 the λ -Fuzzy-Reduct is proposed, and In Section 4 we present a method to compute the distribution reduct, the maximum distribution reduct and the generalized decision reduct efficiently. We illustrate the method and the concept proposed in this paper by an example in Section 5. Finally, we conclude the paper in Section 6.

2 Information Systems and Knowledge Reduction

Formally, an information system is a pair $S = (U, A)$, where U is a non-empty set of objects called the universe and A is a non-empty set of attributes. With every attribute $a \in A$, V_a , the set of its values is associated. V_a is called the domain of a . In an information system, if we distinguish two disjoint classes of attributes, called condition and decision attributes, respectively, then the information sys-

tem is called a decision table and is denoted by $S = (U, A, D)$, where A and D are called condition attributes set and decision attributes set respectively[1].

Any subset B of A determines a binary relation $I(B)$ on U , which is called an indiscernibility relation, and is defined as follows: $(x, y) \in I(B)$ if and only if $a(x) = a(y)$ for every $a \in B$, where $a(x)$ denotes the value of attribute a for object x . Obviously $I(B)$ is an equivalence relation. The family of all equivalence classes of $I(B)$, i.e., a partition determined by B , is denoted by $U/I(B)$, or simply by U/B ; an equivalence class of $I(B)$, i.e., a block of the partition U/B which contains x is denoted by $B(x)$ [1].

In a given information system $S = (U, A)$, let $X \subseteq U, B \subseteq A$. One can characterize X by a pair of lower and upper approximation sets which are defined as follows:

$$\underline{B}(X) = \bigcup_{x \in U} \{B(x) : B(x) \subseteq X\}, \tag{1}$$

$$\overline{B}(X) = \bigcup_{x \in U} \{B(x) : B(x) \cap X \neq \emptyset\}. \tag{2}$$

The lower approximation set $\underline{B}(X)$ contains those objects in U that can be certainly classified to X , whereas the upper approximation set $\overline{B}(X)$ contains those objects in U that can be possibly classified to X .

Let $S = (U, A, D)$ be a decision table, $B \subseteq A$, and let $U/D = \{D_1, D_2, \dots, D_n\}$. A positive region of the partition U/D with respect to B , denoted by $POS_B(D)$, is defined as follows :

$$POS_B(D) = \bigcup_{i=1}^n \underline{B}(D_i). \tag{3}$$

The positive region $POS_B(D)$ contains those objects that can be certainly classified to some decision classes. If $POS_B(D) = U$, then the decision table S is consistent, otherwise it is inconsistent.

For any $x \in U, B \subseteq A$, a decision vector $\mu_B(x)$ is defined as follows:

$$\mu_B(x) = (D(D_1/B(x)), D(D_1/B(x)), \dots, D(D_n/B(x))), \tag{4}$$

where

$$D(D_i/B(x)) = \frac{|D_i \cap B(x)|}{|B(x)|}, i = 1, 2, \dots, n. \tag{5}$$

Obviously, $B(x) \in [0, 1]^n$, and $\mu_B(x)$ is a probability distribution on U/D . The maximum decision function $\gamma_B(x)$ is defined as follows:

$$\gamma_B(x) = \{D_k : D(D_k/B(x)) = \max\{D(D_1/B(x)), \dots, D(D_n/B(x))\}\}. \tag{6}$$

Obviously, $\gamma_B(x)$ contains those decision classes that x can be classified to them with the maximum degree of confidence.

For any $x \in U$, the generalized decision function $\partial_B(x)$ is defined as follows:

$$\partial_B(x) = \{D_k : D(D_k/B(x)) > 0, k \in \{1, 2, \dots, n\}\}. \tag{7}$$

$\partial_B(x)$ contains those decision classes that x can be classified to them with positive degree of confidence.

Let $S = (U, A, D)$ be a decision table, $B \subseteq A$. Some notions of the knowledge reduction are defined as follows[2]:

Definition 1. (1) If $POS_B(D) = POS_A(D)$, then B is a traditional consistent set of S . If B is a traditional consistent set and no proper subset of B is traditional consistent, then B is called a traditional reduct of S .

(2) If $\mu_B(x) = \mu_A(x)$ for any $x \in U$, then B is a distribution consistent set of S . If B is a distribution consistent set and no proper subset of B is distribution consistent, then B is called a distribution reduct of S .

(3) If $\gamma_B(x) = \gamma_A(x)$ for any $x \in U$, then B is a maximum distribution consistent set of S . If B is a maximum distribution consistent set and no proper subset of B is maximum distribution consistent, then B is called a maximum distribution reduct of S .

(4) If $\partial_B(x) = \partial_A(x)$ for any $x \in U$, then B is a generalized consistent set of S . If B is a generalized consistent set and no proper subset of B is generalized consistent, then B is called a generalized decision reduct of S .

A traditional reduct is a minimum subset of the condition attribute set that preserves the positive region; a distribution reduct is a minimum subset of the condition attribute set that preserves the degree in which every object belongs to each decision class; a maximum distribution reduct is a minimum subset of the condition attribute set that preserves all decision classes that x can be classified to them with the maximum degree of confidence. Similarly, a generalized decision reduct is a minimum subset of the condition attribute set that preserves all decision classes that x can be classified to them with the positive degree of confidence. In the latter two cases, the degree of confidence may not be equal to the original one. The possible reduct is proposed and studied in literature [2] and [7]. Actually, it is easy to prove the fact that the possible reduct is equivalent to the generalized decision reduct. Hence, in this paper we only discuss the latter.

3 λ-Fuzzy-Reducts

According to Definition 1, traditional reducts preserve positive region, but do not consider contradictory objects. Hence, after performing knowledge reductions for the traditional reduct, we can obtain certain rules in inconsistent decision tables. On the contrary, we can obtain uncertain rules if knowledge reductions for the other three types of reducts are performed. For examples, maximum distribution reducts preserve all decision rules with the maximum degree of confidence, and generalized decision reducts preserve all decision rules with the positive degree of confidence.

By Definition 1, distribution reducts preserve the degree of confidence in which every object belongs to each decision classes, but it is a harsh requirement. For example, in a decision table $S = (U, A, D)$, $x, y \in U$, we assume $A(x) = (0.2, 0, 03, 0.5)$, $A(y) = (0.2, 0.001, 03, 0.499)$. According to Definition 1, since

decision vectors $A(x) \neq A(y)$, condition attributes in every distribution reduct must discern objects x and y . Namely, $B(x) \neq B(y)$ for any distribution reduct B . In fact, $A(x)$ is very close to $A(y)$ and the difference between them is trivial. It is possible that noise data in the decision table S cause the small difference. In this situation, decision vectors $A(x)$ and $A(y)$ should be considered identical. Thus, we need a similarity measure for decision vectors.

Definition 2. Let $x = (x_1, x_2, \dots, x_m)$ and $y = (y_1, y_2, \dots, y_m)$ be m -dimension vectors, the cosine-distance between x and y is defined as follows:

$$COS(x, y) = \frac{x \bullet y}{|x||y|}, \tag{8}$$

where $x \bullet y = \sum_{i=1}^m x_i y_i$, $|x| = \sqrt{\sum_{i=1}^m x_i^2}$, $|y| = \sqrt{\sum_{i=1}^m y_i^2}$.

Obviously, from the viewpoint of geometry, $COS(x, y)$ denotes the cosine distant between x and y , and it is often used as a similarity measure. For example, in documents classification, researchers utilize the cosine distance to measure similarities between documents and obtain satisfactory qualities of classification[10].

Similarly, in a decision table $S = (U, A, D)$, for two decision vectors $A(x)$ and $A(y)$, $COS(A(x), A(y))$ can measure the similarity between the two decision vectors. Essentially, $COS(A(x), A(y))$ may be treated as a function $U^2 \rightarrow [0, 1]$, so it is a fuzzy relation on U . For the simplicity, we denote $COS(A(x), A(y))$ as $F(x, y)$, then F is a fuzzy relation on U .

According to Definition 2, for any $x, y \in U$, we have $F(x, x) = 1$, and $F(x, y) = F(y, x)$. So the following theorem can be obtained.

Theorem 1. Let $S = (U, A, D)$ be a decision table, F is a fuzzy similarity relation on U .

By Theorem 1, level sets of F are similarity relations on U . On the other hand, according to Definition 1 and Theorem 2, in the rough set theory, the knowledge reduction is based on equivalence relations. Hence, we need a fuzzy equivalence relation here. Usually, a fuzzy equivalence relation can be obtained by computing transitive closure of a given fuzzy similarity relation [11].

Lemma 1. In decision tables $S = (U, A, D)$, the transitive closure of F is

$$TF = F^k, \tag{9}$$

where $k > |U|$, $F^k = F^{k-1} \circ F$, and “ \circ ” is the composite operation[11].

Thus, the following Theorem is obtained.

Theorem 2. Let $0 < \lambda < 1$, λ -level set of TF , denoted by TF_λ , is defined as follows:

$$TF_\lambda = \{(x, y) : TF(x, y) \geq \lambda, x, y \in U\}, \tag{10}$$

and then TF_λ is an equivalent relation on U .

A partition induced by TF_λ is denoted by U/TF_λ . For $x \in U$, a block of the partition U/TF_λ , containing x , is denoted by $[x]_\lambda$. If $y, z \in [x]_\lambda$, then decision vectors $A(y)$ is similar to decision vectors $A(z)$ with the degree of the similarity greater than or equal to the parameter λ . The partition U/TF_λ can be treated as an attribute $\{d_\lambda\}$. Objects in a block have same attribute values, but objects in different blocks have different attribute values. Let $U/TF_\lambda = \{[x_1]_\lambda, [x_2]_\lambda, \dots, [x_m]_\lambda\}$, we define the attribute value as follows:

$$\forall x \in U, \text{ if } x \in [x_i]_\lambda, \text{ then } d_\lambda(x) = i.$$

Obviously, $U/TF_\lambda = U/\{d_\lambda\}$.

Definition 3. Let $0 < \lambda < 1$, decision tables $S = (U, A, D)$, $S_\lambda = (U, A, \{d_\lambda\})$. Traditional reducts of S are called λ -Fuzzy-Reducts of S .

If the parameter $\lambda = 1$, then $U/d_\lambda = U/A$. Hence, 1-Fuzzy-Reducts of S are distribution reducts of S , which indicates the λ -Fuzzy-Reduct are an extension of the distribution reduct. Similarities between decision vectors are considered in the definition 3, and the parameter λ is used to control the degree of similarities. In applications, the parameter λ may be a user-specified threshold. In interactive data mining, user can obtain satisfactory results by tuning the parameter λ . So the λ -Fuzzy-Reduct eliminate the harsh requirements of the distribution reduction. And it is more flexible than the maximum distribution reduct, the traditional reduct, and the generalized decision reduct.

4 Implementations of the Knowledge Reduction

If a decision table $S = (U, A, D)$ is consistent, the four types of reducts defined in Definition 1 are identical. However, in inconsistent decision tables, they are different with each other. Zhang[2][3] researched the relationships among the distribution reduct, the maximum distribution reduct and the generalized decision reduct, and the judgement theorems and discernibility matrices associated with the three types of reducts were proposed. Hence, we can employ approaches based on them to the three types of reducts in inconsistent decision tables. But the computational cost of these approaches is high if data set is large. Thus, the implementations of the knowledge reduction based on the above strategy are not efficient and scalable for data mining applications where data sets are large in most cases. On the other hand, researchers have presented some efficient and scalable methods for the traditional reduct[1][12]. Hence, if the three types of reducts can be converted into the traditional reduct, we can obtain efficient and scalable methods for them. Furthermore, by Definition 3, a λ -Fuzzy-Reduct is a traditional reduct of $S = (U, A, \{d_\lambda\})$, so this approach is used to compute λ -Fuzzy-Reducts efficiently. In order to illuminate this conversion, we introduce the following Lemma.

Lemma 2. Let $S = (U, A, D)$ be a decision table, $B \subseteq A$, we have

$$(1) \forall x \in U, \mu_B(x) = \mu_A(x) \iff \forall x \in U, y \in B(x), \mu_A(y) = \mu_A(x);$$

- (2) $\forall x \in U, \gamma_B(x) = \gamma_A(x) \iff \forall x \in U, y \in B(x), \gamma_A(y) = \gamma_A(x);$
- (3) $\forall x \in U, \partial_B(x) = \partial_A(x) \iff \forall x \in U, y \in B(x), \partial_A(y) = \partial_A(x).$

Proof: Firstly, correctness of the proposition (1) is proofed as follows:

" \implies ": $\forall y \in B(x), \mu_B(y) = \mu_A(y)$. By definition of $\mu_B(y)$, we have $\mu_B(y) = \mu_B(x)$, and $\mu_B(x) = \mu_A(x)$, so $\mu_B(y) = \mu_A(x)$. Hence, we obtain $\mu_A(y) = \mu_A(x)$.

" \impliedby ": $B \subseteq C$, we have $B(x) = \bigcup_{i=1}^m C(y_i)$. For $D_i \in U/D$, we have

$$D(D_i/B(x)) = \sum_{j=1}^m \frac{|D_i \cap C(y_j)|}{|C(y_j)|} \frac{|C(y_j)|}{|B(x)|}. \tag{11}$$

On the other hand, $1 \leq j \leq m, y_j \in B(x), \mu_C(y_j) = \mu_C(x)$, so $1 \leq j, k \leq m$, we have $\mu_C(y_j) = \mu_C(y_k)$, so

$$\frac{|D_i \cap C(y_j)|}{|C(y_j)|} = \frac{|D_i \cap C(y_k)|}{|C(y_k)|}. \tag{12}$$

Hence,

$$D(D_i/B(x)) = \frac{|D_i \cap C(y_1)|}{|C(y_1)|} \sum_{j=1}^m \frac{|C(y_j)|}{|B(x)|} = \frac{|D_i \cap C(x)|}{|C(x)|} = D(D_i/C(x)). \tag{13}$$

According to Equation 13, we have $\mu_B(x) = \mu_C(x)$.

Similarly, we can proof correctness of propositions (2) and (3).

According to Lemma 2, we obtain a theorem as follows:

Theorem 3. *Let $S = (U, A, D)$ be a decision table, $B \subseteq A, C \subseteq A$, we have*

- (1) $\forall x \in U, \mu_B(x) = \mu_A(x) \implies \forall x \in U, \mu_A(x) = \mu_A(x);$
- (2) $\forall x \in U, \gamma_B(x) = \gamma_A(x) \implies \forall x \in U, \gamma_A(x) = \gamma_A(x);$
- (3) $\forall x \in U, \partial_B(x) = \partial_A(x) \implies \forall x \in U, \partial_A(x) = \partial_A(x).$

Definition 4. *Let $S = (U, A, D)$ be a decision table, denoted by $S_\mu = (U, A, D = \{\mu_A\})$, $S_\lambda = (U, A, D = \{\lambda_A\})$, and $S_\partial = (U, A, D = \{\partial_A\})$. S_μ, S_λ , and S_∂ are called the distribution decision table, the maximum distribution decision table, and the generalized decision table, respectively.*

For any $y \in A(x)$, we have $\mu_A(y) = \mu_A(x), \lambda_A(y) = \lambda_A(x)$, and $\partial_A(y) = \partial_A(x)$, so S_μ, S_λ , and S_∂ are consistent decision tables. By Lemma 2 and Theorem 3, we have a theorem as follows:

Theorem 4. *Let $S = (U, A, D)$ be a decision table, $B \subseteq A$, we have the fact that B is a traditional reduct of $S_\mu(S_\lambda$ or $S_\partial)$ if and only if B is a distribution (maximum distribution or generalized decision) reduct of S .*

Theorem 4 indicates a strategy for the implementation of the knowledge reduction for the distribution (maximum distribution or generalized decision) reduct

as follows. Firstly, obtaining $S_\mu(S_\lambda$ or $S_\partial)$ by the decision table S . Secondly, computing the traditional reduct of $S_\mu(S_\lambda$ or $S_\partial)$. Obviously, the time complexity of the first step is $O(mn)$, where m and n , are the cardinalities of attributes and objects in S respectively. The time complexity of the second step depends on the method for the traditional reduct. For example, the time complexity of the second step is $O(m^2nlogn)$ if the method proposed in [12] is applied. Hence, the total time complexity of the knowledge reduction for the distribution (maximum distribution or generalized decision) reduct is

$$O(mn) + O(m^2nlogn) = O(m^2nlogn) \tag{14}$$

The result is the same as the efficient method proposed in [12]. The analysis shows the implementation of the knowledge reduction for the distribution (maximum distribution or generalized decision) reduct proposed in this paper is more efficient than those based on the discernibility matrices.

5 An Example

In this section, we give an example and compare the λ -Fuzzy-Reduct with other types of reducts in Definition 1. A decision table $S = (U, A, D)$ is listed in Table 1, where universe $U = \{t_1 \times 50, t_2 \times 10, \dots, t_{10} \times 90\}$, condition attributes set $A = a, b, c, e, f$, decision attributes set $D = \{d\}$. $t_i \times num$ means that object t_i appears num times in S . Researchers have proposed knowledge reduction algorithms for the traditional reduct based on heuristic information. For example, a knowledge reduction algorithm based on information entropy is presented in [12]. According to Theorem 4 and Definition 4, by using this knowledge reduction algorithm, we can perform knowledge reductions for the λ -Fuzzy-Reduct and all types of reducts in Definition 1.

Table 2 lists computational results, where the following abbreviations are used: TR for *the traditional reduct*, DR for *the distribution reduct*, MDR for *the maximum distribution reduct*, and GDR for *the generalized decision reduct*.

Table 1. A decision Table S

U	a	b	c	e	f	d	Num
t_1	0	0	0	0	1	0	50
t_2	1	1	1	1	1	1	10
t_3	1	1	0	1	1	1	30
t_4	1	1	1	1	1	0	20
t_5	0	0	1	0	1	0	120
t_6	1	1	0	1	0	2	60
t_7	0	1	1	1	1	1	110
t_8	1	1	1	0	1	1	200
t_9	1	1	0	1	1	0	20
t_{10}	0	1	1	1	1	0	90

Table 2. computational results of Table 1

TR	DR	MDR	GDR
$\{a, e, f\}$	$\{a, c, e, f\}$	$\{a, c, e\}$	$\{a, e, f\}$

Table 2 shows the distribution reduct $\{a, c, e, f\}$ only reduces one condition attribute b and it preserves more information of Table 1 than the other types of reducts. By using the distribution reduct $\{a, c, e, f\}$ to generate decision rules, long decision rules is obtained, which means the generalized capability is lower than the other types of reducts[1]. The maximum distribution reduct $\{a, c, e\}$ preserves all decision classes that each object can be classified to them with the maximum degree of confidence. But in table 1, after performing the knowledge reduction, decision values of $t_2, t_9,$ and t_{10} are changed to 0, 1, and 1, respectively. Hence, the maximum distribution reduct is different from the other three types of reducts.

Similarly, by Definition 3, λ -Fuzzy-Reducts of Table 1 are computed, and the results are listed in Table 3 (the parameter is equal to 1, 0.96, 0.94, and 0.8 respectively).

Table 3. λ -Fuzzy-reducts of Table 1

λ	1	0.96	0.94	0.8
Reducts	$\{a, c, e, f\}$	$\{a, c, e, f\}$	$\{a, e, f\}$	$\{f\}$

Table 3 indicates that we can obtain the distribution reduct, the traditional reduct, and the generalized decision reduct when the parameter λ is tuned in the process of the knowledge reduction for λ -Fuzzy-Reducts. In interactive data mining, user can obtain satisfactory reducts and decision rules by controlling the parameter λ . Hence, λ -Fuzzy-Reducts eliminates the harsh requirements of the distribution reducts. And it is more flexible than the other types of reducts. Furthermore, Table 3 shows that λ -Fuzzy-Reducts are sensitive to the parameter λ , so selecting the parameter accurately is very important in practical applications.

6 Conclusions and Future Work

The knowledge reduction in inconsistent decision tables is discussed in this paper.

Firstly, the λ -Fuzzy-Reduct is proposed. The parameter is used to control degree of similarities among decision vectors. It can eliminate the harsh requirements of the distribution reduct, and it is more flexible than the other types of reducts.

Secondly, In order to obtain efficient implementations of knowledge reductions for the distribution reduct, the maximum distribution reduct, and the generalized decision reduct, we present a method to convert these types of reducts into traditional reducts, and then knowledge reductions can be efficiently performed by using efficient algorithms for the traditional reduct.

In the future work, we will further investigate on the λ -Fuzzy-Reduct and utilize this concept in some practical applications such as document classification.

References

1. Pawlak Z, Some issues on Rough Sets. Transactions on Rough Sets I, LNCS 3100(2004) pp. 1-58.
2. Wenxiu Zhang, Jusheng Mi, Weizhi Wu, Approaches to knowledge reductions in inconsistent information systems. International journal of intelligent systems, 18(2003) pp. 989-1000.
3. Zhang Wen-Xiu, Mi Ju-Sheng, Wu Wei-Zhi, Knowledge Reductions in Inconsistent Information Systems. Chinese Journal of Computer, 26/1(2003) pp.12-18.
4. Ziarko W, Variable precision rough set model. Journal of Computer Systems and Science, 46(1993) pp.39-59.
5. Nguyen HS, Slezak D, Approximation reducts and association rules correspondence and complexity results. Proceedings of RSFDGrC'99, Yamaguchi, Japan(1999), LNAI 1711, pp 137-145.
6. Slezak D, Searching for dynamic reducts in inconsistent decision tables. Proceedings of IPMU'98, Paris(1998), pp. 1362-1369.
7. Kryszkiewicz M, Comparative study of alternative type of knowledge reduction in inconsistent systems. International journal of intelligent systems, 16(2001) pp.105-120.
8. S. Nanda, Fuzzy rough sets. Fuzzy Sets and Systems, 45(1992) pp. 157-160.
9. M. Banerjee, S.K. Pal, Roughness of a fuzzy set. Information Science, 93(1996) pp.235-246.
10. Lam W, Ruiz M, Srinivasan P, Automatic text categorization and its application to text retrieval. IEEE Transaction on Knowledge and Data Engineering, 11/6(1999) pp.865-879.
11. YD. Dubois, H. Prade, Fuzzy Sets and Systems-Theory and Applications. Academic Press, New York(1980).
12. Liu Qihe, Li Fan, Min Fan, An efficient knowledge reduction algorithm based on new conditional information entropy. Control and Decision, 20/8(2005)pp.878-882.

Semantic Scoring Based on Small-World Phenomenon for Feature Selection in Text Mining

Chong Huang¹, Yonghong Tian², Tiejun Huang^{1,2}, and Wen Gao^{1,2}

¹ Graduate School, Chinese Academy of Sciences, Beijing 100039, China

² Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
{chuang, yhtian, tjhuang, wgao}@jdl.ac.cn

Abstract. This paper proposes an effective scoring scheme for feature selection in Text Mining, using characteristics of Small-World Phenomenon on the semantic networks of documents. Our focus is on the reservation of both syntactic and statistical information of words, rather than solely simple frequency summarization in prevailing scoring schemes, such as TFIDF. Experimental results on *TREC* dataset show that our scoring scheme outperforms the prevailing schemes.

1 Introduction

In text mining, a scoring scheme weights how much a word can represent the context. Scoring schemes are used, for example, to select representative features (words) of the context. These features can be utilized to extract keywords and summarize documents, or to measure text similarity.

Currently, most scoring schemes are based on statistical or specific structural information. One thing usually lost in these schemes is ordinal information, implying syntactic and semantic information between words. These schemes have drawbacks when applied to a huge number of less structured documents with millions of words, as in a large book archive. Moreover, statistical information is usually set-related. Even if a part of a document changes, the entire data set needs rescoring. In these cases, results of similarity analysis and keyword extraction are far from satisfaction.

The same challenge emerges in our project¹, China-US Million Book Digital Library project [1], with roughly 400,000 digital books amounted to 44.6TB presently. This is the first large international DL cooperation project, aiming at organizing one million Chinese or English books online.

Towards this end, we propose a novel set-independent semantic scoring scheme. This scheme is based on the Small-World Phenomenon (SWP) on complex networks, using both statistical and syntactic information. Moreover, the scoring of each document is independent. To attest the effectiveness of this scheme, we propose a Fully Automatic Metadata-Extracting Algorithm (FAMEA). Experimental results indicate that our scoring scheme returns results in up to 20% rank and set size as in TFIDF.

¹ Funded by the 863 International Cooperation Project of Technology Ministry of China (Project 2003AA119010), CADAL Project Management Center (Project CADAL2004002).
Homepage: <http://www.ulib.org.cn/> and <http://www.cadal.net/>

Section 2 summarizes the current scoring schemes. Section 3 outlines our scoring scheme. Section 4 applies this scheme in keyword extraction of documents. Section 5 describes an experiment on the effectiveness of this scheme.

2 Related Works

Prevailing scoring schemes can be divided into three kinds: freq-based, structural, and information-theoretic. The first class is based on frequency information. Methods can be *Term Frequency (TF)*, *TFIDF*, or *BM25*. They are widely used document relevance analysis [16] and ranking schemes in Information Retrieval (IR). Secondly, some schemes involve specific structure information [17]. Ironically, this information also limits the application of these algorithms. Rooted from *Information Theory*, the last kind puts more emphasis on *Information Gain*, *Mutual Information*, or *Odds Ratio* [16]. The key process is the estimation of priors. The most related work of scoring scheme is the HAL model [18], where a fixed-length sliding window is proposed to capture ordinal information.

Currently, a body of literatures is on the study of complex networks analysis, including some on lexical analysis of words, such as Associative Network [2], WordNet [3], but few use SWP to extract metadata. Zhu et al [4] use SWP to extract keywords of Chinese news web pages, but they apply to a small dataset and lack of consideration of unconnectedness, computational complexity of SWP.

Automatic extraction/assignment of metadata is a hot issue nowadays. Most literatures extract information from tagged documents (HTML files [14, 15]), semi-structured files (papers [17]), but few in extraction from more free text, such as text of books, let alone large eBook archive. Some of them still use semi-automatic approach, called interactive training or annotation. Books possess biggest capacity but least structural info, requiring a more effective extraction algorithm.

3 Semantic Network Model on Small-World Phenomenon (SWP)

Rather than simply as isolated points or linear sequence, we view the original text as a semantic network – words as vertices and relationship between vertices as edges. We can score the nodes in these networks by studying the structure of the networks. To study the infrastructure of these networks, the *SWP* can be a powerful tool. The *SWP*, also known as the Six Degree Separation Rule, is found to be a common phenomenon in complex networks: networks with large amount of vertices still have small average minimum path length L , and high clustering coefficient C [8]. Concluded from Table 1, *SWP* happens in our semantic network model.

There are two ways to establish relationship between words: Structuralist Semantics and Co-occurrences. Structuralist Semantics builds networks on the grounds that neighboring words have syntactic or semantic relationship [5, 6], while Co-occurrence presumes that co-occurrence in some linguistic units (chapter, discourse, paragraph, and sentence) or in a fixed distance is semantically indicative [4, 7, 18]. Because of the huge size of our data set, we choose syntactical relationship, constructing a more convincing and concise semantic network than co-occurrence.

As described in [6], 70% of dependencies are between neighboring words, 17% at a distance of 2. Cancho et al [5] summarize all syntactical relation within distance 2. Interestingly, we find that after StopWord-filtering (most adverbs, articles, prepositions, and modals are omitted), nearly all syntactical relation at a distance of 2 is shortened to 1. Because of this, FAMEA considers only neighboring relationship in the same sentence to establish relationship between words. This relationship holds several properties: transitivity, undirected and uniformly weighted.

Table 1. Stats of dynamics 41100 semantic networks, sampled from 3707 eBook in our data set, where m stands for the number of edges, n for vertices, and CC for Connected Components

X	L	C	n	m	n ² /m	CC
EX	4.240	0.653	565.3	1219.4	262.10	1.207
DX	1.164	0.067	625.4	1940.0	481.78	0.763

Our scoring scheme relies on several dynamics as defined bellow, with the help of some traditional dynamics, such as TF, degree to filter (see in Section 4).

Characteristic Path Length (average shortest path length) L [8]. $L(v)$ denotes the average length of shortest paths started from vertex v , and L is the average of $L(v)$ among all vertices in the network. They are defined as:

$$L(v_i) = \frac{\sum_{v_j \in V} d(v_i, v_j)}{n - 1},$$

$$L = \frac{\sum_{v_i \in V} L(v_i)}{n}.$$

Clustering Coefficient C [8]. $C(v)$ depicts how fully connected vertex v and all its neighbors behave, and C averages $C(v)$ in the scope of all vertices. They are defined as bellow, where R_i is the number of pair-bonding links among v_i and its l neighbors,

$$C(v_i) = \frac{2R_i}{l(l-1)},$$

$$C = \frac{\sum_{v_i \in V} C(v_i)}{n}.$$

Efficiency E [9]. E measures how efficiently information is exchanged over the network: the longer L is, the lower E becomes. They are defined as:

$$E(v_i) = \frac{1}{n-1} \sum_{v_j \in V} \frac{1}{d(v_i, v_j)},$$

$$E = \frac{\sum_{v_i \in V} E(v_i)}{n}.$$

Traffic $T(v)$ [10](*Betweenness Centrality* $B(v)$ [11]). $T(v)$ sums the number of trajectories passing through vertex v , and so identifying the most active hubs, while $B(v)$ measures the number of shortest path length passing through vertex v . Since shortest paths contribute more to the tightness of networks than usual paths, $B(v)$ defeats $T(v)$.

To figure out a dynamic to weight vertices, Watts and Strogatz [8] define a key vertex as a *shortcut* in a *Growth Model* (a network grows from a node to a graph): shortcuts are vertices that decrease L drastically. Keywords in our model conduct similarly, but are viewed in a *Decadence Model*, because we care the status quo more than the evolving process of the network. If these key vertices are removed, L soars, and even the network collapses. In consideration of unconnectedness of our semantic network, we define L as a “harmonic mean” geodesic distance (a variation from [11]). Note that distances of cycles are excluded to avoid infinity (while in [11] they are included):

$$L' = 1 / E$$

$$L'^{-1} = \frac{1}{\frac{1}{2}n(n-1)^{i \neq j}} \sum d_{ij}^{-1} .$$

Actually, L' is a variant harmonic mean of $L(v)$. According to our experiment results, this definition unexpectedly performs slightly better ΔL_v than L . Therefore, we define ΔL_v as

$$\Delta L_v = L' - L'_v, \text{ where } L'_v \text{ denotes } L' \text{ after removing node } v ,$$

and to measure the importance of words in different contexts, we define this score function $S(v)$ of vertex v as:

$$S(v) = \frac{\Delta L_v}{L'}$$

In a word, there are three main assumptions in our semantic network model on SMP: (1) words in a document can characterize it; (2) neighboring in the same sentence after filtering depicts syntactic relationship between words concisely and convincingly; (3) the tighter a semantic network connects, the greater the SWP emerges, indicating a more conspicuous topic. The likelihood of summarizing a text with a word lies in the role it plays in the elasticity and cliquishness of the entire semantic network.

4 Fully Automatic Metadata-Extraction Algorithm (FAMEA)

Based on this semantic network model, we can easily weight the importance of each word in the context and select features. There are three usages of these features. First in ranking schemes and the like, use them as property values to calculate the similarity as in VSM model. Secondly, reduce the dimension of the context, generate a feature set, and categorize it. Finally, extract keywords to represent the context.

To implement this scheme, there are three main challenges. First, the computational complexity of calculating L is a NP-problem. In graph theory, the most famous and efficient algorithm to summarize lengths of all shortest paths is *Floyd Algorithm*,

which takes $O(n^3)$. It will be an intolerable experience for one to extract metadata from a book comprising millions of words. Furthermore, we need to design an effective data structure to store millions of vertices and edges. Finally, unconnectedness is an easily ignored issue in Complex Network Analysis, confining the use of L .

However, these networks possess several important features: (1) Sparsity. Given n vertices, a graph can have maximum n^2 edges, but m , the number of edges in networks with SWP is far less than n^2 , that is $m \ll n^2$. (2) Uniformly weighted and undirected. FAMEA reduces the computational complexity effectively, using these features.

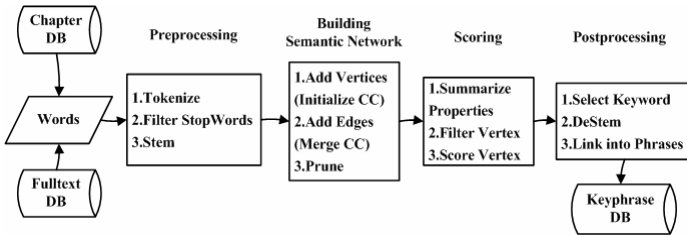


Fig. 1. Workflow of FAMEA. With inputs of contexts, FAMEA outputs extracted keywords.

FAMEA starts in splitting context into words. Before adding words into semantic network, FAMEA omits the words in StopList, including letters, adverbs, prepositions, auxiliary verbs, articles, pronouns. Caldeira et al [7] have proved that this filtering treatment does not modify general behaviors of the network. Since most source books are in English, FAMEA uses PorterStemmer [12] to map words to their stems.

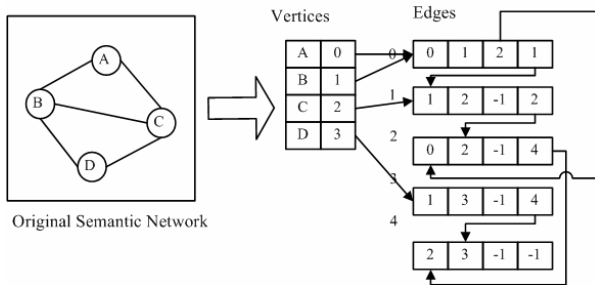


Fig. 2. Structure of Adjacent Multilist

After preprocessing, context becomes a sequence of filtered word, and neighboring words in one sentence have syntactical relationship. FAMEA stores vertices and edges in Adjacent Multilist, to reduce computational complexity of space. Since sparsity of SWP, $m \ll n^2$, the possibility of memory overflow plummets.

Then to avoid unconnectedness, FAMEA initializes each vertex as a connected component. When an edge is introduced and this pair of vertices is in different components, the component in smaller size will be merged into the other. Facts turn out that most semantic networks in our project are really unconnected. To facilitate the

summarization of L , some researchers [4] resort to weighted summary of all connected components, but the tuning of weights will be state-of art. We observe that in our data, most networks have a main connected component with more than 95% vertices of it (shown in Table 1). Steyvers and Tenenbaum [13] have the same discovery in WordNet and Roget's thesaurus, up to even 99% of 22,000 and 29,000 vertices, and conclude this as a main property of SWP. Note that in statistics in Table 1, only 11.5% has a CC more than 1, and 96.9% among which have m/n less than 2, that is, they are approximately linear contexts. Henceforth, most of the semantic networks can be pruned into a main branch. As a result, FAMEA prunes the branches with vertices bellow 5%.

Though calculating L is an NP-problem, Semantic Network model here possesses several important features: (1) Sparsity: $m < n^2$. (2) Unweighted and undirected. Noticing these features, by breath-first searching through Adjacent Multilist, FAMEA effectively reduces the computational complexity from $O(n^3)$ to $O(mn)$, and the best time cost for summarizing L will be $k \frac{n(n+1)}{2}$, where k is the average time cost for a distance. It is common that a context has a vocabulary of millions of words, yet not all of them need a further investigation. Note that the chances are very minute that words with very low TF or degree are keywords. We discover that most keywords have TF ranking above 25% of all, obeying the *20-80 Rule* in statistics, and FAMEA filters vertices lower than this.

Destemming is an ill-posed problem (one-to-many mapping). Currently FAMEA destems words by searching words with the stem and picks up the word with highest TF or is probable involved in a keyphrase, and partially stems it out of forms like plural form of nouns and -ing, -ed tense forms of verbs. FAMEA ranks sequential words on the basis of their harmonic mean of scores, and links some top percent of them. We define the score of a word sequence with length n as:

$$S(v_1, \dots, v_n) = \frac{n}{\sum_{k=1}^n \frac{1}{S(v_k)}} .$$

Humphreys [14] has proved this equation to be reasonable, yet there is one more rationale in our case. Note that in scoring result (Section 3), harmonic mean definition of ΔL , outperforms algebraic mean one, as well as phrase linking. This might be more than coincidence. After mapping the candidate phrases back to context, if it really has PF (Phrase Freq.) higher than a threshold, it will be selected as a keyphrase.

5 Experiments

Since it is hard to attest the effectiveness of the scoring scheme directly, we carry out experiments on FAMEA. Because of the absence of authoritative test set for keyword extraction, we spilt the task into two: one to attest the result in a single book, and the other in information retrieving of *AQUAINT* (a news document set included in *TREC*).

Experiment 1. Attest the individual result of feature extraction in eBooks.

Table 2 bellow is an example result of automatically extracted keyword of FAMEA. The title of this book is “*Price responsiveness of world grain market*”, and the authoritatively assigned subjects by the Library of Congress in USA are “Grain trade”, “Intervention (Federal government)”, and “Elasticity (Economics)”. All keywords bellow are top 7 in score of the entire book. We can witness the keywords share some semantic similarity with the title and subject field of the book.

Table 2. Simplified Example Result of FAMEA. Figures in the first two columns are bookID.

country	import	government	price	economy	variability	elasticity
0.06802	0.05995	0.04110	0.03257	0.03217	0.03194	0.02664

Experiment 2. Attest the overall performance when applied to retrieve in *AQAIN T*.

Data Generation. Since the aim of *TREC* is to test the availability of information rather than summarization, we choose the highest related topics and documents, though it is still challenging to match poorly performed topics of the robust track. Our test set includes 50 topics from *TREC05 Robust Track* (having a matching score at 2 in the test set), and 2098 corresponding documents from *AQAIN T* (including *NYT*, *APW*, and *XIE*). As a baseline, we implement FAMEA with the most popular scoring scheme TFIDF, and we denote SW as our scoring scheme.

Evaluation Measure. To evaluate the improvement, we retort to three measures: the ranking of the query word in the feature set, the normalized weight of the query word, and the size of this feature set. As for the queries that are missing from the feature set, we grant them the size of the set as its ranking and a zero value weight. We use the normalization factor as in vector space,

$$S'(v_i) = \frac{S(v_i)}{\sqrt{\sum_{k=1}^n S^2(v_k)}}$$

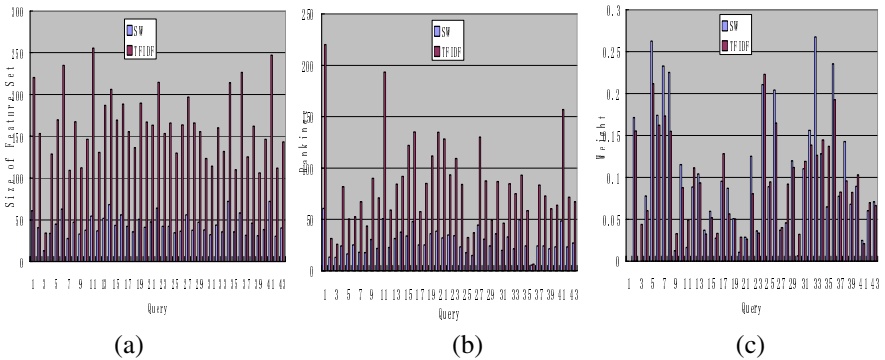


Fig. 3. Query-averaged results of Experiment 2. (a)Comparison of the size of feature set. (b)Comparison of rankings of queries. (c)Comparison of weights of query words. The left and lighter-colored column is SW, and the right and darker one is TFIDF.

Results. Results are shown in the three figures above. We can witness great improvements in rankings (usually SW is 1/2 to 1/5 of TFIDF) and sizes of feature set (TFIDF is usually 5 times of SW). In addition, in most cases, weights of SW are far bigger. In a word, SW greatly improves the performance of the ranking and scoring.

6 Outlook

We are currently working on setting up benchmarks or testbeds for keyword extraction. We are also moving to applying this semantic scoring scheme and FAMEA to other fields, such as text categorization and eBook retrieval systems.

References

1. Huang, T., Tian, Y., et al. Towards a multilingual, multimedia and multimodal digital library platform. *J. Zhejiang Univ. SCI* 2005 6A(11):1188-1192.
2. Nelson, D.L., McEvoy, C.L., & Schreiber, T.A. (1999). The University of South Florida word association norms. <http://www.usf.edu/FreeAssociation>.
3. Fellbaum, C. (Ed.) (1998). *WordNet, an electronic lexical database*. MIT Press.
4. Zhu, M., Cai, Z., Cai, Q., Automatic Keywords Extraction Of Chinese Document Using Small World Structure. In *Proc.s of IEEE ICNLPKE'03*, 2003.
5. I Cancho, R.F., Sole, R. The small world of human language. *Proc. R. Soc. London B*, in press. Also Santa Fe Institute working paper 01-03-016.
6. Lyon, C., Nehaniv, C., Dickerson, B. Entropy Indicators for Investigating Early Language Process. <http://homepages.feis.herts.ac.uk/~comrcml/>
7. Caldeira, S., Lobao, T., et al. The Network of Concepts in Written Texts. (<http://arxiv.org/pdf/physics/0508066>)
8. Watts, D., Strogatz, S., Collective dynamics of small-world networks, *Nature* 393,440 (1998).
9. Latora, V., Marchiori, M. Efficient Behavior of Small-World Networks. *Phys. Rev. Lett.*, 87 (2001), art. No. 198701.
10. Sigman, M., Cecchi, G. Global organization of the Wordnet lexicon. *PNAS*, USA, 99 (2002), pp. 1742-1747.
11. Newman, M. The structure and function of networks, *Comput. Phys. Comm.*, 147(2002), pp. 40-45.
12. Porter, M. The Porter Stemming Algorithm. (<http://www.tartarus.org/~martin/PorterStemmer> 2005)
13. Steyvers, M., Tenenbaum, J. The Large-Scale Structure of semantic networks: Statistical Analyses and a Model for Semantic Growth, 2001. (<http://arxiv.org/abs/cond-mat/>)
14. Humphreys, J. PhraseRate: An HTML Keyphrase Extractor. Technical report, University of California, Riverside. June 2002. <http://infomine.ucr.edu/>
15. Hu, Y., Xin, G., et al. Title extraction from bodies of HTML documents and its application to web page retrieval. In *Proc. of SIGIR'05*, Salvador, Bahia, Brazil, Aug. 2005.
16. Yang, Y., Pedersen, J.O. A Comparative Study on Feature Selection in Text Categorization In *Proc. of the 14th ICML97*, pp. 412---420, 1997
17. Giuffrida, G., Shek, E., Yang, J. Knowledge-based metadata extraction from PostScript files. In *Proceedings of Fifth ACM Conference on Digital Libraries*, 2000.
18. Song, D., and Bruza, P.D. (2003) Towards Context-sensitive Information Inference. *JASIST*, 54(4), pp. 321-334.

A Comparative Study on Text Clustering Methods*

Yan Zheng^{1,2}, Xiaochun Cheng^{2,3}, Ronghuai Huang², and Yi Man¹

¹ School of Computer Science and Technology
Beijing University of Posts and Telecommunications, 100876 Beijing, China
yanzheng@bupt.edu.cn

² Knowledge Science and Engineering Institute
Beijing Normal University, 100875 Beijing, China

³ Middlesex University, UK

Abstract. Text clustering is one of the most important research areas in text mining, which handles the text automatically to discover implicit knowledge. It groups text into different clusters by contents without apriori knowledge. In this paper, different text clustering methods are studied and three text clustering validation criteria are studied and used to evaluate the experimental results. We compare and contrast the effectiveness of k-means and FIHC text clustering methods by experiments, and address the different levels of quality of the resulting text clusters.

1 Introduction

Text mining has attracted more and more researchers [1-15]. Approximately 90% online data are in text format. Text mining has been applied for classifying news and stories according to the contents, filtering spam emails [1], organizing repositories of document-related meta-information for efficient search and retrieval [2], clustering documents or web pages, discovering trends or relations among entities etc. Text mining is related to a number of research areas, such as natural language processing, computational linguistics, statistics and pattern recognition. Unlike in text classification, in text clustering no apriori knowledge is provided. Popular algorithms applied to text clustering include Frequent Itemset-based Hierarchical Clustering (FIHC) [3], k-means method [4], Self Organization Mapping (SOM) [5] and Support Vector Machine (SVM). Hierarchical clustering (also called as systematic clustering) pairs the most similar clusters and organizes all clusters into a category tree. The partitioning clustering method and the hierarchical clustering method are different. The text is clustered into the same level in the former; while the text is partitioned into the nesting clusters in the latter. In addition, according to different representations of the text, text clustering methods are classified into following three categories: Word-based clustering, Knowledge-based clustering and Information-based clustering. We explain more details in following.

* The research is funded by National Natural Science Foundation of China (Project No. 60402011) and Ministry of Education Key Laboratory of Information Management and Information Economics (Project No. F0607-01).

1.1 Word-Based Clustering

In general, concept is the basic unit for the automatic text processing. Concepts are consisted of words. Word is the atomic carrier of information, which can't be decomposed. It is feasible to represent the document by the key words. Selecting an efficient word segment method for preprocessing the documents is necessary, and extracting representative key words is important for the word-based clustering method. Vector Space Model (VSM) is a very popular method of representing the document, which each document is represented as a vector (a series of words). The performance of word-based clustering method will decline due to the high dimensionality and sparseness [6].

1.2 Knowledge-Based Clustering

Knowledge-based clustering mainly depends on an explicit knowledge base. The knowledge representations include semantic webs, predicates, objects, rough set based constructs, neural networks etc. For knowledge-based clustering method, a knowledge base with strong specialty must be constructed manually, which is unlikely to be portable. For specific applications, knowledge-based clustering method can perform the text clustering quickly and accurately.

1.3 Information-Based Clustering

Information-based clustering is sensitive to the context. In the process of text clustering, only useful information is extracted. Information-based clustering analyzes the phrases, text segments surrounding them and latent semantic information. Specially, this method can be used to handle the text with no key words or key phrases, and overcome the limitations of information-based clustering method, such as ambivalent word, thesaurus, phrase and so on.

In order to identify the different applicability of different text clustering methods, we study several text clustering methods and compare the validity of text clustering by experiments. Through the comparative study we address the clustering quality achieved with the text clustering methods of FIHC and k-means.

2 Text Clustering Methods and Evaluation

At present, the research on text clustering is intensive in the world. Due to space limitation, we will restrict our review of text clustering method to popular ones.

2.1 Text Clustering Methods

Popular text clustering methods mainly include hierarchical clustering, partitioning clustering, neural network clustering, Learning Vector Quantization (LVQ) clustering etc.

Hierarchical Clustering is based on a hierarchical decomposition of object set. In the formation of the layers, the splitting (top-down) and the merging (bottom-up) operations can be conducted. For making up the restrictions of merging or splitting, the

clustering performance can be improved by analyzing the object links of each layer or by integrating the other clustering techniques. The hierarchical clustering is to build a spanning tree, among them including the cluster information and similarities of inner a cluster and between the clusters [9]. It generates the nested clusters, and the accuracy is high. But, every time of merging, the similarities between the clusters should be compared globally before choosing the best two, as a result, the speed is slow and the method is not suitable for applications with high volume of textual data [7][8].

Partitioning Clustering is different from the hierarchical clustering. The partitioning clustering is to divide the document set into several clusters horizontally. Initially, given k divisions of the document set, then it improves the clustering accuracy by moving the objects from one cluster to another cluster gradually. The partitioning clustering is high speed, but it needs preset the parameter of the number of the clusters. The clustering result is greatly affected by the selection of the parameter.

K-means method is a typical model of partitioning clustering. It (also called hard c-means) was firstly presented by MacQMen. Suppose X is a set of texts, which consists of n objects. It is required to form k clusters. K-means method divides the n objects into k clusters ($k < n$) based on a division function, so that the objects in each cluster are similar, while those in different clusters are dissimilar.

The main idea of k-means can be described as follows.

Originally, select the k objects randomly, each object represents the average or centre of a cluster. The rest objects are clustered into different categories according to their positions, namely each of them is partitioned into the closest cluster. Then, the centre of each cluster is recalculated iteratively until the division function converges.

Usually the division function in k-means is defined as follows:

$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2 \quad (1)$$

In Eq.1, E is the sum of the square error of all objects, m_i is the centre of the cluster C_i ; p denotes a point in the space. Both p and m_i are multidimensional. It is proposed that the clusters that are formed in terms of the division function E are as independent and compact as possible. K-means method divides the n objects into the k clusters in order to make the objects in the same clusters more similar but the objects in different clusters more dissimilar. The similarity can be calculated by Euclidean distance, cosine function and so on.

K-means method is a typical algorithm for the text clustering. The main advantages are quick and efficient. But the outputs would vary from the parameters, i.e., the results of the text clustering are related to the choosing of parameters. In addition, k-means tends to get into a local minimum. These shortcomings greatly restrict its applications.

SOM is one of the most important models in the feedback neural networks. It was developed by simulating the activities of a neuron firstly by Teuvo Kohonen in 1982 [9]. In recent years, this method has been applied in many areas, such as classification, speech identification, automatic control, combination optimization, data mining etc. SOM depends on weight convergence and topology orderliness. However, it is difficult to analyze the convergence and orderliness in SOM. The existing research mostly concentrates on one dimension cases. SOM is an unsupervised learning algorithm, and obtains the clustering by learning iteratively. Competition is carried on

among the neurons; the winner is the one whose weight value is the most close to the current object. In order to approximate the expectation value, the weight values of the winner and nearest neighbors need to be adjusted. SOM simulates the processing of the human brain; it is very useful for visualization of two or three dimension spaces.

The basic structure of SOM is made up of the input layer, the competition layer and output layer. Output layer is also called mapping layer, the number of the input nodes is the same as the number of dimensions of the feature vector of the text. SOM is a fully connected network.

SOM has been widely used in the clustering analysis, the image processing, speech identification and marketing trend predict. However, the traditional SOM network has many shortcomings. In case of a small quantity of modes, the initial weight values of the network heavily affect the convergence, and the clustering results depend on the order of the presentation of the modes.

SVM derived from statistics theory is a general text clustering method developed in recent years. To certain degree, SVM is similar to neural network. At present, it has been developed continuously [10]. This method can avoid the local optimal problem, and overcome the dimension disaster skillfully. It is overwhelming in solving the small quantity samples or nonlinear and high dimensional problems. By learning SVM can find those support vectors that have the better distinction ability to the different clusters automatically, construct the partition with the maximum dissimilarity, and obtain the good accuracy and flexibility. These advantages make SVM a hotspot in the machine learning area. SVM has been applied in mode identification (include the character identification, text classification, face detection etc.) and in nonlinear system control. It lacks the integration ability of the apriori knowledge and can't support the incremental learning.

The traditional text clustering methods mentioned above can't handle high dimensional, high volume data properly, can not support result visualization, meaningful clustering labels with text diversity.

FIHC is different from the classical text clustering methods that similarity between documents plays a central role in the construction of a cluster. FIHC is "cluster-centered", and it measures the cohesiveness of a cluster directly using frequent itemsets, so that documents in the same cluster share more common items than those in different clusters.

The novelty of FIHC is that it exploits frequent itemsets for defining a cluster, drastically reduces the dimensionality of the document set, and organizes the topic tree to provide a sensible structure for browsing documents. The feature of FIHC is higher clustering accuracy. The number of clusters is an optional parameter. It is easy to browse with meaningful cluster description.

2.2 Evaluation Method

There is no unified evaluation standard for the validity of text clustering. Most existing evaluation methods for text clustering use the evaluation methods for information retrieval or text categorization. The problems in the evaluation of the text clustering validity are:

1. Text clustering is an unsupervised learning process. At the beginning of clustering, there is no apriori knowledge. Therefore, the corresponding relations between automatic and manual clustering do not exist.
2. Clusters determined by the manual depend on the subjectivity. So the comparison can not be made among the different clustering methods. Although some researchers use the same text materials, their clustering standards are not the same.
3. As most implementations of text clustering methods are targeting a specific text resource, it is difficult to apply the same evaluation standard of the text clustering to others.
4. The difference of the implementation methods results in the difference of the representations of the outputs. For example, a fuzzy clustering output is the value of the membership function of the text to different clusters; while a partitioning clustering output is the probability belonging to a certain cluster.

Recall rate and precision, which are performance parameters for evaluating traditional information retrieval, are also useful for the evaluation of text clustering algorithms [7]. Recall rate is the ratio of the texts in a correct cluster to all texts of the same cluster. Precision is the ratio of the texts in a correct cluster to all texts that are divided into the cluster.

Recall rate and precision have the trade-off phenomenon. The transformation between recall rate and precision can be obtained to certain degree in terms of adjusting parameters and thresholds. For example, given a fixed value for recall rate, the text clustering can meet the requirement in terms of adjusting the parameters but at the cost of the precision, and vice versa. To evaluate the validity of text clustering only by a single parameter may have potential misunderstanding. For example, if a text clustering method divides all texts into a single category, the recall rate should be 100%, while the precision would be very low. On the contrary, even if it fails to divide all texts into a category where they should have been clustered, the precision could be very high; while the recall rate could be unacceptable. Therefore, an evaluation method based on recall rate and precision is proposed.

F-measure proposed by C. J. Rijsbergen [11] is defined as the following:

$$F = \frac{2}{\frac{1}{r} + \frac{1}{p}} = \frac{2rp}{r+p} \quad (2)$$

Where r denotes the text recall rate; p denotes the precision. F-measure evaluates the accuracy of text clustering by balancing r and p .

Generally speaking, when evaluating the validity of text clustering, it is necessary to consider as many aspects as possible.

3 Experiment

By implementing FIHC and k-means methods, the text clustering experiments were conducted using 265 English documents. The results of the experiment are compared. For the convenience of comparison, all the documents have been clustered manually,

and they are divided into following three clusters: medication (80 documents), education (100 documents) and sports (85 documents).

The comparison of the accuracy of the two text clustering methods is shown in Table 1 and Fig. 1.

Table 1. Comparison of the accuracy of the two text clustering methods

	FIHC		k-means Method	
	Sum of the clustered documents	Sum of the document clustered correctly	Sum of the clustered documents	Sum of the document clustered correctly
Medication	72	62	96	32
Education	46	37	81	34
Sports	90	77	85	32

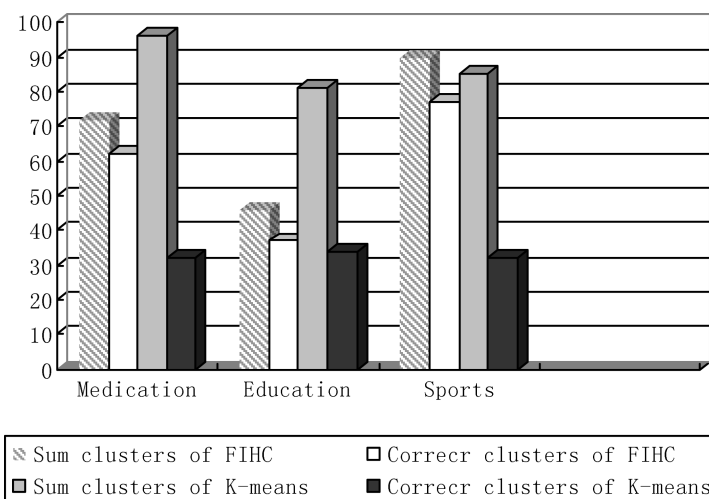


Fig. 1. Comparison of the accuracy of the two text clustering methods

The comparison of the validity of the two text clustering method is shown in Table 2 and Fig.2.

Table 2. Comparison of the validity of the two text clustering methods

Clustering algorithm	FIHC	k-means Method
Recall Rate (Average)	68.4%	37.2%
Precision (Average)	84.0%	37.6%
F-measure	0.754	0.374

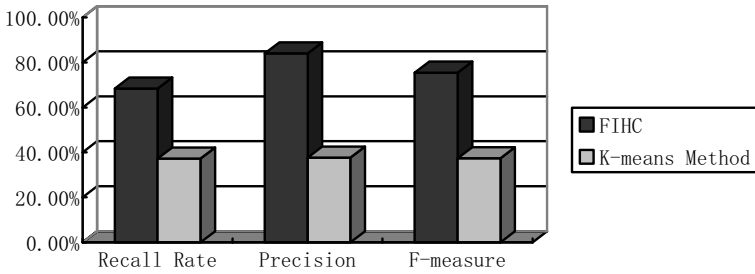


Fig. 2. Comparison of the validity of the two text clustering methods

From the experiments, we can see, FIHC is more accurate because it uses larger search space and it is able to generate the nested clusters. However, its speed is slower because it needs to compare the similarity of all the clusters before choosing the closest two clusters among them during the clustering process. K-means method obtains the clustering validity to a certain extent. It takes shorter time to converge. However, an initial partition is needed and will have influence on the clustering result. If the parameters are not chosen properly, it perhaps can't realize the text clustering.

4 Discussion

In this paper, we analyze the applicability of the different text clustering methods and compare the validity of text clustering of FIHC and k-means methods using three different evaluation standards.

From the above experiments, we conclude as follows:

1. The number and the distribution of the texts would affect the clustering results.
2. K-means method depends on the selection of the initial parameter, and it is sensitive to the unevenly distributed text set.
3. With the increasing of the amount of the texts, the clustering speed of FIHC method could reduce dramatically.

The comparative study reported in this paper will shed lights to improve the validity of text clustering, such as by fusing hierarchy clustering and partition clustering methods. The classical text clustering methods can not satisfy some special requirements for modern scalable applications such as high dimensionality, high volume, sparse distribution etc.

The future direction in text clustering is to realize the streaming text clustering on XML streams, on SMS message, or on real-time broadcasting news etc. Another important research area is to explore efficient evaluation methods of the validity of text clustering that calculate more factors not penalize specializations and generalizations.

References

1. Sasaki, M., Shinnou, H.: Spam Detection Using Text Clustering, International Conference on Cyberworlds. (2005) 316–319
2. Min Kang, Asakimori, K., Utsuki, A., Kaburagi, M.: Automated text clustering system on responses to open-ended questions in course evaluations. 6th International Conference on Information Technology Based Higher Education and Training. (2005) F4B/18 –F4B/22
3. Benjamin CM, Fung, Ke Wang and Martin Ester: Hierarchical Document Clustering Using Frequent Itemsets. In Proceedings of the 2003 SIAM International Conference on Data Mining (SDM'03). San Francisco, CA. (2003) 59–70
4. Rocchio J J.: Document retrieval systems, Optimization and evaluation. Harvard University, Cambridge, MA (1966)
5. Taeho Jo, Japkowicz, N.: Text clustering with NTSO (neural text self organizer), Proceedings of 2005 IEEE International Joint Conference on Neural Networks, Vol. 1. (2005) 558–563
6. Luying Liu, Jianchu Kang, Jing Yu, Zhongliang Wang: A comparative study on unsupervised feature selection methods for text clustering. Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (2005) 597–601
7. Shi Zhongzhi: Knowledge Discovery. Beijing: Tsinghua University Press. (2002)
8. Jian-Suo Xu, Li Wang, TCBLHT: a new method of hierarchical text clustering. Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Vol.4. (2005) 2178–2181
9. V.V Tolat: An analysis of Kohonen's self-organizing maps using a system of energy functions. *Biol.Cybern.* (1990) 64:155-164
10. Fuliang Yin, Jun Wang, Chengan Guo: A Novel Approach to Clustering Analysis Based on Support Vector Machine Advances in Neural Networks: International Symposium on Neural Networks, Dalian, China. (2004), Proceedings, Part I, 565–570
11. Iwayama Makoto, Tokunaga Takenobu: Hierarchical Bayesian clustering for automatic text classification. Department of Computer Science Tokyo Institute of Technology, TechRep, TR95-0015, (1995)
12. Rigouste, L.; Cappe, O.; Yvon, F.; Inference for probabilistic unsupervised text clustering, Processing of 2005 IEEE/SP 13th Workshop on Statistical Signal. (2005) 387–392
13. McNeil, A.R., Sarkodie-Gyan, T.: A neural network based recognition scheme for the classification of industrial components, Proceedings of 1995 IEEE International Conference on Fuzzy Systems. Vol. 4. (1995) 1813–1818
14. Cao Sulì, Zeng Fuhua, Cao Huanguang: Automatic Chinese Text Classification System Based on the Frequency Vector of the Chinese Word. *Journal of Shanxi University (Natural Science Edition)*. Vol. 22 (2). (1999) 44-49
15. R.C. Dubes, A. K. Jain: Algorithms for Clustering Data. Prentice Hall College Div, Englewood Cliffs, NJ. (1998)

Concept Based Text Classification Using Labeled and Unlabeled Data

Ping Gu, Qingsheng Zhu, and Xiping He

Dept. of Computer Science, Chongqing University
400044 Chongqing, China
guping2k@cqu.edu.cn

Abstract. Recent work has shown improvements in text clustering and classification by integrating conceptual features extracted from background knowledge. In this paper we address the problem of text classification with labeled data and unlabeled data. We propose a Latent Bayes Ensemble model based on word-concept mapping and transductive boosting method. With the knowledge extracted from ontologies, we hope to improve the classification accuracy even with large amounts of unlabeled documents. We conducted several experiments on two well-known corpora and the results are compared with Naïve Bayes and TSVM classifiers.

1 Introduction

Text classification is usually based on supervised learning, with feature vectors as representatives of text documents. However, most methods are typically built with the bag of terms model. In this representation, documents are often represented through binary variables or absolute frequencies. Learning algorithms are thus restricted to detecting patterns in the terminology only and conceptual patterns are ignored. This may lead to some deficiencies, for example, polysemous word “bank” is often treated as one single feature while they actually have multiple distinct meanings(financial institution, sloping land). A number of attempts [1][2][3] have been made to incorporate semantic knowledge into the vector space representation. So far, however, most methods only work with labeled training set, when the distribution of training set is different from test set, this may result in some generalization error.

In this paper, we consider using word-concept mapping and transductive boosting in improving text classification with partially labeled data. Our approach is based on a probabilistic model, where the concepts are extracted from the Wordnet[4] and used as latent variables. The learning process involved in estimating the probabilities for word-concept and concept-topic pairs with small training set, which is then executed on the test set with transductive boosting to obtain improved predictions. Hence, our approach offers an attractive venue for using unlabeled data to improve supervised learning. For confirming the validity of our approach, experiments are conducted on two well-known text corpora, the results are compared with Naïve Bayes and TSVM models.

2 Preliminaries

In text classification, typical document representation is based on bags of terms, this representations disregard conceptual similarity of terms, as a sequence, are not sufficiently robust with respect to the variations in term usage. For enriching the term representation with concepts, we build on the background knowledge source like lexicons and ontology [5].

Definition 2.1. A core ontology is a tuple $O := (C, \leq_c)$, consisting of a set C whose elements are called concept identifiers, and a partial order \leq_c on C , called concept hierarchy or taxonomy.

Definition 2.2. A lexicon for an ontology O is a tuple $Lex := (S_c, ref_c)$ consisting of a set S_c , whose elements are called signs for concepts, and a relation $ref_c \subseteq S_c \times C$ called lexical reference for concepts. Based on ref_c , for $s \in S_c$, we define $ref_c(s) = \{c \in C \mid (s, c) \in ref_c\}$.

For the scope of this paper, we have chosen Wordnet as the background knowledge, as it fits to the generality of most corpus. Wordnet comprise a core ontology and a lexicon, it organizes words and word expressions of different syntactic categories into synsets (concept), each of which represents an underlying concept and links these through semantic relations. It comprises a total of 109377 concepts and 144684 lexical entries. For a term t appearing in document d , $ref_c(t)$ allows for retrieving its corresponding concepts.

3 Latent Bayes Model Based on Word to Concept

In this section we introduce the Latent Bayes model based on supervised learning, and use a text-context similarity comparison for an initial mapping of words to concepts.

3.1 Basic Latent Bayes Model

Given m labeled training documents $D = \{d, \dots, d_m\}$ with topic labels $T = \{t_1, \dots, t_m\}$ (+1 for positive, -1 for negative) and n unlabeled test examples d_{m+1}, \dots, d_{m+n} . Let $F = \{f_1, \dots, f_n\}$ be the set of lexical features in the documents and $C = \{c_1, \dots, c_k\}$ be the concepts in ontologies. Our Latent Bayes model for feature topic co-occurrence can be depicted as follow:

1. Select a topic t with probability $P[t]$;
2. Pick a latent variable c with probability $P[c|t]$, the probability that concept c describes the topic t ;
3. Generate a feature f with probability $P[f|c]$.

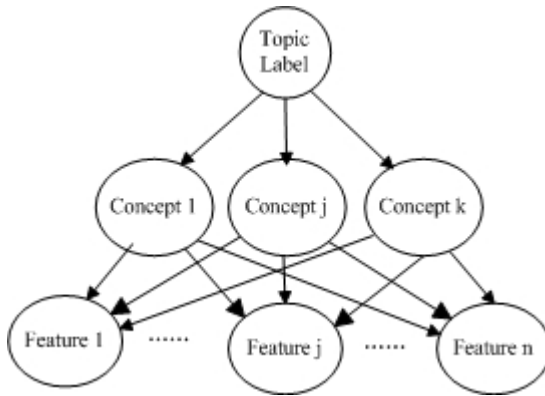


Fig. 1. Graphical model representation for the Latent Bayes Model by means of word-concept

The model is based on two independence assumptions: the influences of concepts to features are assumed to be casually-independent, i.e., $P[f|c_1, c_2]=P[f|c_1]P[f|c_2]$. Also, we assumed that features f are conditionally independent of the topics t , given the latent concept variable c : $P[(f,t)|c]=[f|c] \cdot P[t|c]$. To describe the generative process of an example (f, t) , we sum up over all the possible values that the latent variables might take

$$P[f, t] = \sum_c P[c]P[(f, t) | c] . \tag{1}$$

The likelihood of observing pairs (f, t) can be expressed as:

$$L = \prod_{f,t} P[f, t]^{n(f,t)} . \tag{2}$$

Where $n(f, t)$ is the number of occurrences of feature f in the training set of topic t . The learning problem can be expressed as a maximization of the observed data log-likelihood:

$$l = \sum_{f,t} n(f,t) \log(P[f, t]) = \sum_{f,t} n(f,t) \log(\sum_c P[c]P[(f, t) | c]) . \tag{3}$$

Direct maximization of the log-likelihood by partial derivatives is difficult. We can employ Expectation Maximization (EM) algorithm for model fitting. In the E-Step, we compute the posterior probabilities for the latent variables, taking as evidence the observed data (current estimates of the model parameters).

$$p[c | (f, t)] = \frac{P[f | c]P[c | t]}{\sum_c P[f | c]P[c | t]} . \tag{4}$$

In the M-Step, we can update the current parameters based on the expected complete data log-likelihood:

$$P[f | c] = \frac{\sum_t n(f, t) P[c | (f, t)]}{\sum_f \sum_t n(f, t) P[c | (f, t)]} \tag{5}$$

$$P[c | t] = \frac{\sum_f n(f, t) P[c | (f, t)]}{\sum_c \sum_f n(f, t) P[c | (f, t)]} \tag{6}$$

$$P[t] = \frac{\sum_{f,c} n(f, t) P[c | (f, t)]}{\sum_t \sum_{f,c} n(f, t) P[c | (f, t)]} \tag{7}$$

According to this model, words related to similar topics tends to be generated by the same concept, whereas a word with multiple meanings tends to receive a high generating probability from a few corresponding concepts. Thus, we can explain feature-topic associations by means of latent concepts.

Once the marginal distribution describing this latent model has been estimated, we can use Bayes rules to predict which topic is classified for each document.

$$P[t | d] = \frac{P[d | t] P[t]}{P[d]} = \frac{P[d | t] P[t]}{\sum_t P[d | t] P[t]} \tag{8}$$

Where

$$P[d | t] = \prod_{f \in d} P[f | t] = \prod_{f \in d} \sum_{c \in C} P[f | c] P[c | t] \tag{9}$$

3.2 Disambiguation as a Pre-initialization

EM algorithm tends to stop in a local maximum of the likelihood function, especially when assignment of words to concepts is ambiguous. Our pre-initialization proposal can help mapping features to concepts and concepts to topics by disambiguation [6].

Let w be a word that we want to map to the ontological senses. By querying the Wordnet for the possible meanings of word w , we first get synsets for each of the word meanings. Then, word sense disambiguation is applied by comparing the similarity among local context around w and each of its possible meanings.

The context for word w is a window around its offset in the document, the context for the concept is taken from the ontology: synonyms, hypernyms, hyponyms, holonyms, siblings and short textual descriptions. For each of the candidate senses c_i , we compare the cosine similarity measure between the tfidf vectors of $\text{context}(w)$ and $\text{context}(c_i)$.

$$\cos(\text{context}(w), \text{context}(c_i)) = \frac{\text{context}(w) \cdot \text{context}(c_i)}{\|\text{context}(w)\| \cdot \|\text{context}(c_i)\|} \tag{10}$$

In a similar way, we relate concepts to topics based on similarity of bags-of-terms. The context for a topic t is defined to be the bag of features selected from the training set by decreasing Mutual Information values. For our implementation, the window size was set to 8 and the top 100 terms were used with regards to MI rank.

Once all the similarity measures for feature-concept and concept-topic pairs have been computed, we normalize and interpret them as estimates of the probabilities $P[f | c]$ and $P[c | t]$. The computed values are then used for initializing EM in the model fitting process.

4 Ensemble by Transductive Boosting

Learning the basic Latent Baye Model needs large amount of labeled training data. However, in many applications, we faced with only small training set. It is crucial that the classifier be able to generalize well using little training data. In this section, we consider using inexpensive test data to augment the basic Latent Bayes model by combining AdaBoost[7] and Transduction[8] method.

Transduction is a general learning framework that minimizes classification errors for the test data. Joachims[9] adapted it to TSVM and obtained an improvement in classification performance. Nevertheless, the possibility of a decrease in performance remains if the ratio of positive to negative examples is different in the training and test set. As a solution, we propose a novel transductive boosting algorithm to enhance the model’s generalization capability even when the distribution is different in the training and test set.

Given m labeled training data and n test data with initial labels $t_{m+1}^*, \dots, t_{m+n}^*$ equals to 0, we define m^+ be the number of training data with positive class; $n_{labeled}$ be the number of test data with a labeled class; $n_{labeled}^+$ be the number of test data with a labeled positive class. The major steps of transductive boosting algorithm are summarized as follows:

- (1) Initialize the weights for the training data $D_1(i) = 1/m$ and the test data $D_1(j) = 0$;
- (2) For $t = 1, \dots, T$ repeat (3) – (6)
- (3) Train a Latent Bayes model $h_t(f)$ with the labeled data under weight D_t ;
- (4) Calculate the parameter $\frac{1}{2} \ln(1 - e_t) / e_t$ where $e_t = \sum_{i: h_t(f_i) \neq y_i} D_t(i)$;
- (5) Update the weight of training data based on $D_{t+1}(i) = D_t(i) \exp(-\alpha_t y_i h_t(f_i)) / Z_t$;
- (6) If $n_{labeled} = 0$ or $m^+ / m \geq n_{labeled}^+ / n_{labeled}$ then

$$\text{select test sample } j \text{ which maximize } H(f_j) = \sum_{k=1}^t \alpha_k h_k(f_j);$$

$$y_j = +1; D_{t+1}(j) = \beta; (\beta \text{ is a small value})$$

update the weight of the already labeled data as $D_{t+1}(i) = D_t(i)/Z_t'$,

End if

If $n_{labeled} \neq 0$ and $m^+ / m < n_{labeled}^+ / n_{labeled}$ then

select test sample j which minimize $H(f_j) = \sum_{k=1}^t \alpha_k h_k(f_j)$

$y_j = -1$; $D_{t+1}(j) = \beta$;

update the weight of the labeled data as $D_{t+1}(i) = D_t(i)/Z_t'$;

End if

(7) Return the final hypothesis as $H(f_j) = \sum_{k=1}^t \alpha_k h_k(f_j)$;

In the above algorithm, Z_t is the normalized factor for $\sum_{i=1}^m D_{t+1}$ equals 1, and Z_t' is also the normalizing factor such that the sum of the data without j equals $1 - \beta$. In step (1), we initialize the weights of the training and test data. Step (3)-(6) is to perform labeling on both the training and test data. By assuming that the ratio of positive and negative examples is the same as which in the training data, in every round, we label the class only for the most reliable test data and give a small value β to the weight of selected test data, because the reliability of the labels is lower than that of the training data.

By selecting data to lower the probability of wrongly labeling t_j^* and maximize $t_j^* H(f_i)$ at each round, we can produce Latent Bayes Ensemble that minimize the cost function over the training and test set:

$$Cost(H(f)) = \frac{1}{m+n} \left(\sum_{i=1}^m \exp(-t_i H(f_i)) + \sum_{j=m+1}^{m+n} \exp(-t_j^* H(f_j)) \right) \quad (11)$$

5 Experimental Results

To evaluate the Latent Bayes Ensemble(LBE) model, we conducted experiments on two well-known datasets and compare it to the Naïve Bayes and TSVM models.

5.1 Experimental Settings

We adopted two data collections(Reuters-21578 and 20-Newsgroups) in the experiments. The Reuters-21578 corpus is a standard text classification benchmark. In our implements, ModApte split is used to divide the corpus into 9,603 training documents and 3299 test documents. We parse and only select the documents belonging

to five topics(earn, acq, crude, trade, money-fx), this split the collection into approximately 5000 files for training and 2000 files for testing. The 20-Newsgroups collection consists of documents from 20 different newsgroups. The latest 4000 documents are used for training, and a random selected 10000 documents are used for test set. Feature selection was done by empirical mutual information with set size equals 100. For both collections, a total of 2410-6551 concepts were extracted from Wordnet according to different depth in the ontologies. To evaluate the performance of text classification, we choose Micro $F1$ [10] as the performance criteria. Each classifier was trained and tested for eight times and their evaluation results were averaged. For transductive boosting, the iteration T was fixed to 1,000 in all of the experiments and $\beta = 0.01$.

5.2 Results

We first carried the experiment to compare the Naïve Bayes model, single Latent Bayes(SLB) model and the LBE model when different number of unlabelled test data are incorporated into the training set. Table 1 shows the average results on Reuters-21578 corpus. The labeled training set includes 300 positive and 300 negative examples.

Table 1. Micro $F1$ for different training instances in Reuters-21578 corpus

Unlabelled Examples	Naïve Bayes	SLB	LBE
0	62.5%	63.1%	63.5%
50	64.1%	64.8%	68.1%
100	63.8%	64.6%	70.6%
400	70.5%	71.7%	78.3%
800	71.8%	73.5%	82.1%
1500	71.9%	74.0%	86.2%

From the results, we can see that by incorporate semantic knowledge, SLB is superior to Naïve Bayes model, but the improvement is slight. By analyzing the nature of Reuters-21578, we found that its vocabulary is small and uniform, each topic can be described with standard vector terms, so the performance improved by word semantics is not obvious. While LBE model is significantly superior to Naïve Bayes and SLB model for the adoption of boosting method, especially when the amount of unlabelled examples increase largely.

We conducted similar experiment on another dataset: 20-Newsgroups collection and select 500 positive and 500 negative examples as the training set. Table 2 shows the results on 20-Newsgroups collection. Except observing similar improvement with LBE as above, we can also see a significant improvement by exploiting semantics in documents. It is because there is a richer vocabulary, semantics in which can have more important compact.

Table 2. Micro *F1* for different training instances in 20-Newsgroups corpus

Unlabelled Examples	Naïve Bayes	SLB	LBE
0	52.5%	56.6%	56.9%
100	57.8%	63.9%	69.1%
200	64.1%	69.2%	74.2%
500	63.7%	69.0%	77.5%
1000	66.4%	74.9%	82.0%
2000	65.9%	75.1%	83.1%

The third experiment compare the performance of Naive Bayes, TSVM and LBE model when the ratio of positive to negative in the training set(1:2) is different from test set (1:9). The results with 20-Newsgroups collection are shown in Fig. 2

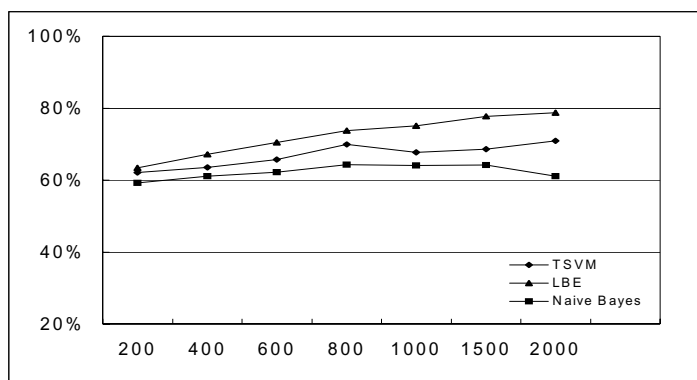


Fig. 2. Micro *F1* when the ratio of positive to negative examples in the training set is 1:2 and test set is 1:9

The performance of LBE is only slightly decreased compared with that of SVM when the distributions in the training and test set are significantly distinct. This confirm the effectiveness of the Latent Bayes Ensemble model with the labeled and unlabeled data even their distribution is different.

6 Conclusion

In this paper, we proposed a Latent Bayes Ensemble model based on word-concept to automatic text classification. The approach proposed seems to be beneficial for collections with a rich natural language vocabulary, setups in which classical terms-only methods risk to be trapped in the semantic variations. Furthermore, we apply a transductive boosting method for ensemble creation in order to make efficient use of the

large amount of unlabeled data. The experimental results indicate that our approach is effective for both the labeled and unlabeled set, especially when we do not know the ratio of positive to negative examples in the test data.

References

1. Hotho, A.; Staab, S. and Stumme, G.: Ontologies Improve Text Document Clustering. In Proceedings of ICDM, 2003 . IEEE Computer Society. (2003)
2. A.Maedche and S.Saab. Ontology Learning for the Semantic Web. IEEE Intelligent Systems, 16(2),(2001)
3. Bloehdorn, S. and Hotho, A.:Text Classification by Boosting Weak Learners based on Terms and Concepts. In Proceedings of ICDM . IEEE Computer Society. (2004)
4. G. Miller. WordNet: A lexical database for english. CACM, 38(11):39–41, (1995).
5. E. Bozsak et al. Kaon: Towards a large scale semantic web. In Proceedings of EC-Web, Aix-en-Provence, France, LNCS 2455 Springer. (2002) 304–313.
6. Bhattacharya, I., & Getoor, L. & Bengio, Y.. Unsupervised Sense Disambiguation Using Bilingual Probabilistic Models. Meeting of the Association for Computational Linguistics. (2004)
7. Y. Freund and R. E. Schapire.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), (1997).
8. Vapnik, V. N.: Statistical Learning Theory. Wiley. (1998).
9. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Proc. of 10th European Conference on Machine Learning (ECML-98), (1998) 137-142.
10. Manning,C.D., Schutze, H.. Foundations of Statistical Natural Language Processing. Cambridge: MIT Press. (2000)

Learning Semantic User Profiles from Text

M. Degemmis, P. Lops, and G. Semeraro

Dipartimento di Informatica - Università di Bari
Via E. Orabona, 4 - 70125 Bari - Italia
{degemmis, lops, semeraro}@di.uniba.it

Abstract. This paper focuses on the problem of choosing a representation of documents that can be suitable to induce more advanced *semantic* user profiles, in which *concepts* are used instead of *keywords* to represent user interests. We propose a method which integrates a word sense disambiguation algorithm based on the WordNet IS-A hierarchy, with two machine learning techniques to induce *semantic user profiles*, namely a relevance feedback method and a probabilistic one. The document representation proposed, that we called *Bag-Of-Synsets* improves the classic *Bag-Of-Words* approach, as shown by an extensive experimental session.

Keywords: User Profiles, Text Categorization, Word Sense Disambiguation, WordNet.

1 Introduction

Machine learning techniques can be used to induce from text documents a keyword-based structured model of the interests of a user, the *user profile*, as shown in our previous work [4]. There are information access scenarios that cannot be solved through straightforward matching of queries and documents represented by keywords. For example, a user interested in retrieving “interesting movies” cannot easily express this form of information need as a query suitable for search engines. In these problematic information scenarios, a possible solution could be to develop methods able to analyze documents the user has already deemed as interesting in order to discover relevant concepts to be stored in his personal profile. Keyword-based approaches are unable to capture the *semantics* of the user interests. They are primarily driven by a string-matching operation that suffers from problems of: POLYSEMY, the presence of multiple meanings for one word; SYNONYMY, multiple words have the same meaning. More accurate profiles that capture concepts expressing users’ interests from relevant documents are needed. These *semantic* profiles will contain references to concepts defined in lexicons or, in a further step, ontologies. After discussing the main works related to our research, we describe in Section 3 a strategy to represent documents by using word sense disambiguation based on WordNet [10]. Sections 4 and 5 describe how this representation can be exploited by both a Rocchio relevance feedback [14] and Naïve Bayes [11] to learn semantic user profiles. Two different prototypes have been developed in order to implement the proposed approaches: *RocchioProfiler* and *ITem Recommender (ITR)*, whose effectiveness is evaluated in Section 6. Conclusions are drawn in Section 7.

2 Related Work

Our work was mainly inspired by several works. *Syskill & Webert* [13] learns user profiles as Bayesian classifiers. *ifWeb* [1] supports users in document searching by maintaining user profiles which store both interests and explicit *dis*interests. *SiteIF* [7] exploits a sense-based representation to build a user profile as a semantic network whose nodes represent senses of the words in documents requested by the user. *Fab* [2] adopts a Rocchio [14] relevance feedback method to create and update the user personal model (selection agent) that are directly compared to determine similar users for collaborative recommendations. According to these successful works, we conceived our content-based systems as text classifiers able 1) to deal with a sense-based document representation and 2) to distinguish between interests and *dis*interests of users. The strategy we propose to shift from a keyword-based document representation to a sense-based document representation is *to integrate lexical knowledge in the indexing step of training documents*. Several methods have been proposed to accomplish this task. Scott and Matwin proposed to include WordNet information at the feature level by expanding each word in the training set with *all* the synonyms for it in WordNet, in order to avoid a WSD process [15]. This approach has shown a decrease of effectiveness in the obtained classifier, mostly due to the word ambiguity problem. Some kind of disambiguation is required, as suggested by subsequent works. Bloedhorn and Hotho compared three strategies to map words to senses: No WSD, most frequent sense as provided by WordNet, WSD based on context [3]. They found positive results on the Reuters 25178, the OSHUMED and the FAODOC corpus.

3 Semantic Indexing by Using WSD

We extend the classic *Bag-Of-Words* (BOW) model [16] to a model in which the meanings (senses) corresponding to the words in the documents are considered as features. A procedure is needed to assign senses to words. The goal of a WSD algorithm is to associate the appropriate meaning or sense s to a word w in document d , by exploiting its *context* C , that is a set of words surrounding w [8]. The sense s is selected from a predefined set of possibilities, the *sense inventory*. In the proposed algorithm, it is obtained from WordNet version 1.7.1 (<http://wordnet.princeton.edu>). The basic building block for WordNet is the SYNSET (SYNONYM SET), a set of words with synonymous meanings which represents a specific sense of a word. We addressed the WSD problem by proposing an algorithm based on semantic similarity between synsets computed by exploiting the hyponymy/hypernymy (IS-A) relation in WordNet. The text in d is processed by (1) tokenization, part-of-speech tagging (POS) and lemmatization; (2) synset identification with WSD. Then, synset identification phase is performed by WSD. The core idea behind the proposed WSD algorithm is to disambiguate w by determining the degree of *semantic similarity* among candidate synsets for w and those of each word in C . Thus, the proper synset assigned to w is that with the highest similarity with respect to its context of use. The

measure of semantic similarity adopted in this work is the Leacock-Chodorow measure [6]. Similarity between synsets a and b is measured by the number of nodes in the shortest path from a to b (function `SinSim`, lines 24-28). Let w be the word to be disambiguated. The procedure starts by defining the context C of w as the set of words in the same slot of w having the same POS as w . Next, the algorithm identifies both the sense inventory X for w , and the sense inventory X_j for each word w_j in C . The sense inventory T for the whole context C is given by the union of all X_j . After this step, we measure the similarity of each candidate sense $s_i \in X$ to that of each sense $s_h \in T$ and then the sense assigned to w is the one with the highest similarity score. The WSD procedure is

Algorithm 1. The WordNet-based WSD algorithm

```

1: procedure WSD( $w, d$ )      ▷ finds the proper synset of a polysemous word  $w$  in
   document  $d$ 
2:    $C \leftarrow \{w_1, \dots, w_n\}$       ▷  $C$  is the context of  $w$ . For example,
    $C = \{w_1, w_2, w_3, w_4\}$  is a window with radius=2, if the sequence of words
    $\{w_1, w_2, w, w_3, w_4\}$  appears in  $d$ 
3:    $X \leftarrow \{s_1, \dots, s_k\}$     ▷  $X$  is sense inventory for  $w$ , that is the set of all candidate
   synsets for  $w$  returned by WordNet
4:    $s \leftarrow \text{null}$                 ▷  $s$  is the synset to be returned
5:    $\text{score} \leftarrow 0$               ▷  $\text{score}$  is the similarity score assigned to  $s$  wrt to the context  $C$ 
6:    $T \leftarrow \emptyset$             ▷  $T$  is the set of all candidate synsets for all words in  $C$ 
7:   for all  $w_j \in C$  do
8:     if  $\text{POS}(w_j) = \text{POS}(w)$  then      ▷  $\text{POS}(y)$  is the part-of-speech of  $y$ 
9:        $X_j \leftarrow \{s_{j1}, \dots, s_{jm}\}$   ▷  $X_j$  is the set of  $m$  possible senses for  $w_j$ 
10:       $T \leftarrow T \cup X_j$ 
11:     end if
12:   end for
13:   for all  $s_i \in X$  do
14:     for all  $s_h \in T$  do
15:        $\text{score}_{ih} \leftarrow \text{SINSIM}(s_i, s_h)$   ▷ computing similarity scores between  $s_i$ 
       and every synset  $s_h \in T$ 
16:       if  $\text{score}_{ih} \geq \text{score}$  then
17:          $\text{score} \leftarrow \text{score}_{ih}$ 
18:          $s \leftarrow s_i$  ▷  $s$  is the synset  $s_i \in X$  having the highest similarity score
           wrt the synsets in  $T$ 
19:       end if
20:     end for
21:   end for
22:   return  $s$ 
23: end procedure
24: function SINSIM( $a, b$ )          ▷ The similarity of the synsets  $a$  and  $b$ 
25:    $N_p \leftarrow$  the number of nodes in path  $p$  from  $a$  to  $b$ 
26:    $D \leftarrow$  maximum depth of the taxonomy          ▷ In WordNet 1.7.1  $D = 16$ 
27:    $r \leftarrow -\log(N_p/2 \cdot D)$ 
28:   return  $r$ 
29: end function

```

fundamental to obtain a synset-based vector space representation that we called Bag-Of-Synsets (BOS). In this model, a synset vector corresponds to a document, instead of a word vector. Each document is represented by a set of *slots*. Each slot is a textual field corresponding to a specific feature of the document. In our application scenario, documents are movie descriptions represented by five slots: *title* (the title of the movie); *cast* (names of the actors); *director* (name of the director); *summary* (a short text that presents the main points of the narration); *keywords* describing the main topics of the movie. The text is represented according to the BOS model by counting separately the occurrences of a synset in the slots in which it appears. Assume that we have a collection of N documents. Let m be the index of the slot, for $n = 1, 2, \dots, N$, the n -th document is reduced to five bags of synsets, one for each slot:

$$d_n^m = \langle t_{n1}^m, t_{n2}^m, \dots, t_{nD_{nm}}^m \rangle$$

where t_{nk}^m is the k -th synset in slot s_m of document d_n and D_{nm} is the total number of synsets appearing in the m -th slot of document d_n . For all n, k and m , $t_{nk}^m \in V_m$, which is the vocabulary for the slot s_m (the set of all different synsets found in slot s_m). Document d_n is finally represented in the vector space by five synset-frequency vectors:

$$f_n^m = \langle w_{n1}^m, w_{n2}^m, \dots, w_{nD_{nm}}^m \rangle$$

where w_{nk}^m is the weight of the synset t_k in the slot s_m of document d_n and can be computed in different ways: It can be simply the number of times synset t_k appears in slot s_m or a more complex TF-IDF score. The BOS document representation is obtained by using the following rules: (1) each monosemous word w in a slot of a document d is mapped into the corresponding WordNet synset; (2) for each pair of words $\langle noun, noun \rangle$ or $\langle adjective, noun \rangle$, a search in WordNet is made to verify if at least one synset exists for the bigram $\langle w_1, w_2 \rangle$. In the positive case, algorithm 1 is applied on the bigram, otherwise it is applied separately on w_1 and w_2 ; in both cases all words in the slot are used as the context C of the word(s) to be disambiguated; (3) each polysemous unigram w is disambiguated, using all words in the slot as the context C of w . The proposed technique should allow to obtain profiles able to recommend documents semantically closer to the user interests. The difference with respect to keyword-based profiles is that synset unique identifiers are used instead of words.

4 A Relevance Feedback Method for Learning WordNet-Based Profiles

We consider the problem of learning user profiles as a binary text categorization task: The set of categories is $C = \{c_+, c_-\}$, where c_+ is the positive class (user-likes) and c_- the negative one (user-dislikes). In the Rocchio algorithm for text categorization, documents are represented with the vector space model and the major heuristic component is the TFIDF word weighting scheme [14]:

$$\text{TFIDF}(t_k, d_j) = \underbrace{\text{TF}(t_k, d_j)}_{\text{TF}} \cdot \underbrace{\log \frac{N}{n_k}}_{\text{IDF}} \tag{1}$$

where N is the total number of documents in the training set and n_k is the number of documents containing the term t_k . $\text{TF}(t_k, d_j)$ computes the frequency of t_k in document d_j . Learning combines vectors of positive and negative examples into a prototype vector \vec{c} for each class in the set of classes C . The method computes a classifier $\vec{c}_i = \langle \omega_{1i}, \dots, \omega_{|T|i} \rangle$ for category c_i (T is the *vocabulary*, that is the set of distinct terms in the training set) by means of the formula:

$$\omega_{ki} = \beta \cdot \sum_{\{d_j \in \text{POS}_i\}} \frac{\omega_{kj}}{|\text{POS}_i|} - \gamma \cdot \sum_{\{d_j \in \text{NEG}_i\}} \frac{\omega_{kj}}{|\text{NEG}_i|} \tag{2}$$

where ω_{kj} is the TFIDF weight of the term t_k in document d_j , POS_i and NEG_i are the set of positive and negative examples in the training set for the specific class c_i , β and γ are control parameters that allow setting the relative importance of *all* positive and negative examples. To assign a class \tilde{c} to a document d_j , the similarity between each prototype vector \vec{c}_i and the document vector \vec{d}_j is computed and \tilde{c} will be the c_i with the highest value of similarity. We propose a modified version of this method *able to manage documents structured in slots and represented by WordNet synsets*. As reported in Section 3, each document d_j is represented in the vector space by five synset-frequency vectors:

$$f_j^m = \langle w_{j1}^m, w_{j2}^m, \dots, w_{jD_{jm}}^m \rangle$$

where D_{jm} is the total number of different synsets appearing in the m -th slot of document d_j and w_{jk}^m is the weight of the synset t_k in the slot s_m of document d_j , computed according to a synset weighting strategy described in [5]. We do not report here the details of the strategy because they do not add anything to the discussion. Given a user u and a set of rated movies in a specific genre G (e.g. *Comedy*), the aim is to learn a profile able to recognize movies liked by the user in that genre. For a specific user u , we maintain separate profiles, one for each subject category he/she provided some feedback (ratings). Learning consists in inducing one prototype vector for *each slot*: These five vectors will represent the user profile. Each prototype vector could contribute in a different way to the calculation of the similarity between the vectors representing a movie and the vectors representing the user profile. More formally, we compute one prototype vector $\vec{p}_i^m = \langle \omega_{1i}^m, \dots, \omega_{|V_m|i}^m \rangle$ for each slot s_m and for each class c_i (c_+ and c_- , user-likes and user-dislikes, respectively, V_m is the vocabulary for slot s_m , that is the set of all distinct synsets appearing in slot s_m) by using the ratings given by the user on movie descriptions in genre G . In other words, the method builds two profiles for user u and genre G : The positive profile (composed by 5 prototypes \vec{p}_+^m , corresponding to the slots) is learned from positive examples, the negative profile (\vec{p}_-^m) is learned from negative examples. Each rating $r_{u,j}$ on the document d_j is a discrete judgment ranging from 1 to 6 used to compute the coordinates of the vectors in both the positive and the negative user profile:

$$\omega_{ki}^m = \sum_{\{d_j \in POS_i\}} \frac{w_{jk}^m \cdot r'_{u,j}}{|POS_i|} \quad (3) \quad \omega_{ki}^m = \sum_{\{d_j \in NEG_i\}} \frac{w_{jk}^m \cdot r'_{u,j}}{|NEG_i|} \quad (4)$$

where $r'_{u,j}$ is the normalized value of $r_{u,j}$ ranging between 0 and 1 (respectively corresponding to $r_{u,j} = 1$ and 6), $POS_i = \{d_j \in T_r | r_{u,j} > 3\}$, $NEG_i = \{d_j \in T_r | r_{u,j} \leq 3\}$, and w_{jk}^m is the weight of the synset t_k in the slot s_m of document d_j , computed as in [5]. Equations (3) and (4) differ from the classical formula in the fact that the parameters β and γ are substituted by the ratings $r'_{u,j}$ that give a different weight to each document in the training set. The similarity between a profile and a movie is obtained by computing five partial similarity values between each pair of corresponding vectors \vec{p}_i^m and \vec{d}_j^m . A weighted average of the five values is computed, assigning a different weight α_m to reflect the importance of a slot in classifying a movie. In our experiments, we used $\alpha_1 = 0.1$ (title), $\alpha_2 = 0.15$ (director), $\alpha_3 = 0.15$ (cast), $\alpha_4 = 0.25$ (summary) and $\alpha_5 = 0.35$ (keywords). The values α_m were defined according to experiments not reported in the paper for the sake of brevity. Each experiment consisted in a run of the Rocchio algorithm by using a different selection of α_m values. Here with the term *run* we intend executing all the experimental sessions on the 10 “Genre” EachMovie datasets described in Table 2 and whose results are reported in Table 3. The α_m values reported here are those that allowed to obtain the best predictive accuracy, corresponding to results in Table 3. Since the user profile is composed by both the positive and the negative profiles, we compute two similarity values, one for each profile. The document d_j is considered as interesting only if the similarity value of the positive profile is higher than the similarity of the negative one.

5 A Naïve Bayes Method for User Profiling

In the Naïve Bayes approach to text categorization, the learned probabilistic model is used to classify a document d_i by selecting the class with the highest probability. As a working model for the naïve Bayes classifier, we use the multinomial event model [9] to estimate the *a posteriori* probability, $P(c_j | d_i)$, of document d_i belonging to class c_j :

$$P(c_j | d_i) = P(c_j) \prod_{w \in V_{d_i}} P(t_k | c_j)^{N(d_i, t_k)} \quad (5)$$

where $N(d_i, t_k)$ is defined as the number of times word or token t_k appeared in document d_i . Notice that rather than getting the product of all distinct words in the corpus, V , we only use the subset of the vocabulary, V_{d_i} , containing the words that appear in the document d_i . Since each instance is encoded as a vector of BOS, one for each slot, Equation (5) becomes:

$$P(c_j | d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m=1}^{|S|} \prod_{k=1}^{|b_{im}|} P(t_k | c_j, s_m)^{n_{kim}} \quad (6)$$

where $S = \{s_1, s_2, \dots, s_{|S|}\}$ is the set of slots, b_{im} is the BOS in the slot s_m of the instance d_i , n_{kim} is the number of occurrences of the synset t_k in b_{im} . ITR implements this approach to classify documents as interesting or uninteresting for a particular user. To compute (6), we only need to estimate $P(c_j)$ and $P(t_k|c_j, s_m)$ in the training phase of the system. The documents used to train the system are the same movie descriptions used for RocchioProfiler. An instance labeled with a rating r , $1 \leq r \leq 3$, belongs to class c_- (user-dislikes); if $4 \leq r \leq 6$ then the instance belongs to class c_+ (user-likes). Each rating was normalized to obtain values ranging between 0 and 1:

$$w_+^i = \frac{r-1}{MAX-1}; \quad w_-^i = 1 - w_+^i \quad (7)$$

where MAX is the maximum rating that can be assigned to an instance. The weights in (7) are used for weighting the occurrences of a word in a document and to estimate the probability terms from the training set TR . The prior probabilities of the classes are computed according to the following equation:

$$\hat{P}(c_j) = \frac{\sum_{i=1}^{|TR|} w_j^i + 1}{|TR| + 2} \quad (8)$$

Witten-Bell smoothing [17] has been adopted to compute $P(t_k|c_j, s_m)$, by taking into account that documents are structured into slots:

$$P(t_k|c_j, s_m) = \begin{cases} \frac{N(t_k, c_j, s_m)}{V_{c_j} + \sum_i N(t_i, c_j, s_m)} & \text{if } N(t_k, c_j, s_m) \neq 0 \\ \frac{1}{V_{c_j} + \sum_i N(t_i, c_j, s_m)} \frac{1}{V - V_{c_j}} & \text{if } N(t_k, c_j, s_m) = 0 \end{cases} \quad (9)$$

where $N(t_k, c_j, s_m)$ is the count of the weighted occurrences of the word t_k in the training data for class c_j in the slot s_m , V_{c_j} is the total number of unique words in class c_j , and V is the total number of unique words across all classes. $N(t_k, c_j, s_m)$ is computed as follows:

$$N(t_k, c_j, s_m) = \sum_{i=1}^{|TR|} w_j^i n_{kim} \quad (10)$$

In (10), n_{kim} is the number of occurrences of the term t_k in the slot s_m of the i^{th} instance. The sum of all $N(t_k, c_j, s_m)$ in the denominator of equation (9) denotes the total weighted length of the slot s_m in the class c_j . The final outcome of the learning process is a probabilistic model used to classify a new instance in the class c_+ or c_- . The model can be used to build a personal profile that includes those words that turn out to be most indicative of the user's preferences, according to the value of the conditional probabilities in (9).

6 Experimental Evaluation of Synset-Based Profiles

The goal of this phase is to evaluate whether the BOS versions of ITR and RocchioProfiler actually improve the performances with respect to the BOW

versions of the systems. For this purpose, two experimental sessions have been conducted, one for each system. The experimental work has been carried out on a collection of 1,628 textual descriptions of movies rated by 72,916 real users, the EachMovie dataset¹. The movies are rated on a 6-point scale that was mapped linearly into the interval [0,1]. The original dataset does not contain any information about the content of the movies. The content information for each movie was collected from the Internet Movie Database². We gathered the *Title*, the *Director*, the *Genre*, that is the category of the movie, the list of *Keywords*, the *Summary* and the *Cast*. Appropriate preprocessing (tokenization, stopword elimination, stemming) has been performed to represent documents in the BOW model. The content of slots *title*, *director* and *cast* was only tokenized because we observed that the process of stopwords elimination produced some unexpected results: for example, after stopword elimination, the slots *title* that are made exclusively of stopwords (like “*It*”) became empty. Moreover, it makes no sense to perform stemming and stopwords elimination on slots containing only proper names. In order to represent them also in the BOS model, documents in the dataset have been disambiguated using Algorithm 1, obtaining a reduction of the number of features (close to 38% - see Table 1). This result is due to the fact that synonym words have been represented by the same synset. Bigram recognition gave a minor contribution to feature reduction. It should be noticed also that only few people (like ‘Stanley Kubick’) are recognized by WordNet, thus a low number of entities is included in the BOS model.

Table 1. Number of features used to represent movies

Slot	# Feature BOW	# Feature BOS
Title	3,080	2,516
Cast	46,568	352
Director	3,559	120
Summary	77,015	67,217
Keywords	42,074	37,785
Total	172,296	107,990

Movies are subdivided into different genres: *Action*, *Animation*, *Classic*, *Comedy*, *Art_Foreign*, *Drama*, *Family*, *Horror*, *Romance*, *Thriller*. For each genre or category, a set of 100 users was randomly selected among users that rated n items, $30 \leq n \leq 100$ in that movie category (only for ‘animation’, the number of users that rated n movies was 33, due to the low number of movies in that genre). For each category, a dataset of at least 3000 triples (user,movie,rating) was obtained (at least 990 for ‘animation’). Table 2 summarizes the data used for the experiments. The number of movies rated as positive and negative in that genre is balanced in datasets 2, 5, 7, 8 (55-70 % positive, 30-45% negative), while is unbalanced in datasets 1, 3, 4, 6, 9, 10 (over 70% positive).

¹ EachMovie dataset no longer available for download:

<http://www.cs.umn.edu/Research/GroupLens/>

² IMDb, <http://www.imdb.com>

Table 2. 10 ‘Genre’ datasets obtained from the original EachMovie dataset

Id	Genre	Genre	Number ratings	% POS	% NEG
1		Action	4,474	72	28
2		Animation	1,103	57	43
3		Art.Foreign	4,246	76	24
4		Classic	5,026	92	8
5		Comedy	4,714	63	37
6		Drama	4,880	76	24
7		Family	3,808	64	36
8		Horror	3,631	60	40
9		Romance	3,707	73	27
10		Thriller	3,709	72	28
			39,298	72	28

As our content-based profiling systems are conceived as text classifiers, their effectiveness is evaluated by classification accuracy measures *precision* (Pr) and *recall* (Re) [16]. Also used is F1 measure, a combination of precision and recall:

$$F1 = \frac{2 \times Re \times Pr}{Pr + Re}$$

We adopted the Normalized Distance-based Performance Measure (NDPM) originally proposed by Yao [18] to compare the ranking imposed by the user ratings with the classification scores given by both RocchioProfiler (the similarity score for the class *likes*) and ITR (the a-posteriori probability of the class *likes*). Values range from 0 (agreement) to 1 (disagreement). The adoption of both classification accuracy and rank accuracy metrics gives us the possibility to evaluate both whether the systems are able to recommend good items and how these items are ranked. For example, even if the top ten items ranked by the systems were relevant, a rank accuracy metric might give a low value because the best item is actually ranked 10th. In all the experiments, a movie description d_i is considered as *relevant* by a user if the rating is greater or equal to 4. RocchioProfiler considers an item as relevant if the similarity score for the class *likes* is higher than the one for the class *dislikes*, while ITR considers an item as relevant if the a-posteriori probability of the class *likes* is greater than 0.5. We executed one experiment for each user in the dataset: the ratings of each specific user and the content of the rated movies have been used for learning the user profile and measuring its predictive accuracy, using the aforementioned measures. Each experiment consisted in: (1) selecting ratings of the user and the content of the movies rated by that user; (2) splitting the selected data into a training set Tr and a test set Ts ; (3) using Tr for learning the corresponding user profile; (4) evaluating the predictive accuracy of the induced profile on Ts , using the aforementioned measures. The methodology adopted for obtaining Tr and Ts was the 10-fold cross validation. Table 3 reports the results obtained over all 10 genres by RocchioProfiler. We notice a 2% improvement on average in precision of the BOS model over the BOW model. In more detail, the BOS model

Table 3. Comparison between BOW and BOS profiles obtained by RocchioProfiler

Id Genre	Precision		Recall		F1		NDPM	
	BOW	BOS	BOW	BOS	BOW	BOS	BOW	BOS
1	0.74	0.75	0.84	0.86	0.76	0.79	0.46	0.44
2	0.65	0.64	0.70	0.70	0.68	0.63	0.34	0.38
3	0.77	0.85	0.80	0.87	0.77	0.84	0.46	0.48
4	0.92	0.94	0.94	0.96	0.93	0.94	0.45	0.43
5	0.67	0.69	0.72	0.75	0.67	0.70	0.44	0.46
6	0.78	0.79	0.84	0.87	0.80	0.81	0.45	0.45
7	0.68	0.74	0.79	0.84	0.73	0.77	0.41	0.40
8	0.64	0.69	0.78	0.84	0.69	0.73	0.42	0.44
9	0.75	0.76	0.83	0.85	0.76	0.77	0.48	0.48
10	0.74	0.75	0.84	0.85	0.77	0.78	0.45	0.44
Mean	0.74	0.76	0.81	0.84	0.76	0.78	0.44	0.44

outperforms the BOW model on datasets 3 (+8%), 7 (+6%), 8 (+5%). Only dataset 2 showed no improvement. This is probably due both to the low number of ratings and to the specific features of the movies, in most cases stories, that makes difficult the disambiguation. Also recall and F1-measure (+3%) obtained by the BOS model are improved over those obtained by the BOW model. In particular, the improvement was observed again on dataset 3 (+7%), 7 (+5%), 8 (+6%). This could be an indication that the improved results are independent from the distribution of positive and negative examples in the datasets: datasets 7 and 8 are balanced, while dataset 3 is unbalanced. NDPM has not been improved, but it remains acceptable. This could be interpreted as the BOS method has improved the classification of items whose score (and ratings) is close to the relevant / not relevant threshold, thus items for which the classification is highly uncertain. A Wilcoxon signed ranked test ($p < 0.05$) has been performed to validate the results [12]. We considered each experiment as a single trial for the test. The test confirmed that there is a statistically significant difference in favor of the BOS model with respect to the BOW model as regards precision, recall and F1-measure, and that the two models are equivalent in defining the ranking of the preferred movies with respect to the score for the class “likes”. The results of the comparison between the profiles obtained from documents represented using the two indexing approaches by ITR are reported in Table 4. We can notice a significant improvement of BOS over BOW both in precision (+8%) and recall (+10%). The BOS model outperforms the BOW model on datasets 5 (+11% of precision, +14% of recall), 7 (+15% of precision, +16% of recall), 8 (+19% of precision, +24% of recall). Only on dataset 4 we have not observed any improvement, probably because precision and recall are already very high, thus there isn’t much room for improvement. These results could be an indication that ITR improvements depend on the balanced distribution of positive and negative examples in the dataset (see Table 2). NDPM remains stable, even if classification accuracy was improved. The Wilcoxon test confirmed that there is a statistically significant difference in favor of the BOS model with respect

Table 4. Performance of the two versions of ITR on 10 different datasets

Id	Genre	Precision		Recall		F1		NDPM	
		ITR	ITR	ITR	ITR	ITR	ITR	ITR	ITR
		BOW	BOS	BOW	BOS	BOW	BOS	BOW	BOS
1		0.70	0.74	0.83	0.89	0.76	0.80	0.45	0.45
2		0.51	0.57	0.62	0.70	0.54	0.61	0.41	0.39
3		0.76	0.86	0.84	0.96	0.79	0.91	0.45	0.45
4		0.92	0.93	0.99	0.99	0.96	0.96	0.48	0.48
5		0.56	0.67	0.66	0.80	0.59	0.72	0.46	0.46
6		0.75	0.78	0.89	0.92	0.81	0.84	0.46	0.45
7		0.58	0.73	0.67	0.83	0.71	0.79	0.42	0.42
8		0.53	0.72	0.65	0.89	0.58	0.79	0.41	0.43
9		0.70	0.77	0.83	0.91	0.75	0.83	0.49	0.49
10		0.71	0.75	0.86	0.91	0.77	0.81	0.48	0.48
Mean		0.67	0.75	0.78	0.88	0.73	0.81	0.45	0.45

to the BOW model as regards precision, recall and F1-measure, and that the two models are equivalent in defining the ranking of the preferred movies with respect to the score for the class “likes”. The conclusion is that both techniques significantly improves their overall accuracy when shifting from BOW to BOS: +2% F1 improvement for RocchioProfiler, +8% F1 improvement for ITR.

7 Conclusions

We presented a strategy that integrates machine learning methods with WSD based on WordNet for inducing semantic user profiles. Our hypothesis is that substituting words with synsets produces a more accurate document representation that could be successfully used by learning algorithms to infer accurate user profiles. This hypothesis is confirmed by the experimental results obtained by two different learning methods.

Acknowledgments

This research was partially funded by the European Commission under the 6th Framework Programme IST Integrated Project VIKEF No. 507173, Priority 2.3.1.7 Semantic-based Knowledge Systems - <http://www.vikef.net>.

References

1. F. Asnicar and C. Tasso. ifweb: a prototype of user model-based intelligent agent for documentation filtering and navigation in the word wide web. In *Proc. of 1st Int. Workshop on adaptive systems and user modeling on the WWW*, 1997.
2. M. Balabanovic and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

3. S. Bloedhorn and A. Hotho. Boosting for text classification with semantic features. In *Proc. of 10th ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining, Mining for and from the Semantic Web Workshop*, pages 70–87, 2004.
4. M. Degemmis, P. Lops, S. Ferilli, N. D. Mauro, T. Basile, and G. Semeraro. Learning user profiles from text in e-commerce. In *Proceedings of 1st Int. Conf. on Advanced Data Mining and Applications, ADMA 2005*, number 3584 in Lecture Notes in Computer Science, pages 370–381. Springer-Verlag, 2005.
5. M. Degemmis, P. Lops, and G. Semeraro. Wordnet-based word sense disambiguation for learning user profiles. In *Proc. of the 2nd European Web Mining Forum*, 2005.
6. C. Leacock and M. Chodorow. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press, 1998.
7. B. Magnini and C. Strapparava. Improving user modelling with content-based techniques. In *Proc. 8th Int. Conf. User Modeling*, pages 74–83. Springer, 2001.
8. C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, US, 1984.
9. A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *Proceedings of the AAAI/ICML-98 Workshop on Learning for Text Categorization*, pages 41–48. AAAI Press, 1998.
10. G. A. Miller. Wordnet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.
11. T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.
12. M. Orkin and R. Drogin. *Vital Statistics*. McGraw-Hill, New York, 1990.
13. M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.
14. J. Rocchio. Relevance feedback information retrieval. In G. Salton, editor, *The SMART retrieval system - experiments in automated document processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.
15. S. Scott and S. Matwin. Text classification using wordnet hypernyms. In *COLING-ACL Workshop on usage of WordNet for in NLP Systems*, pages 45–51, 1998.
16. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 2002.
17. I. Witten and T. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4), 1991.
18. Y. Y. Yao. Measuring retrieval effectiveness based on user preference of documents. *Journal of the American Society for Information Science*, 46(2):133–145, 1995.

Audiovisual Integration for Racquet Sports Video Retrieval

Yaqin Zhao¹, Xianzhong Zhou², and Guizhong Tang³

¹ College of Mechanical and Electronic Engineering, Nanjing Forestry University, 210037 Nanjing, China

yaqinzhao@163.com

² School of Management and Engineering, Nanjing University, 210093 Nanjing, China
zhouxizh@public1.ptt.js.cn

³ School of Automation, Nanjing University of Technology, 210009 Nanjing, China
tanggz128@163.com

Abstract. This paper presents a new audiovisual integration scheme for racquet sports video structure indexing and highlight generating. Instead of using low-level features, the method is built upon the combination of visual and audio features. With respect to prior information about this kind of video content and editing rules, visual features based on dominant color and motion attention model are applied to classify shots into two classes: global view shots and non-global view shots. The classification algorithm is independent of predefined court color, and much robust to lighting conditions. Afterwards, among shots important auditory features including both ball hitting and applause are detected for identifying interesting events with strong semantic meaning, such as missed serves, aces, rallies and replays in tennis video. Finally, a reasonable model is built to rank rally events by excitement. The results showed the scheme could effectively identify typical scenes for retrieving highlights.

1 Introduction

Retrieval and summarization of sports video have retrieved increasingly interest in recent years. However, the research is mainly following the work on field sports video (soccer video and baseball video). Little attention has been paid to racquet sports video analysis, although there are some works done on tennis. Hidden Markov model has been used for recognition of strokes [1] and structure analysis of tennis video [2] [3]; several approaches [3-5] have been proposed for event detection and structure analysis using multimedia information.

Although some recent efforts have been made on tennis video, few researches focused on a general method for different kinds of racquet sports video analysis except for [6]. This paper presents a fusion scheme of visual and auditory modalities to detect interesting events with strong semantic meaning in racquets video. Instead of grouping video shots into various clusters using an unsupervised clustering algorithm [6], in our classification algorithm, two shot classes are identified only, global views and non-global view. Segmented shots are classified based on both low-level visual features and perceptual attention of video shots. Therefore, the similarity measure is

more robust to strong variations in illumination conditions than only using low-level visual features, which is especially important for outdoor tennis match. In addition, it is unnecessary for first evaluating the game court color. Typical scenes have a strong relationship with obvious audio features. Thus SVM is applied to detect important audio features for identifying interesting events with strong semantic meaning. Finally, a model is built to rank rally events by excitement.

The remainder of this paper is organized as follow. Section 2 and 3 present the details of video shots classification and audio classification based on SVM. In section 4 we focus on interesting events detection and rally ranking, Experimental results are presented and discussed in section 5, and finally, in section 6, we give our conclusions.

2 Semantic Shot Classification

In a racquet video, global views contain much of the pertinent information. Thus we identify global views from all segmented shots. The classification algorithm labels the shots according to two classes: global views and non-global views. The whole process is carried out in MPEG compressed domain.

After sports video is segmented into shots, the content of each shot is represented by a single keyframe because one keyframe is usually enough to illustrate the whole content of a view for a racquet video. Global views are characterized by rather homogeneous color content, although other views are characterized by scattered color content. Therefore the color content similarity between two shots is measured by using the intersection of normalized color histogram of their keyframes.

Motion attention model is addressed to compute the attention of humans when their viewing videos and used for the generation of video skimming [7] [8]. In this paper, motion attention model is used to select the features of motion objects attended by people. Global view shots are mainly composed of play shots. As we know, the number of players is 2 or 4 in a racquet sports. Human attention usually focuses on ball and players in play shots, whereas one player or other interesting persons are usually attended in non-global views. Hence the number of motion objects is fixed in most global views, but the number is indefinite for non-global views. Moreover, visual size of motion objects in most global views are also more stable than that in other views. Therefore, a motion attention model is utilized to compute the motion attention regions of video shots. For example, players and ball are motion attention areas in a rally shot. The number of located motion regions is restricted to at most 3 since it is hard for humans to focus on more than 3 objects simultaneously. So for each keyframe a threshold β_i is set to remove the regions that their areas are lower than β_i . Only the first three biggest regions are hold in one keyframe. The β_i value is set as $\beta_i = 0.01MaxRS_i$ considering the proportion of one player to one ball in motion attention region, where $MaxRS_i$ denotes the biggest attention region of one keyframes k_i . The located motion-attended regions of three typical views in tennis videos are showed in Fig.1. Based on the observations and analysis above, we choose three features to identify global views. One is color feature (low-level features), and other two ones are the number and the size of motion attention regions (high-level features). The similarity between two shots is defined as a weighted function:

$$Sim(S_i, S_j) = Sim(k_i, k_j) = \omega_1 F_{color} + \omega_2 F_{number} + \omega_3 F_{size} \tag{1}$$

$$F_{color} = 1 - \frac{1}{A(k_i, k_j)} \sum_h \sum_s \sum_v \min\{H_i(h, s, v), H_j(h, s, v)\} \tag{2}$$

$$F_{number} = |RN_i - RN_j| / 3 \tag{3}$$

$$F_{size} = |MaxRS_i - MaxRS_j| / \max(MaxRS_i, MaxRS_j) \tag{4}$$

where $A(k_i, k_j) = \min\left\{ \sum_h \sum_s \sum_v H_i(h, s, v), \sum_h \sum_s \sum_v H_j(h, s, v) \right\}$, $H_i(h, s, v)$ is the

normalized 256-bin HSV color histogram of the keyframe k_i , RN_i is the number of motion attention regions of the keyframes k_i , and $MaxRS_i$ denotes the biggest attention region of the keyframes k_i .

Since the color content of global views includes much more pixels of match court than that of other views, dominant colors ratio is usually much higher than that in other views. Thus dominant colors ratios are used to find a reference keyframe of a global view. Here, the dominant color of one keyframe denotes highest percentage of the color in all colors of the keyframe. We select m keyframes whose dominant color ratios are more than 40% as candidate reference keyframes, and then minimize the median distance of all the candidate reference keyframes using least median square method. Finally, one randomly selected reference keyframe k_{ref} is obtained. For a given threshold β_1 , if the similarity between one keyframe k_i and the reference keyframe meets,

$$Sim(k_i, k_{ref}) \leq \beta_1 \tag{5}$$

and then the shot corresponding with the keyframe k_i will be classified into the global views, otherwise it will be identified as a non-global views.

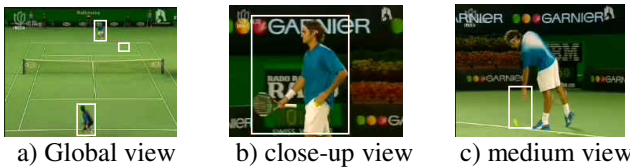


Fig. 1. Motion attention regions of three typical views in tennis videos

3 Audio Classification

Even though most of global views are rallies or ace scenes, it is not sufficient to base the judgment solely on visual features since some rally shots are partitioned into the non-global view shot class. In addition, it is difficult to fine identify interesting events in terms of visual information only. Audio features can help reduce the amount of

false alarms and identify more events. In racquet sports, there are several major audio events, such as ball hitting, applause, cheer and commentator speech.

As mentioned previously, the video stream is segmented in a sequence of shots. The shots are considered as the base entity and features describing audio content for each shot are extracted to identify interesting events. We have carried out experiments based on SVM to evaluate the performance of audio features in the classification. The selected features are the mean value and the standard deviation of Short-time energy, the mean value and the standard deviation of Zero-crossing rate, the standard deviation of different Zero-crossing rate, and the mean value and the standard deviation of MFCC. Both the training and testing clip length is chosen as one second in this paper.

4 Interesting Events Identification and Ranking

Generally speaking, racquet sports has no distinct exciting events e.g., “shot on goal” events in soccer video. But people further hope to browse these sports video by play scenes, such as serves and rally scenes in tennis. Besides, replay shots are exciting parts or highlights of one match. Based on the above analysis, we identify four interesting events in tennis: missed serve, ace, rally and replay.

4.1 Audiovisual Integration for Event Identification

We take example for tennis video to present our audiovisual integration scheme for event identification. As mentioned previously, the non-global view shot class can include some break shots. We can reduce the false alarms by detecting whether ball hitting and applause at the end of exchange happen or not, as showed in Fig.2. In deed, a rally and a serve are coupled with the presence of ball hitting and applause which happen when a player score a point, whereas a missed serve is only characterized by the presence of ball hitting. Ace scenes are discriminated from rally scenes by taking into account the number of ball hitting sound in one shot because the ball is usually hit only once in ace scenes.

Replay scenes are used for recurring exciting moment in one match. Before replay scenes, special transitions are inserted to separate a replay from the live broadcast. Therefore replay scenes can be detected based on dissolve transition identification. In other words, the shots between two dissolve transitions are probably a replay.

Rule 1: IF there is no ball hitting and no applause at the end of one global view shot
THEN the shot is a break shot
ELSE it is a play shot

Rule 2: IF both ball hitting and applause exist at the end of one non-global view shot
THEN the shot is a play shot
ELSE it is a break shot

Rule 3: IF there is no applause at the end of one play shot
THEN it is a missed serve
ELSE judge whether the number of ball hitting sound is 1
IF it is 1 THEN the shot is ace
ELSE the shot is rally

Rule 4: IF there is dissolve transition right before and after one break shot
 THEN judge whether there is plain speech and whether there is no music coupling with it
 IF both are satisfied THEN it is a replay
 ELSE it is not

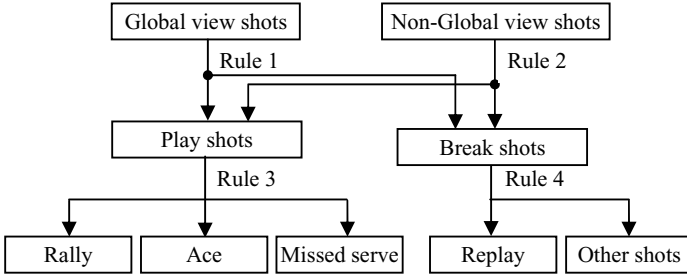


Fig. 2. The method for event detection

4.2 Rally Ranking

After rallies are identified, we rank these rallies by their exciting degree to more suitable for human skimming requirement. A model of rally scenes is built based on general human opinion on the judgment of racquet sports highlights. It seems that people prefer to watch the long time confront, therefore the duration of a rally is an important factor for ranking. After highlight, long time applauses are nature response of the audience. Based on the above analysis, a model of exciting degree of a rally is defined as,

$$R_{rally} = w_{duration} T_{duration} + w_{applause} E_{applause} \tag{6}$$

where R_{rally} is a weight function, $T_{duration}$ denotes the duration of a rally, and $E_{applause}$ is the relative energy of the applause. The higher the R_{rally} value, the more exciting the corresponding rally will be.

In racquet video, the replay recurs the exciting moment just happened. The shot right before a replay scene should be thus sorted prior to all other rallies. First, the rallies replayed are labeled and classified into one group. And then, the R_{rally} value of each rally is computed to evaluate its exciting degree. All labeled rallies are first sorted in descending order according to their R_{rally} values. Namely, all replayed shots are sorted prior to all other non-labeled rallies. Also non-labeled rallies are sorted in descending order according to their R_{rally} values.

5 Experimental Results

Experimental data were composed of 7 tennis video clips, 9 table tennis video clips and 12 badminton video clips (each clip correspond with a game). The original

signals were digitized at 25frames/s, which is the basic unit for feature extraction. The audio signal samples were collected with 44.1kHz and 16 bit/sample. For tennis, 3 video clips were used for audio training, and other 4 video clips were used for testing. In table tennis, 4 video clips were used for audio training, and other 5 video clips were used for testing. In badminton, 5 video clips were used for audio training, other 7 different video clips were used for testing.

5.1 The Threshold β_1

The threshold β_1 is introduced to judge whether one keyframe is similar to reference keyframe. Therefore, the threshold β_1 affects shot classification results based on visual features to a certain extent. As seen from Fig. 3, when the value of β_1 is set too high ($0.3 \leq \beta_1 < 0.5$), satisfactory clustering results cannot be obtained since some medium view shots are falsely classified into the global view class. In addition, considering refinement by audio features, the lower threshold β_1 should be set. Thus it is suitable to set as $0.2 \leq \beta_1 < 0.3$.

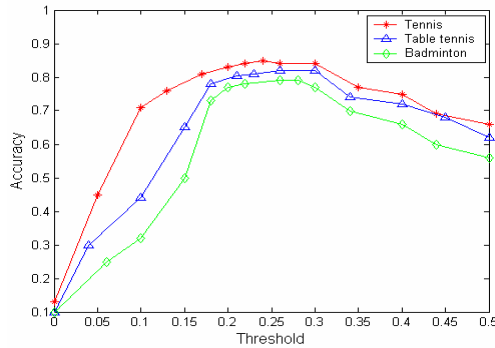


Fig. 3. The threshold β_1

5.2 Event Detection

The classification rates of 4 tennis video clips were averaged, and the results are showed in Table 1. For only using visual features, the shot duration were used for distinguishing serve scenes from rallies. Missed serves and aces were distinguished by the cumulative duration of consecutive non-global views. It can be seen that the classification rate of serve scenes and the recall rate of rallies are low, because some short rallies are identified as serve serves. The high precision rate of rallies showed that the similarity measures work well to discriminate non-global view shots since few non-global views are classified as rally. For solely using audio features, both precision rate and recall rate for rallies, missed serves and aces are increased. This suggests that audio features are more effective to describe rally scenes than visual features. Actually, as we mentioned previously, a rally is essentially coupled with ball

hitting and applause, although a missed serve is only characterized by the presence of ball hitting. Thus there are less confusion between serves and rallies. However, replays detection presents a low recall rate because they are not characterized by a representative audio content. After using both audio-visual features, there is much improvement for all typical events detection.

Table 1. Events identification results in tennis

Events	Visual features only		Audio features only		Audiovisual integration	
	Precision	Recall	Precision	Recall	Precision	Recall
Rallies	94.2%	41.3%	92.1%	80.2%	98.6%	92.5%
Aces	72.6%	67.1%	93.6%	82.2%	95.2%	90.2%
Missed serve	69.5%	52.4%	83.8%	81.5%	86.4%	88.3%
Replay	97.0%	82.8%	67.3%	49.9%	87.3%	90.8%
Average	83.3%	60.9%	84.2%	73.4%	91.8%	90.4%

The results of 5 table tennis clips are also listed in Table 2. It can be seen that the results of the rally event detection only using single features is not satisfactory. Sometimes the ball hitting is mixed with some other sounds, thus some rallies without ball hitting detected are missed by audio classification. In additional, some shots are not easy to be correctly classified only using visual features. In our method, there are two significant improvements for identifying rally scenes. Those missed rally scenes by video classification are successfully detected with the compensation of visual features. Meanwhile, those shots falsely classified by visual classification are repartitioned into the correct classes depending on the rectification of audio features.

Table 2. Rally identification results in table tennis

Video clips	Visual features only		Audio features only		Audiovisual integration	
	Precision	Recall	Precision	Recall	Precision	Recall
Clip 1(16:32)	82.5%	73.6%	79.4%	67.7%	85.1%	80.1%
Clip 2(30:19)	86.6%	82.3%	71.3%	68.1%	89.6%	88.4%
Clip 3(28:45)	76.7%	68.7%	75.9%	73.3%	80.7%	82.7%
Clip 4(36:29)	81.2%	70.4%	69.5%	70.8%	83.2%	91.7%
Clip 5(39:50)	83.9%	71.6%	80.5%	71.9%	85.2%	90.4%
Average	82.1%	73.3%	75.3%	70.3%	84.7%	86.6%

6 Conclusions

An integration scheme of audio and visual features is presented for racquet sports video retrieval. Both dominant color and human motion attention are regarded as visual features for computing similarity between two keyframes. Therefore, comparing with color-based classification methods, the classification algorithm is less constrained by color distribution and lighting condition. Rally ranking model is built based on broadcast video editing rules (replay) and human opinion on the judgment of racquet sports highlights. Three kinds of racquet sports video data totaling with 28

different video clips are chosen to perform experiments. The results showed an encouraging improvement in structure analysis and highlight extraction by combining audio and visual cues.

Acknowledgement

We are grateful for the support of Natural Science Council of Jiangsu in China under grant BK2004137.

References

1. M. Petkovic, Z. Zivkovic, W. Jonker: Recognizing Strokes in Tennis Videos using Hidden Markov Models. In: Proceedings of IASTED International Conference on Visualization, Imaging and Image Processing, Spain, (2001)
2. E. Kijak, L. Oisel, and P. Gros: Temporal Structure Analysis of Broadcast Tennis Video using Hidden Markov Models. In: SPIE Storage and Retrieval for Media Databases, (2003) 289–299
3. E. Kijak, G. Gravier, L. Oisel, P. Gros: Audiovisual Integration for Tennis Broadcast Structuring. In: Proceedings of International Conference on Multimedia and Exhibition, (2003)
4. R. Dayhot, A. Kokaram, N. Rea: Joint Audio Visual Retrieval for Tennis Broadcasts. In: IEEE International Conference on Acoustics, Speech, & Signal Processing, Hong Kong, (2003)
5. M. Xu, LY. Duan, CS. Xu, Q. Tian: A Fusion Scheme of Visual and Auditory Modalities for Event Detection in Sports Video. In: IEEE International Conference on Acoustics, Speech, & Signal Processing, Hong Kong, (2003) 333-336
6. LY. Xing, QX. Ye, WG. Zhang: A Scheme for Racquet Sports Video Analysis with the Combination of Audio-Visual Information. In: Proceedings of International Conference on Visual Communications and Image Processing, Vol. 5960, (2005)
7. YF. Ma, HJ. Zhang: A Model of Motion Attention for Video Skimming. In: Proceedings of International Conference on Image Processing, Vol. 1, (2002) 129-132
8. CW. Ngo, YF. Ma, HJ. Zhang: Video Summarization and Scene Detection by Graph Modeling. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 2, (2005) 296-305

A Correlation Approach for Automatic Image Annotation^{*}

David R. Hardoon¹, Craig Saunders¹, Sandor Szedmak²,
and John Shawe-Taylor¹

¹ University of Southampton, ISIS Research Group, Southampton, U.K.

² University of Helsinki, Department of Computer Science, Helsinki, Finland

Abstract. The automatic annotation of images presents a particularly complex problem for machine learning researchers. In this work we experiment with semantic models and multi-class learning for the automatic annotation of query images. We represent the images using scale invariant transformation descriptors in order to account for similar objects appearing at slightly different scales and transformations. The resulting descriptors are utilised as visual terms for each image. We first aim to annotate query images by retrieving images that are similar to the query image. This approach uses the analogy that similar images would be annotated similarly as well. We then propose an image annotation method that learns a direct mapping from image descriptors to keywords. We compare the semantic based methods of Latent Semantic Indexing and Kernel Canonical Correlation Analysis (KCCA), as well as using a recently proposed vector label based learning method known as Maximum Margin Robot.

1 Introduction

Due to an increasing rise of multimedia data that is available both on-line and off-line, we are faced with the problematic issue of our ability to access or make use of this information, unless the data is organised in such a way that allows efficient browsing, searching and retrieval. One of these issues is image labelling or multi-labelling where we would like to annotate an image with several keywords that best describe it. Several solutions have been proposed using keyword association to images and image segments [1,2,14,18].

Recently in [7,8], it was suggested that methods that use region-based image descriptors generated by automatic segmentation or through fixed shapes may lead to poor performance, as regularly used rectangular regions image descriptors are not robust to a variety of transformations such as rotation. They have suggested using Scale Invariant Feature Transformation (SIFT) [9] feature, which are scale invariant, and utilising them as ‘visual’ terms in a document. We then have a bag-of-visualterms model for each image, and this can then be processed in a similar fashion to bag-of-words models for text documents.

^{*} The authors would like to acknowledge the financial support of the European Community IST Programme; PASCAL Network of Excellence grant no. IST-2002-506778.

In this work we follow the layout suggested by [8] and test their proposed annotation approach with KCCA and Maximum Margin Robot (MMR)[15], a new vector label based learning method. We also suggest learning the association between the keywords and images directly, and therefore learning the association between keywords and particular SIFT descriptors. When a new query image is encountered new keywords could be predicted/generated according to its SIFT descriptors.

The paper is laid-out as follows. In Section 2 we introduce Latent Semantic Indexing and its usage in this context. We continue the semantic model discussion by describing in detail Kernel Canonical Correlation Analysis in Section 3. In Section 4 we discuss Maximum Margin Robot a new vector label based learning method. Section 5 describes the data representation used in this work. This is followed by the experimental setup in Section 6 and our presented results in Section 7. Our final remarks and discussion are given in Section 8.

2 Latent Semantic Indexing

Latent Semantic Indexing (LSI)¹ is a classical approach to information retrieval. This approach is a vector based information retrieval method that uses a training collection. Given a term document training matrix A (or image training matrix) with rows as training examples, LSI uses the Singular Value Decomposition (SVD) to factor A into its singular vectors. We are able to apply a noise reduction on the data by projecting the training data into the computed k largest singular vectors. LSI uses this in order to learn the structure of the training collection and to project new test queries into the same semantic space. We are able to write SVD as $A' = U\Sigma V'$, where X' is the transpose of a matrix or vector X . We denote the k -dimensional approximation of A as $\tilde{A}' = U_k\hat{\Sigma}_kV'_k$. The rank reduced \tilde{A}' is an approximation of the of the original A' and V_k is the data in the projected semantic space, which can be seen in the following

$$V'_k = \hat{\Sigma}_k^{-1}U'_k\hat{\Sigma}_kV'_k = \hat{\Sigma}_k^{-1}U'_k\tilde{A}' = (\tilde{A}U_k\hat{\Sigma}_k^{-1})'.$$

Since we are looking for a similarity measure, we project the query document \mathbf{q} into the k semantic feature space of A and look for the closest matching image from the training corpus. Therefore, $\max_i \langle \mathbf{v}_i^k, \mathbf{q}U_k\hat{\Sigma}_k^{-1} \rangle$ will give us the image from the training corpus with the largest inner project with the query image. Where \mathbf{q} is the query image vector and \mathbf{v}_i^k are the row vectors of V_k .

3 Kernel Canonical Correlation Analysis

Proposed by Hotelling in 1936, Canonical Correlation Analysis (CCA) is a technique for finding pairs of basis vectors that maximise the correlation between

¹ Also known as Latent Semantic Analysis (LSA).

the projections of paired variables onto their corresponding basis vectors. Correlation is dependent on the chosen coordinate system, therefore even if there is a very strong linear relationship between two sets of multidimensional variables this relationship may not be visible as a correlation. CCA seeks a pair of linear transformations one for each of the paired variables such that when the variables are transformed the corresponding coordinates are maximally correlated.

Consider the linear combination $x = \mathbf{w}'_x \mathbf{x}$ and $y = \mathbf{w}'_y \mathbf{y}$. Let \mathbf{x} and \mathbf{y} be two random variables from a multi-normal distribution, with zero mean. The correlation between x and y can be defined as $\max_{\mathbf{w}_x, \mathbf{w}_y} \rho = \mathbf{w}'_x C_{\mathbf{x}\mathbf{y}} \mathbf{w}_y$ subject to $\mathbf{w}'_x C_{\mathbf{x}\mathbf{x}} \mathbf{w}_x = \mathbf{w}'_y C_{\mathbf{y}\mathbf{y}} \mathbf{w}_y = 1$. $C_{\mathbf{x}\mathbf{x}}$ and $C_{\mathbf{y}\mathbf{y}}$ are the non-singular within-set covariance matrices and $C_{\mathbf{x}\mathbf{y}}$ is the between-sets covariance matrix.

We suggest using the kernel variant of CCA [4] since due to the linearity of CCA useful descriptors may not be extracted from the data. This may occur as the correlation could exist in some non linear relationship. The kernelising of CCA offers an alternate solution by first projecting the data into a higher dimensional feature space $\phi : \mathbf{x} = (x_1, \dots, x_n) \rightarrow \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_N(\mathbf{x}))$ ($N \geq n$) before performing CCA in the new feature space.

Given the kernel functions κ_a and κ_b let $K_a = \mathbf{X}\mathbf{X}'$ and $K_b = \mathbf{Y}\mathbf{Y}'$ be the kernel matrices corresponding to the two representations of the data. Let \mathbf{X} be the matrix whose rows are the vectors $\phi_a(\mathbf{x}_i)$, $i = 1, \dots, \ell$ and similarly \mathbf{Y} be a matrix with rows $\phi_b(\mathbf{y}_i)$. Substituting into primal CCA equation gives $\max_{\alpha, \beta} \rho = \alpha' \mathbf{K}_a \mathbf{K}_b \beta$ subject to $\alpha' \mathbf{K}_a^2 \alpha = \beta' \mathbf{K}_b^2 \beta = 1$. This is the dual form of the primal CCA optimisation problem given above, which can be cast as a generalised eigenvalue problem and for which the first k generalised eigenvectors can be efficiently found.

The theoretical analysis shown in [5] suggests to regularise kernel CCA as it shows that the quality of the generalisation of the associated pattern function is controlled by the sum of the squares of the weight vectors norms. Due to space limitation we refer the reader to [5,6] for a detailed analysis and the regularised form of KCCA. One aspect we will mention here though is that it is not the case that when using a linear kernel KCCA reduces to standard CCA (see the aforementioned articles for details). Using a linear kernel and KCCA has advantages over CCA, the most prominent of which in our case is speed; this is why we use this variant here. We are able to apply a similar procedure to that used in LSI to find the most matching image from the training corpus to the query image. Whereas here we project the data into the semantic space using a selection of the found eigenvectors corresponding to the largest correlation values.

3.1 Keywords Reconstruction

We are faced with the problem of creating a new document d^* (i.e. a set of keywords) that best matches our image query. Based on the idea of CCA we are looking for a vector that has maximum covariance to the query image with respect to the weight matrices α and β . Let $f = K_x^i \alpha$, where the vector K_x^i contains the kernelised inner products between the query image i and the images occurring in the training set. We have $\max_{d^*} \langle f, W_y' d^* \rangle$, where W_y is the matrix

containing the weight vectors as rows. The need to use the weight vectors for the documents limits us to the use of linear kernels.

For simplicity we assume that the expected structure of the document is of a single keyword that is the most relevant keyword for the query image. Let n be the number of known keywords in the training dataset. We may say that the vector d^* gives a convex combination of the columns of the identity matrix (i.e. $\|d^*\| = 1$), thus it satisfies the constraints

$$\sum_{i=1}^n d_i^* = 1, \quad d_i^* \geq 0 \quad i = 1, \dots, n. \quad (1)$$

The problem becomes $\max_{d^*} f'W_y'd^*$ under the same constraints. Let $c = f'W_y'$ we have $\max_{d^*} cd^*$. Due to the constraints in equation (1) the components of the optimum solution d^* is equal to

$$(d)_i^* = \begin{cases} 1 & i = \arg \max_j c_j, \\ 0 & \text{otherwise.} \end{cases}$$

This generates a document containing a single keyword. We modify the original maximisation problem to relax the optimum solution to include keywords above a threshold T . The new relaxed formulation will generate a document with varying number of keywords, depending on T . We are able to use the value of c_j to rank the relevance of the selected keywords. We do this by sorting the values of \mathbf{c} and taking the keywords relating to the largest values of \mathbf{c} above threshold T .

4 Maximum Margin Robot

The Support Vector Machine(SVM) has been shown to be a very useful method of machine learning, but is restricted to directly solving binary classification problems only. There is a strong demand for extending the underlying idea towards multi-class classification and learning when the outputs have complex structure. The known approaches are tackling with the exploding computational complexity and the range of potential applications becomes very limited. There is a straightforward algebraic generalisation of the SVM which can handle arbitrary vector outputs and preserves the same computational complexity of its binary ancestor. The structural learning problems can then be solved via an embedding into a properly chosen vector space. The learning strategy in the vector label learning can be stated as a three-phase process:

Embedding: where the structures of the input and output objects are represented in properly chosen Hilbert spaces, reflecting the similarity and the dissimilarity of the objects.

Optimisation: has to find the similarity based matching between the input and the output representations,

Inversion (Pre-image problem): has to recover the best fitting output structure of its vector representation.

The variant of vector valued learning we introduce was born as a reinterpretation of the variables and parameters occurring in the Support Vector Machine: In the original representation $y_i \in \{-1, +1\}$ are binary outputs and \mathbf{w} is the normal vector of the separating hyperplane. While in the new representation $y_i \in \mathcal{Y}$ are arbitrary outputs $\psi(y_i) \in \mathcal{H}_\psi$ embedded labels in a linear vector space, and \mathbf{w}^T is a linear operator projecting the input space into the output space. The output space is a one dimensional subspace in the SVM.

The details of reinterpretation of MMR² are given in Table 1. Due to limited space we refer the reader to [15] where the method is first introduced.

Table 1. SVM and MMR interpretation

Binary class learning	Vector label learning
Support Vector Machine (SVM)	Maximum Margin Robot (MMR)
$\min \frac{1}{2} \underbrace{\mathbf{w}^T \mathbf{w}}_{\ \mathbf{w}\ _2^2} + C \mathbf{1}^T \xi$	$\frac{1}{2} \underbrace{\text{tr}(\mathbf{W}^T \mathbf{W})}_{\ \mathbf{W}\ _{Frobenius}^2} + C \mathbf{1}^T \xi$
$\text{w.r.t. } \boxed{\mathbf{w} : \mathcal{H}_\phi \rightarrow \mathbb{R}}, \text{ normal vec.}$ $\boxed{b \in \mathbb{R}}, \text{ bias}$ $\xi \in \mathbb{R}^m, \text{ error vector}$	$\boxed{\mathbf{W} : \mathcal{H}_\phi \rightarrow \mathcal{H}_\psi}, \text{ linear operator}$ $\boxed{\mathbf{b} \in \mathcal{H}_\psi}, \text{ translation(bias)}$ $\xi \in \mathbb{R}^m, \text{ error vector}$
$\text{s.t. } \boxed{y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b)} \geq 1 - \xi_i$ $\xi \geq \mathbf{0}, i = 1, \dots, m$	$\boxed{\langle \psi(y_i), \mathbf{W} \phi(\mathbf{x}_i) + \mathbf{b} \rangle_{\mathcal{H}_\psi}} \geq 1 - \xi_i$ $\xi \geq \mathbf{0}, i = 1, \dots, m$

5 Data Representation

There is a great deal of importance on the textural and image means of representation, as we would like to be able to extract as much detailed information as possible for the learning process. Various approaches have been suggested such as colour moments and Gabor texture descriptors[17] as well as scale invariant interest points[10] and affine invariant interest point detector [11]. Scale Invariant Feature Transformation (SIFT) have been introduced by [9] and have been shown to be superior to other descriptors[12]. This is due to the fact that the SIFT descriptors are designed to be invariant to small shifts in position of the salient (i.e. prominent) region. SIFT transforms the image data into scale invariant coordinates relative to local features. The underlying idea of SIFT is to extract distinctive invariant features from an image such that it could be used to perform reliable matching between different views of an object or scene. Since we are aiming to learn the association of a keyword to an object, which could appear in different angles and scenes, we find SIFT ideal for the image representation.

² MMR code - <http://www.ecs.soton.ac.uk/~ss03v/mmr.html>

Documents are usually represented by word frequency. That is, the number of occurrences of each word in the document is counted and a vector of word-frequencies is created. Although this simplistic approach is usually sufficient for good performance we describe Term Frequency Inverse Document Frequency (TFIDF)[16], which computes the following

$$\text{TFIDF}(d_i, w_j) = |\{w_j \in d_i\}| \log \left(\ell |\{d_i \in D: w_j \ni d_i\}|^{-1} \right).$$

The TFIDF is a means of amplifying the influence of words that occur often in a document but relatively rarely in the whole collection. We apply the TFIDF on the image SIFT descriptors as they were post processed as to mimic the concept of words (SIFT descriptors) in documents (images), the pseudo-details of this procedure are given in the following section and further information can be found in [7]. In the experiments results section we compare the application of TFIDF on the visual terms as well as keeping them as frequency vectors.

6 Experimental Setup

We have used the University of Washington Ground Truth Image Database³, which contain 697 public-domain images that have been annotated with an average of ~ 5 keywords per image and with an overall of 287 keywords in the dictionary. [8] has kindly provided us with the post processed data. SIFT descriptors were computed from the images and then clustered using the batch k-means clustering algorithm with random starting points in order to build a vocabulary of ‘visual’ words [7]. Each image in the entire data-set then had its feature vectors quantised by assigning the feature vector to the closest cluster. This amounted into a uniform feature vector of 3000 visual terms. TFIDF was applied on the new image feature vector to amplify the influence of SIFT descriptors that occur often in an image but rarely in the whole set of images. The keywords have been stemmed, having errors corrected and merging plural terms into singular forms. Henceforth the original 287 terms were reduced to 170.

We find the frequency of the keywords in the dictionary to be very uneven⁴, therefore further reduce the keywords by removing the all keywords that only have one occurrence throughout the database. This rendered us with 132 keywords in the dictionary. The keywords were represent as a frequency vector.

We have repeated all experiments 10 times where in each repeat the database was randomly and evenly split into a training and testing set. The 10 repeats are in order to obtain some statistical verification for the used methods. In each run we use the same random split across all methods. We use the same number of dimensional selection k for the LSI semantic project as in [8] ($k = 40$) since we initially try to reproduce their LSI results.

³ <http://www.cs.washington.edu/research/imagedatabase/groundtruth/>

⁴ $\sim 33\%$ of the words have more then 10 occurrences and $\sim 3\%$ have more then 100 occurrences in the database.

Using the described method in [6,5] for the selection of the KCCA regularisation parameter we find a regularisation value of $\tau = 0.2$, and by manual testing a feature selection set to 10 to give good results. While the SVD is only applied on the training and query images, KCCA aims to learn the correlation between the training images and their associated keywords. MMR is similar to KCCA but learns the keywords as a multi-label of the images. We use linear kernels across the methods.

6.1 Performance Measure

In order to assess the performance of the discussed methods, we present two complementing measures from the content based retrieval literature. We first consider the *normalised score* measure, as suggested by [1]. This measure gives a value of 1 if the image is annotated exactly correctly, 0 for predicting nothing or everything and a value of -1 if the exact complement of the original word set is predicted. Throughout the experimentation we multiply this measure by 100.

Let r be the number of correctly predicted keywords, n be the number of original keywords, w be the number of incorrectly predicted keywords and N the number of words in the dictionary. We are able to define the normalised score measure to be $E_{NS} = r(n)^{-1} - w(N - n)^{-1}$. The problem with the normalised score measure is that if we consider an annotation method that annotates an image exactly, then the normalised score does not sufficiently weight the incorrect guesses. This was demonstrated by [13], where they have shown that the normalised score is maximised when their annotation system returned 40 keywords per image on a test database with an average of 18.5 keywords per image. This shows that the normalised score may not account for the added noise (i.e. incorrect keywords) once all correct keywords have been selected.

We therefore choose to use the precision and recall evaluation as the main measure of the methods performances. We are able to define *recall* as $\text{Recall} = r(n)^{-1}$, and *precision* as $\text{Precision} = r(r + w)^{-1}$. We would like to have a high ratio of correctly annotated keywords to the number of keywords annotated and a high overall ratio of correct keywords (i.e. high precision and high recall).

7 Results

In the following section we present our obtained results. Throughout the presented results, best results are highlighted in bold. Initially we present the methods run-time in seconds; KCCA - 2.61, MMR - **0.19** and LSI - 57.42. We find that the vector-label learning algorithm MMR is able to solve the multi-label optimisation problem ~ 13.74 times faster than KCCA and ~ 302.2 times faster than applying the SVD procedure.

In our first task we aim to annotate a query by retrieving to it the most similar image from the training corpus. The query image is then annotated with the keywords from the found matching image. In this task we compare KCCA, MMR and LSI, we also provide an indication of how good the image annotation

approach would perform if the “matching” image would have been drawn *randomly* from the training corpus. In Tables 2 and 3 we give the normalised score measure for the methods on the testing and training set.

Table 2. Image Retrieval Results Comparison (Train Set)

Method	Precision	Recall	E_{NS}
KCCA (10) - TFIDF	68.77% ± 1.38%	80.79% ± 1.29%	79.41 ± 1.34
MMR - TFIDF	36.98% ± 5.43%	35.99% ± 2.41%	33.07 ± 2.50
LSI (40) - TFIDF	20.34% ± 5.04%	21.07% ± 5.67%	17.42 ± 5.15
KCCA (10) - FV	68.45% ± 1.56%	80.42% ± 1.60%	79.03 ± 1.67
MMR - FV	31.28% ± 1.95%	24.08% ± 2.24%	28.47 ± 2.01
LSI (40) - FV	20.67% ± 2.81%	20.64% ± 2.98%	17.67 ± 2.96

We are able to observe that all methods are able to find on average matching images that contain keywords that do not contain *everything* or *nothing* (an E_{NS} value of 0), but that KCCA with a feature selection of 10 is able to find more images with a similar keyword annotation. It is surprising to observe that LSI and Random have a similar performance level. As discussed in the previous section the normalised score measure may not be an ideal performance measure, therefore we provide in Table 2 the precision and recall performance on the training set and in Table 3 the precision and recall performance measure on the testing set. We again observe that LSI on average has a similar performance to random. Although as indicated by the large standard deviation, there are random splits of training and testing that produce a recall and precision value of $\sim 35\%$. We are assured that learning is occurring when we compare KCCA and




Table 3. Image Retrieval Results Comparison (Test Set)

Method	Precision	Recall	E_{NS}
KCCA (10) - TFIDF	37.01% ± 1.22%	45.92% ± 1.11%	42.95 ± 1.16
MMR - TFIDF	34.15% ± 5.32%	32.95% ± 1.39%	29.96 ± 1.44
LSI (40) - TFIDF	19.97% ± 5.44%	20.54% ± 5.82%	17.08 ± 5.58
KCCA (10) - FV	36.58% ± 1.37%	45.14% ± 1.46%	42.23 ± 1.46
MMR - FV	21.73% ± 1.44%	29.11% ± 1.48%	26.30 ± 1.44
LSI (40) - FV	19.31% ± 3.23%	18.93% ± 2.82%	16.30 ± 3.36
Random	19.27% ± 0.92%	19.21% ± 0.92%	16.26 ± 0.9

MMR to random. We are able to see that MMR produces twice the recall and precision and KCCA twice the performance of precision and ~ 2.5 times of recall. We find that the application of TFIDF on the ‘visual’ terms does boost results implying that increasing the weighting of SIFT descriptors that occur frequently within an image but not so in overall images, helps the learning process. We were unable to reproduce the LSI results given in [8] where it performed best.

In Table 4 we give an example of three query images and the keywords of the retrieved images from the various methods. We do not display the actual retrieved images due to lack of space.

Table 4. Image Annotation via Matching Image Retrieval

Original			
	Tree Trunk, Log Ground, Elk Greenery	Partially Cloudy Sky Tree, Water Hill	Football Field, Band Partially Cloudy Sky Track, Tree, Post
MMR	Building, Tree, Grass Leafless Tree, Bush Clear Sky, Sidewalk	Clear Sky, Tree, Bush Street, Building Car, People	Stadium, Stand, People Football Field, Band Track, Banner, Tree Post
KCCA	Tree Trunk, Log Ground, Elk Greenery	Partially Cloudy Sky Ground, Water, Hill Tree, Grass	Cloudy Sky, Bridge Water
LSI	Ocean, Building Tree, Sky	Sky, Cloud, Building Ocean	Cloud, Tree, Palace

In the second experiment we aim to predict a multi-label using the MMR and generate a new best matching document to the query using KCCA. In both methods we predict/create a new document containing the exact number of keywords as with the original query.

In Tables 5 and 6 we again provide the normalised score measure for completeness. We are able to observe that here the performance of random extremely degrades from that quoted performance in Tables 2 and 3 while that of KCCA and MMR stays similar. In Table 5 we give the performance on the training set and in Table 6 the performance on the testing set is displayed. We notice that the recall and precision values are equivalent to each other, we presume that this occurs due to the fact that for each query image we predict/create a different set of keywords (according to the number of keywords in the query image).

We observe that although we are now trying to predict/generate keywords directly from an image rather than finding a similar image and using its keywords, our results are similar across the two approaches. This similarity is not surprising as in both approaches we are learning the association between images and words and not images to images, we only change our testing criterion in each annotation procedure. We find as in the previous annotation approach that the application of TFIDF increases the methods performance.




Table 5. Keyword Generation Results Comparison (Train Set)

Method	Precision & Recall	E_{NS}
KCCA (10) - TFIDF	68.1% ± 1.28%	67.01 ± 1.29
MMR -TFIDF	37.12% ± 0.96%	34.78 ± 1.01
KCCA (10) - FV	68.50% ± 1.36%	67.42 ± 1.38
MMR - FV	28.30% ± 1.43%	25.64 ± 1.49

Table 6. Keyword Generation Results Comparison (Test Set)

Method	Precision & Recall	E_{NS}
KCCA (10)	38.16% ± 1.41%	36.06 ± 1.43
MMR	31.42% ± 1.77%	28.9 ± 1.82
KCCA (10) - FV	36.80% ± 1.36%	34.60 ± 1.38
MMR - FV	23.75% ± 2.05%	20.97 ± 2.09
Random	3.63% ± 0.37%	0.13 ± 0.28

Table 7. Keyword Generation

			
Original	Tree, Clear Sky, People Stands, Football Field Scoreboard, Stadium	Tree, Sky, Cloud Temple	Tree Trunk, Water Greenery, Elk
MMR	Stadium, Football Field Cloudy Sky, People Track, Band, Tree	Snow, Sky, Temple, Tree	Water Fall, Fields, Red Square Duck Pond
KCCA	Tree, Pole, Struct People, Overcast Sky Football Field,Car	Overcast Sky,, Tree Partially Cloudy Sky Ground	Ground, Grass, Tree Building

In Table 7 we give an example of three query images and the keywords that were predicted/generated from the various methods. While performing quite accurately on image 1 it is interesting to observe that in image 2 MMR replaced *Cloud* with *Snow*, while KCCA learnt the association of the keywords which described the surroundings of the image. The third image query shows a more complicated example due to the density of elements within it. It is visible that MMR keyword prediction, except for *Fields*, could not really describe the image although *Water Fall* and *Duck Pond* could be somewhat understood as there

is water in the image. KCCA generated an incorrect annotation of *Building* probably due to the high density of trees which could resemble the structure of a building.

We find that in both image annotation procedures KCCA and MMR perform extremely well in comparison to LSI and random, indicating that 1) learning the association of keywords to image descriptors using superior semantic models can produce good results and 2) we are able to learn the association as a multi-label task while retaining the complexity of the learning to a practical minimum. It is interesting to note that while applying TFIDF on the visual terms boosts results for both LSI and MMR, KCCA seems to stay constant in its keyword prediction and annotation performance. We believe that this shows that even without increasing the weighting of frequently occurring SIFT descriptors within an image, KCCA is able to find matching correlation between the keywords and those SIFT descriptors.

8 Conclusions

Two annotation procedures were presented; the first aiming to retrieve an image best matching a query image and the second aiming to annotate a query image directly. We have shown that the direct annotation can produce as good results as an image comparison. Although the analogy of annotating an image based on the most similar image is adequate we believe that learning the relationship between keywords and image descriptors to be a more interesting and challenging task. In our results we show that it is indeed possible to learn this association directly and still provide good results. In future work we would like to explore enhancing the annotation accuracy by combining several image descriptors[3] as well as examining a new non orthogonal representation of the keywords as labels for the MMR method. Further work on KCCA parameter selection and experimental reproduction on a larger database.

References

1. Kobus Barnard, Pinar Duygulu, David Forsyth, Nando de Fretias, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.
2. D. Blei and M. Jordan. Modeling annotated data. In *Proc. of the 26th Intl. Association for Computing Machinery Special Interest Group Information Retrieval Conference (ACM SIGIR)*, 2003.
3. Jason D. R. Farquhar, David R. Hardoon, Hongying Meng, John Shawe-Taylor, and Sandor Szedmak. Two view learning: SVM-2K, theory and practice. In *Advances of Neural Information Processing Systems 19*, 2005.
4. Colin Fyfe and Pei Ling Lai. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 2001.
5. David R. Hardoon. *Semantic Models for Machine Learning*. PhD thesis, University of Southampton, 2006.

6. David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: an overview with application to learning methods. *Neural Computation*, 16:2639–2664, 2004.
7. Jonathon S. Hare and Paul H. Lewis. On image retrieval using salient regions with vector spaces and latent semantics. In *Image and Video Retrieval: Third International Conference (CIVR)*, 2005.
8. Jonathon S. Hare and Paul H. Lewis. Saliency-based models of image content and their application to auto-annotation by semantic propagation. In *Proceedings of Multimedia and the Semantic Web / European Semantic Web Conference*, 2005.
9. D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th IEEE International Conference on Computer vision*, pages 1150–1157, Kerkyra Greece, 1999.
10. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 525–531, Hawaii USA, 2001.
11. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the 2002 European Conference on Computer vision*, pages 128–142, Copenhagen Denmark, 2002.
12. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *International Conference on Computer Vision and Pattern Recognition*, pages 257–263, 2003.
13. F. Monay and D. Gatica-Perez. On image auto-annotation with latent space models. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*. ACM Press, 2003.
14. J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Gcap: Graph-based automatic image captioning. In *Proc. of the 4th International Workshop on Multimedia Data and Document Engineering (MDDE 04), in conjunction with Computer Vision Pattern Recognition Conference (CVPR 04)*, 2004.
15. J. Rousu, C.J. Saunders, S. Szedmak, and J. Shawe-Taylor. Learning hierarchical multi-category text classification models. In *ICML*. 2005.
16. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Berlin, 1983.
17. N. Sebe, Q. Tian, E. Louprias, M. Lew, and T. Huang. Evaluation of salient point techniques. *Image and Vision Computing*, 21:1087–1095, 2003.
18. E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images using dual-wing harmoniums. In *Uncertainty in Artificial Intelligence '05*, 2005.

Fast Discovery of Time-Constrained Sequential Patterns Using Time-Indexes

Ming-Yen Lin¹, Sue-Chen Hsueh², and Chia-Wen Chang¹

¹ Department of Information Engineering and Computer Science,
Feng-Chia University, Taiwan
linmy@fcu.edu.tw, m9318119@fcu.edu.tw

² Department of Information Management,
Chaoyang University of Technology, Taiwan
schsueh@mail.cyut.edu.tw

Abstract. Sequential pattern mining is to find out all the frequent sub-sequences in a sequence database. In order to have more accurate results, constraints in addition to the support threshold need to be specified in the mining. Time-constraints cannot be managed by retrieving patterns because the support computation of patterns must validate the time attributes for every data sequence in the mining process. In this paper, we propose a memory time-indexing approach (called METISP) to discover sequential patterns with time constraints including minimum/maximum/exact gaps, sliding window, and duration. METISP scans the database into memory and constructs time-index sets for effective processing. Utilizing the index sets and the pattern-growth strategy, METISP efficiently mines the desired patterns without generating any candidate or sub-database. The comprehensive experiments show that METISP outperforms GSP and DELISP in the discovery of time-constrained sequential patterns, even with low support thresholds and very large databases.

1 Introduction

The problem of mining sequential patterns is becoming an active research topic. A typical example is to find all the frequent sub-sequences in a retail database of customer purchasing sequences. The mining results disclose not only the frequent items bought together, but also the sequences of frequently appeared item-sets.

Improving the efficiency of sequence mining algorithms has been the focus of many studies [4, 6, 8, 10] while increasing the accuracy of mining results tends to be more desirable in practice. Various constraints, such as item, length, super-pattern, duration, and gaps, can be specified to find the interesting patterns. Many constraints can be handled by a post-processing on the result of sequential pattern mining without constraints. Nevertheless, time constraints affect the support computation of patterns so that they cannot be handled without modifying the mining algorithms for the consideration of time attributes.

The issue of mining sequential patterns with time constraints was first addressed in [3]. Three time constraints including minimum gap (abbreviated as *mingap*), maximum gap (abbreviated as *maxgap*) and sliding time-window (abbreviated as

swin) are specified to enhance the semantics of sequence discovery. For example, without time constraints, one may find a pattern $\langle(\text{PC}, \text{PRINTER})(\text{LCD-PROJECTOR})\rangle$, which means most customers could buy LCD-PROJECTOR after purchasing PC and PRINTER. However, the pattern could be insignificant if the time interval between the two transactions is too long. Such patterns could be filtered out if the maximum gap constraint is specified. Similarly, the minimum gap restricts the minimum time difference between adjacent transactions and reinforces the discovery.

Essentially, a data sequence is defined to support a pattern if each element of the pattern is contained in an individual transaction of the data sequence. Sometimes users may not mind whether the items of an element are bought in the same transaction, or in adjoining transactions if the adjoining transactions occur within a specified time interval. For instance, given a sliding time-window of 4, a data sequence $\langle_5(\text{PC})_7(\text{DVD-DRIVE})_{10}(\text{DVD-R})\rangle$ can support pattern $\langle(\text{PC}, \text{DVD-DRIVE})\rangle$. Sliding time-window constraint relaxes the definition of an element and broadens the applications of sequential patterns.

In addition to the three time constraints, duration and exact gap constraints are usually specified for finding actionable patterns. Duration indicates the maximum total time-span allowed for a pattern. Users would not obtain a pattern having transactions conducted over one year if the duration of 365 days is given. Exact gap can be used to find patterns, within which adjacent transactions occur exactly the specified time difference.

Although there are many algorithms dealing with sequential pattern mining [1, 2, 5, 12], few handle the mining with the addition of time constraints [9, 11, 13]. The GSP algorithm [3] solves the constraints except duration and exact gap, within the Apriori framework. In addition, GSP must scan the database k times to discover patterns having k items. The *cSPADE* algorithm [13] extends the vertical mining algorithm *SPADE* [12] to deal with time constraints. The *cSPADE* algorithm checks *mingap* and *maxgap* while doing temporal joins. Nevertheless, the huge sets of frequent 2-sequences must be preserved to generate the required classes for the *maxgap* constraint [13]. While it is possible for *cSPADE* to handle constraints like *maxgap/mingap* by expanding the id-lists and augmenting the join-operations with temporal information, it does not appear feasible to incorporate *swin*. The *swin* constraint was not mentioned in *cSPADE*. The PrefixSpan algorithm is extended in [11] to push the prefix-monotone constraint [11] into the pattern-growth framework. However, the sliding time-window constraint is neither monotonic nor anti-monotonic. Handling sliding time-window constraint is a non-trivial task of modifying the Prefix-growth algorithm since the prefix-monotonic feature is no longer valid. The DELISP algorithm [7], fully functionally equivalent to GSP on time constraint issues, solves the problem within pattern-growth framework. However, the size of the projected databases might accumulate to several times the size of the original database.

In this paper, we propose a new algorithm called METISP (MEMory Time-Indexing for Sequential Pattern mining) for handling time constraints on sequential patterns. METISP deals with minimum/maximum/exact gap, sliding time-window and duration constraints by memory time-index sets within the pattern-growth framework [6, 7, 10, 11].

The rest of the paper is organized as follows. The problem is formulated in Section 2. Section 3 presents the METISP algorithm. The experimental evaluation is described in Section 4. Section 5 concludes our study.

2 Problem Statement

Let $\mathcal{P} = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ be a set of literals, called *items*. An *itemset* $I = (\beta_1, \beta_2, \dots, \beta_q)$ is a nonempty set of q items such that $I \subseteq \mathcal{P}$. A *sequence* s , denoted by $\langle e_1 e_2 \dots e_w \rangle$, is an ordered list of w *elements* where each *element* e_i is an itemset. Without loss of generality, we assume the items in an element are in lexicographic order. The *length* of a sequence s , written as $|s|$, is the total number of items in all the elements in s . Sequence s is a k -*sequence* if $|s| = k$. The sequence database DB contains $|DB|$ data sequences. A *data sequence* ds has a unique identifier sid and is represented by $\langle {}_{t_1}e_1' {}_{t_2}e_2' \dots {}_{t_n}e_n' \rangle$, where element e_i' occurred at time t_i , $t_1 < t_2 < \dots < t_n$.

A user gives a parameter *minsup* (**minimum support**) and four time-constraints *maxgap* (**maximum gap**), *mingap* (**minimum gap**), *swin* (**sliding window**) and *duration* to discover the set of all time-constrained sequential patterns. A sequence s is a *time-constrained sequential pattern* if $s.sup \geq minsup$, where $s.sup$ is the *support* of the sequence s and *minsup* is the user specified minimum support threshold. The *support* of s is the number of data sequences *containing* s divided by $|DB|$. A data sequence $ds = \langle {}_{t_1}e_1' {}_{t_2}e_2' \dots {}_{t_n}e_n' \rangle$ *contains* a sequence $s = \langle e_1 e_2 \dots e_w \rangle$ if there exist integers $l_1, u_1, l_2, u_2, \dots, l_w, u_w$ and $1 \leq l_1 \leq u_1 < l_2 \leq u_2 < \dots < l_w \leq u_w \leq n$ such that the five conditions hold: (1) $e_i \subseteq (e_{l_i}' \cup \dots \cup e_{u_i}')$, $1 \leq i \leq w$ (2) $t_{u_i} - t_{l_i} \leq swin$, $1 \leq i \leq w$ (3) $t_{u_i} - t_{l_{i-1}} \leq maxgap$, $2 \leq i \leq w$ (4) $t_{l_i} - t_{u_{i-1}} \geq mingap$, $2 \leq i \leq w$ (5) $t_{u_w} - t_{l_1} \leq duration$. Assume that t_j , *mingap*, *maxgap*, *swin*, and *duration* are all positive integers, $maxgap \geq mingap \geq 1$. When *mingap* is the same as *maxgap*, the time gap is called *exact gap*.

Table 1. Example sequences database (DB) and the time-constrained sequential patterns

Sid	Sequence	Time-constrained sequential pattern (<i>minsup</i> =50%, <i>mingap</i> =3, <i>maxgap</i> =15, <i>swin</i> =2, <i>duration</i> =25)
C1	$\langle {}_3(c)_5(a, f)_{18}(b)_{31}(a)_{45}(f) \rangle$	$\langle (a) \rangle, \langle (a)(b) \rangle, \langle (a)(d) \rangle, \langle (a, c) \rangle,$
C2	$\langle {}_6(a, c)_{10}(b)_{17}(e)_{18}(a)_{24}(c, d) \rangle$	$\langle (a, c)(b) \rangle, \langle (b) \rangle, \langle (b)(a) \rangle, \langle (b)(d) \rangle, \langle (b)(e) \rangle,$
C3	$\langle {}_1(b)_{20}(b, g)_{27}(e)_{34}(d, g)_{35}(g) \rangle$	$\langle (b)(e)(d) \rangle, \langle (c) \rangle, \langle (c)(b) \rangle, \langle (c)(e) \rangle, \langle (c, d) \rangle,$
C4	$\langle {}_5(a)_{10}(d)_{21}(c, d)_{26}(e) \rangle$	$\langle (d) \rangle, \langle (e) \rangle, \langle (e)(d) \rangle$

For example, given a sequence DB in Table 1, the *mingap*=3, *maxgap*=15, *swin*=2 and *duration*=25. The sequence $\langle (a, c)(b) \rangle$ is contained in data sequences C1 $\langle {}_3(c)_5(a, f)_{18}(b)_{31}(a)_{45}(f) \rangle$ because element (a, c) can be contained in the transaction combining ${}_3(c)$ and ${}_5(a, f)$ for $5-3 \leq 2$ (*swin*). Meanwhile, the other constraints can be satisfied for $18-5 \geq 3$ (*mingap*), $18-3 \leq 15$ (*maxgap*) and total time span $18-3 \leq 25$ (*duration*). Similarly, considering sequence $\langle (a)(b)(a) \rangle$, it can be contained in C1 for the other con-

straints but it fails the *duration* constraint ($31-5>25$). Given $mingap = maxgap = 7$, that is, exact gap = 7, $C3 \langle (b)_{20}(b, g)_{27}(e)_{34}(d, g)_{35}(g) \rangle$ contains pattern $\langle (b)(e)(d) \rangle$ but it does not contain pattern $\langle (b)(d) \rangle$ because $34-20 \neq 7$ (exact gap).

3 METISP: Memory Time-Indexing for Sequential Pattern Mining

In this section, we propose a new algorithm, called METISP, for time-constrained sequential pattern mining. Section 3.1 introduces the terminology used in METISP. The detail of the algorithm is described in Section 3.2. The technique for handling very large databases is illustrated in Section 3.3. For convenience, we refer to the sequence database as *DB* and a data sequence as *ds* in the following context.

3.1 Terminology Used in METISP

Definition 1. (frequent item) An item x is called a *frequent item* in *DB* if $\langle (x) \rangle_{sup} \geq minsup$.

Definition 2. (type-1 pattern, type-2 pattern, stem, prefix) Given a frequent pattern P and a frequent item x in *DB*, P' is a *type-1 pattern* if it can be formed by adding an itemset of the single item x after the last element of P , and a *type-2 pattern* if formed by appending x to the last element of P . The item x is called the *stem* of the new frequent pattern P' . The *prefix pattern* (abbreviated as *prefix*) of P' is P .

Definition 3. (initial time, last-start time, last-end time, time index) Let the first element of a frequent pattern P be *FE* and last element be *LE*. If ds contains P by having $FE \subseteq e_\delta \cup e_{\delta 1} \cup \dots \cup e_\epsilon$ and $LE \subseteq e_\gamma \cup e_{\gamma 1} \cup \dots \cup e_\omega$, where $e_\delta, \dots, e_\omega$ are elements in ds , the occurring time t_δ, t_γ , and t_ω for itemsets e_δ, e_γ and e_ω , respectively, are named *initial time* (abbreviated as **it**), *last-start time* (abbreviated as **lst**) and *last-end time* (abbreviated as **let**) of P in ds . Every occurrence of timestamp $it:st:et$ is collected altogether as $[it_1:lst_1:let_1, it_2:lst_2:let_2, \dots, it_k:lst_k:let_k]$, $it_i \leq lst_i \leq let_i$ for $1 \leq i \leq k$. Such a timestamp lists is called the *time index* of P in ds .

Definition 4. (valid time periods) Given a time index of P in ds $[it_1:lst_1:let_1, it_2:lst_2:let_2, \dots, it_k:lst_k:let_k]$, the time periods of the itemsets in ds to be used for finding a potential stem x of pattern P' , where P is the prefix of P' , are called *valid time periods* (abbreviated as *VTPs*).

Lemma 1. Given a time index of P in ds , the valid time period to form a type-1 pattern must satisfy $let_i + mingap \leq VTP \leq \min\{lst_i + maxgap, it_i + duration\}$, $1 \leq i \leq k$.

Lemma 2. Given a time index of P in ds , the valid time period to form a type-2 pattern must satisfy $let_i - swin \leq VTP \leq \min\{lst_i + swin, it_i + duration\}$, $1 \leq i \leq k$.

Note that the potential stem found using Lemma 2 can be pruned if it is lexicographically smaller than the items of the last element in P . For any type-2 pattern, the valid time periods are checked on not violating the minimum/maximum gap constraints between the last two elements. In case P has only one element and the VTP to be used is earlier than st_i , then *duration* constraint must be checked additionally.

Input: *DB* (a sequence database), *minsup* (minimum support), *mingap* (minimum gap), *maxgap* (maximum gap), *swin* (sliding time-window), *duration* (duration)

Output: the set of all time-constrained sequential patterns

1. Load *DB* into memory (as *MDB*) and scan *MDB* once to find all frequent items.
2. for each frequent item *x*,
 - (1) form the sequential pattern $P = \langle x \rangle$ and output *P*.
 - (2) scan *MDB* once to construct *P-Tidx*, time index set of *x*.
 - (3) call *MineType1*(*P*, *P-Tidx*)
 - (4) call *MineType2*(*P*, *P-Tidx*)

Subroutine: MineType1(P, P-Tidx)

Parameter: *P* = prefix pattern, *P-Tidx* = time index set

1. for each data sequence *ds* in the *P-DB*, // *P-DB*: sequences indicated in *P-Tidx*
 - (1) use the corresponding time index to collect the type-1 VTPs satisfying

$$let_i + mingap \leq VTP \leq \min\{lst_i + maxgap, it_i + duration\}, 1 \leq i \leq k$$
 - (2) for each item in the VTPs, add one to its support count.
2. for each item *x'* that has support greater than or equal to *minsup*,
 - (1) form the type-1 pattern *P'* by extending stem *x'* and output *P'*.
 - (2) scan the VTP of each *ds* in *P-DB* to construct *P'-Tidx*, time index set of *x'*.
 - (3) call *MineType1*(*P*, *P'-Tidx*).
 - (4) call *MineType2*(*P'*, *P'-Tidx*).

Subroutine: MineType2(P, P-Tidx)

Parameter: *P* = prefix pattern, *P-Tidx* = time index set

1. for each data sequence *ds* in the *P-DB*, // *P-DB*: sequences indicated in *P-Tidx*
 - (1) use the corresponding time index to collect the type-2 VTPs satisfying

$$let_i - swin \leq VTP \leq \min\{lst_i + swin, it_i + duration\}, 1 \leq i \leq k$$
 - (2) for each item in the VTPs, add one to its support count.
2. for each item *x'* that has support greater than or equal to *minsup*,
 - (1) form the type-2 pattern *P'* by appending stem *x'* and output *P'*.
 - (2) scan the VTP of each *ds* in *P-DB* to construct *P'-Tidx*, time index set of *x'*.
 - (3) call *MineType1*(*P*, *P'-Tidx*).
 - (4) call *MineType2*(*P'*, *P'-Tidx*).

Fig. 1. Algorithm METISP

3.2 METISP Algorithm

Fig. 1 outlines the METISP Algorithm. METISP mines patterns within the pattern-growth framework similar to the pseudo projection version [10] of Prefixspan algorithm, while it can handle constraints minimum/maximum gaps, duration and sliding time-window. Assume that the *DB* can fit into the main memory; METISP first loads *DB* into memory (as *MDB*) and scans *MDB* once to find all frequent items. With respect to each frequent item, METISP then constructs a time index-set for the 1-sequence item and recursively forms time-constrained sequential patterns of longer length. The time index-set is a set of (data-sequence pointer, time index) pairs. Only those data sequences containing that item would be included.

In Fig. 1, *MineType1*(*P*, *P-Tidx*) mines type-1 patterns having prefix *P* by effectively locating VTPs using Lemma 1. Likewise, *MineType2*(*P*, *P-Tidx*) discovers

type-2 patterns having prefix P by fast locating VTPs using Lemma 2. The P -TIdx is the time index-set for P . METISP therefore needs not search the data sequences irrelevant to P . Moreover, the VTPs ensure that METISP locates and counts the effective stems which can form valid patterns, instead of the whole set of items in the data sequence. By pushing time attributes deep into the mining process, METISP efficiently discovers the desired patterns.

3.3 Handling Very Large Databases by Partition-and-Validation Technique

Still, some databases might be too large for the main memory to accommodate in a batch. In this case, the time-constrained sequential patterns are discovered by a partition-and-validation technique. The DB is partitioned so that each partition can be handled in main memory by METISP. The number of partitions is minimized by reading as many data sequences into main memory as possible to constitute a partition. The set of potential patterns in DB is obtained by collecting the discovered patterns after running METISP on these partitions. The true patterns can be identified with only one extra database pass through support counting against all the data sequences in DB one at a time. Therefore, we may employ METISP to mine databases of any size, of any minimum support, in just two passes of database scanning.

4 Experimental Results

Extensive experiments were performed to assess the performance of the METISP algorithm. Time-constrained mining algorithms including GSP [3] and DELISP [7] were compared with METISP, using the IBM synthetic dataset generation program [2]. Here, we describe the result of dataset C10-T2.5-S4-I1.25 having 100000 data sequences ($|DB| = 100k$), with $N_s=5000$, $N_l=25000$ and $N=1000$. The detailed parameters are addressed in [3]. The results of varying $|C|$, $|T|$, $|S|$, and $|I|$ were consistent. All the experiments were performed on AMD 2400+ PC with 1GB memory running the Windows XP.

The total execution times of the three algorithms on minimum support, *mingap*, *maxgap*, and *swin* were compared. Fig. 2 shows that, on average, the total execution times of GSP and DELISP are about 5.6 and 3.9, respectively, times of METISP. We show the results of varying *mingap* in Fig. 3, varying *maxgap* in Fig. 4, and varying *swin* in Fig. 5. No other constraints were set in the experiments so as to observe the effect of the single constraint. METISP outperforms GSP and DELISP for all these experiments. In fact, METISP constantly mines the patterns about 55 seconds, for the distinct experiments. The results of changing duration and exact-gap values are shown in Figs 6 and 7, respectively. We depict only the results of METISP because GSP and DELISP cannot handle the constraints.

The results of scaling up the database sizes are depicted in Fig. 8 (a) and Fig. 8 (b). Dealing with the mining of $|DB| = 2000K$ data sequences invoked the partition-and-validation technique and the database was partitioned into two segments. The result of scaling up databases indicated that METISP has good linear scalability.

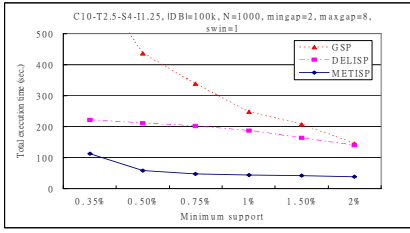


Fig. 2. Effects of varying minimum support

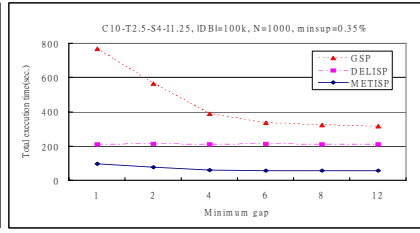


Fig. 3. Effects of varying minimum gap

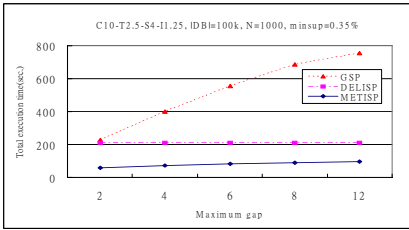


Fig. 4. Effects of varying maximum gap

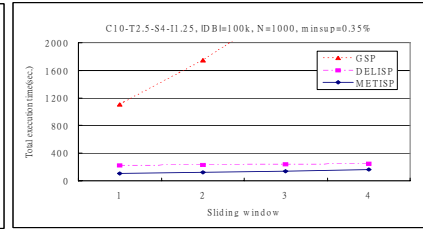


Fig. 5. Effects of varying swin

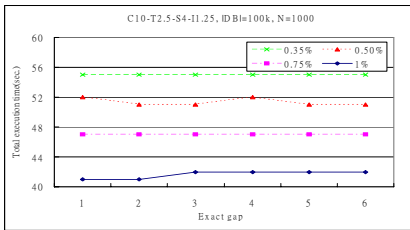


Fig. 6. Effects of varying exact gap

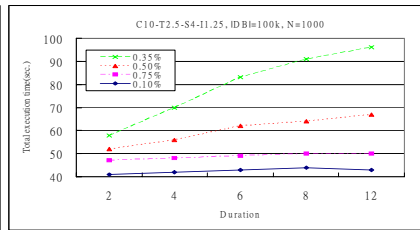
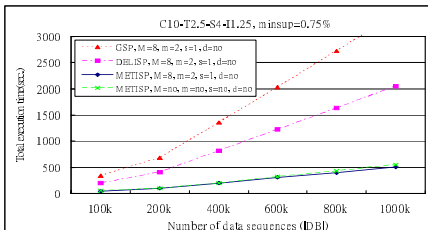
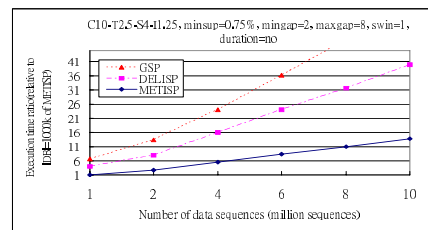


Fig. 7. Effects of varying duration



(a) IDB: 100k~1000k



(b) IDB: 1000k~10000k

Fig. 8. Linear scalability of the database size

5 Conclusion

We have presented METISP, an efficient memory time-indexing algorithm for mining sequential patterns with various time constraints, including minimum/maximum/exact gaps, sliding time-window, and duration. Using the pattern-growth approach together with the new memory time-index sets, METISP effectively shrinks the search space of potential patterns. Considering that some extra-large databases still might not fit into the main memory, we use the partition-and-validation technique to mine the very large database. The comprehensive experiments show that METISP is very efficient and outperforms both GSP and DELISP algorithms.

Acknowledgements

The authors thank the reviewers' precious comments. This research was supported by National Science Council of Taiwan under grant number NSC-94-2213-E-035-012.

References

1. Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. Sequential PATTERN Mining using A Bitmap Representation. Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining, 2002, 429-435.
2. Agrawal, R., and Srikant, R. Mining Sequential Patterns. Proceedings of the 11th International Conference on Data Engineering, Taipei, Taiwan, 1995, 3-14.
3. Agrawal, R., and Srikant, R. Mining Sequential Patterns: Generalizations and Performance Improvements. Proceedings of the 5th International Conference on Extending Database Technology, Avignon, France, 1996, 3-17.
4. Chiu, D. Y., Wu, Y. H., and Chen, A. L. P. An Efficient Algorithm for Mining Frequent Sequences by a New Strategy without Support Counting. Proceedings of the 20th International Conference on Data Engineering, 2004, 375-386.
5. Garofalakis, M. N., Rastogi, R., and Shim, K. SPIRIT: Sequential Pattern Mining with Regular Expression Constraints. Proceedings of the 25th International Conference on Very Large Data Bases, Edinburgh, Scotland, Sep. 1999, 223-234.
6. Lin, M. Y., and Lee, S. Y. Fast Discovery of Sequential Patterns through Memory Indexing and Database Partitioning. Journal of Information Science and Engineering. Volume 21 No. 1, Jan. 2005, 109-128.
7. Lin, M. Y., and Lee, S. Y. Efficient Mining of Sequential Patterns with Time Constraints by Delimited Pattern-Growth. Knowledge and Information Systems. Volume 7, Issue 4, May 2005, 499-514.
8. Massegli, F., Cathala, F., and Poncelet, P. The PSP Approach for Mining Sequential Patterns. Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery, Nantes, France, 1998, 176-184.
9. Orlando, S., Perego, R., and Silvestri, C. A new algorithm for gap constrained sequence mining. Proceedings of the 2004 ACM Symposium on Applied Computing, 2004, 540-547.
10. Pei, J., Han, J., Moryazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. Prefix-Span: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. Proceedings of the 17th International Conference on Data Engineering, Heidelberg, Germany, April 2001, 215-224.

11. Pei, J., Han, J., and Wang, W. Mining sequential patterns with constraints in large databases. Proceedings of the Eleventh International Conference on Information and Knowledge Management, 2002, 18-25.
12. Zaki, M. J. SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning Journal, Volume 42, Jan.-Feb. 2001, 31-60.
13. Zaki, M. J. Sequence Mining in Categorical Domains: Incorporating Constraints. Proceedings of the 9th International Conference on Information and Knowledge Management, Washington, DC, Nov. 2000, 422-429.

Multi-dimensional Sequential Pattern Mining Based on Concept Lattice*

Yang Jin and Wanli Zuo

College of Computer Science & Technology, JiLin University, ChangChun, P.R. China
mail_jy@sina.com, wanli@jlu.edu.cn

Abstract. Multi-dimensional sequential pattern mining attempts to find much more informative frequent patterns suitable for immediate use. In this paper, a novel data model called multi-dimensional concept lattice is proposed and, based on which, a new incremental multi-dimensional sequential pattern mining algorithm is developed. The proposed algorithm integrates sequential pattern mining and association pattern mining with a uniform data structure and makes the mining process more efficient. The performance of the proposed approach is evaluated on both synthetic and real-life financial date sets.

1 Introduction

Sequential pattern mining, aiming at finding frequently occurring subsequences as patterns arranged in time order, is now an important data mining problem with broad applications, including the analyses of customer purchase behavior, web access patterns, medical diagnosis, and so on, but traditional sequential pattern [1] is not informative enough. For example, a doctor who is confronted with symptoms that could relate to either one of two or more different ailments, would need to know more about the patient's age and medical history to make an accurate diagnosis. Thus it is necessary to introduce additional background information to enrich a sequence pattern, which is just what multi-dimensional sequential pattern mining cares for.

Multi-dimensional sequential pattern mining was introduced by Jiawei Han etc.in [2]. An example pattern is "A student aged 25 who purchased an IBM computer is likely to buy a HP printer shortly after". In this case, ((age=25) and (occupation="student")) tell about the background information of pattern XY.

Our contribution lies in the following four facets: (1) proposes the general definition of multi-dimensional sequential pattern mining; (2) applies concept lattice theory to mine multi-dimensional sequential pattern; (3) defines multi-dimensional concept lattice model and integrates sequential pattern mining and association pattern mining based on it; (4) being derived from concept lattice, the TDCL-SB algorithm presented in this paper is inherently incremental.

* This work was sponsored by Natural Science Foundation of China (NSFC) under Grant No. 60373099.

The remaining of the paper is organized as follows. In section 2, we review related work. Section 3 is devoted to the definition of multi-dimensional concept lattice model, with in-depth study on two-dimensional concept lattice with single base dimension. Section 4 develops TDCL-SB algorithm. An extensive performance study comparing our algorithm with related work is reported in section 5. We draw our conclusions and discuss future work in section 6.

2 Related Work

Multi-dimensional sequential pattern mining is derived from the theory of sequential pattern mining and association rule mining. For example, the PSFP algorithm [2] integrates sequential pattern mining algorithm PrefixSpan [3] with association rule mining algorithm FP-growth [4]. In this paper, we take a different approach by proposing multi-dimensional concept lattice model and the algorithm TDCL-SB is devised on the theory of concept lattice or Galois lattice [5] and ordered concept lattice [6]. This section will discuss multi-dimensional sequential pattern mining and the related work PSFP briefly.

2.1 Multi-dimensional Sequential Pattern Mining

In [2], multi-dimensional sequential pattern mining is defined as *the process of mining one or more unordered dimensions of information alongside a single ordered dimension of information*. However, a multi-dimensional sequential pattern may involve two or more ordered dimensions on some occasions. And the above definition obviously can not cover such conditions. We propose a more generalized definition.

Definition 1. Multi-dimensional sequential pattern mining is *the process of mining one or more unordered dimensions of information alongside one or more ordered dimensions of information*. The multi-dimensional sequence dataset is of schema $(RID, D_1, \dots, D_m, D_{m+1}, \dots, D_n)$, where RID is a primary key, D_1, \dots, D_m are the ordered dimensions and D_{m+1}, \dots, D_n are the unordered dimensions.

To our knowledge, [2] is the only published work concerning multi-dimensional sequential pattern mining problem. PSFP attempts to mine sequential patterns from the base dimension using the sequential pattern algorithm PrefixSpan, alongside using the FP-growth algorithm to determine the association patterns obtained from the unordered dimensions. But the mining process is not based on an identical data structure because different models are employed by PrefixSpan and FP-growth. Moreover, PrefixSpan, the critical routine of PSFP, divides the target dataset into many subsets according to prefix projections in order to reduce the average sequence length. However, the space cost of those projected databases is relatively higher, especially when the main memory is low, the space cost becomes the bottleneck.

Table 2.1. A Multi-dimensional Sequence Dataset Tm(let minsup=2)

Custom ID	Transaction dimension D1	Task-relevant dimensions		
		D2	D3	D4
1	<(bd)cb(ac)>	1	5	8
2	<(bf)(ce)b(fg)>	2	6	7
3	<(ah)(bf)abf>	1	6	8
4	<(be)(ce)d>	3	4	9
5	<a(bd)bc(bade)>	2	6	8

Tm in table 2.1 is an example multi-dimensional sequence dataset presented in [2]. Results of PSFP are not shown here and you can refer to [2] for details.

3 Multi-dimensional Concept Lattice

3.1 Multi-dimensional Concept Lattice

Multi-dimensional concept lattice is a novel data model, which can be considered as a generalization from concept lattice [5] and ordered concept lattice [6]. This section defines the basic terminology of multi-dimensional concept lattice.

Definition 2. Given a formal context $T=(O,D_1,R_1,D_2,R_2,\dots,D_m,R_m,\dots,D_n,R_n)$, where O is a finite set of objects, D_1,\dots,D_m are m sets of sequences, D_{m+1},\dots,D_n are $(n-m)$ sets of attributes and $R_k(k=1,\dots,n)$ is a binary relation between O and D_k , then there is a unique n -dimensional concept lattice L with m base dimensions and $(n-m)$ background or additional dimensions corresponding to this given context.

A concept in the n -dimensional concept lattice L is of schema $(Y, X_1,\dots,X_m,\dots,X_n)$, where Y is a set of objects called *the extension of the concept*, $X_i(i=1,\dots,m)\in\rho(D_i)$ is a set of common sequential patterns shared by all objects in Y on D_i , which is called the *task-relevant intension* or *base intension*, $X_j(j=m+1,\dots,n)\in\rho(D_j)$ is a set of common association patterns shared by objects in Y on D_j which is called *background intension* or *additional intension*.

Particularly, when $m=1$ the n -dimensional concept lattice is called n -dimensional concept lattice with single base; when $m=0$ and $n=1$ the n -dimensional concept lattice is just the *traditional concept lattice* or *Galois lattice* [5]; and when $m=1$ and $n=1$ the n -dimensional concept lattice is *ordered concept lattice* [6]. Therefore the n -dimensional concept lattice is an extension or generalization of the concept lattice and the ordered concept lattice.

Definition 3. The basic operators of n -dimensional concept lattice are given by the following definitions:

$$x \subseteq y \Leftrightarrow \begin{cases} x \text{ is sub-sequence of } y & \text{if } x,y \in \rho(D_i) \quad i=1 \dots m \\ x \text{ is subset of } y & \text{if } x,y \in \rho(D_j) \quad j=m+1 \dots n \end{cases} \tag{1}$$

$$x \cap y \Leftrightarrow \begin{cases} \text{list of common sub-sequence of } x \text{ and } y & \text{if } x,y \in \rho(D_i) \\ x \cap y & \text{if } x,y \in \rho(D_j) \quad j=m+1 \dots n \end{cases} \tag{2}$$

Note: ‘ \subseteq ’ and ‘ \cap ’ are valid only when the two operands are homogenous, namely they are either base dimensions or additional dimensions.

For an n -dimensional concept lattice, its maximum unit is denoted by $(\{\}, \rho(D_1), \dots, \rho(D_m), \dots, \rho(D_n))$, the minimum unit is denoted by $(\rho(O), \Lambda, \dots, \Lambda, \phi, \dots, \phi)$, where ‘ Λ ’ is a meta-symbol representing a sequence with no element, ‘ ϕ ’ represents an empty attribute set, and ‘ $_$ ’ represents $\rho(D_i), i=1 \dots n$.

Definition 4. Let L be a n -dimensional concept lattice with m base dimensions and $H_1=(Y_1, X_{11}, \dots, X_{m1}, \dots, X_{n1}), H_2=(Y_2, X_{12}, \dots, X_{m2}, \dots, X_{n2})$ are two concepts on L , then the order relation on L is defined by $H_1 < H_2 \Leftrightarrow \exists j X_{j1} \subset X_{j2} \forall i X_{i1} \subseteq X_{i2} \quad i, j \in [1, n] \text{ and } i \neq j$.

3.2 Two-Dimensional Concept Lattice with Single Base

Here we take a close look at two-dimensional concept lattice with single base in some depth because it is used in our proposed algorithm in the next chapter.

Our work is based on the four assumptions under which the multi-dimensional sequential pattern mining task is applicable described in [2]. The direct multi-dimensional concept lattice to deal with the task is the n -dimensional concept lattice with single base. Let its context take the form of $(O, D_1, R_1, D_2, R_2, \dots, D_n, R_n), R_2, \dots, R_n$ can be considered as the same relation R conforming to definition 5, thus the above context is changed into $(O, D_1, R, D_2, R, \dots, D_n, R)$. Moreover, let $D = D_2 \times \dots \times D_n$, we can transform $(O, D_1, R_1, D_2, R, \dots, D_n, R)$ equally to (O, D_1, R_1, D, R) according to the last two assumptions, which is the context of two-dimensional concept lattice with single base.

Example 1. The process of equivalent transformation from four-dimensional lattice with single base to two-dimensional lattice is illustrated in Fig. 3.1.

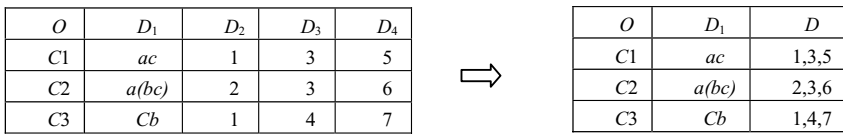


Fig. 3.1. the process of dimension compression

4 TDCL-SB Algorithm

Algorithm TDCL-SB: Two-Dimensional Concept Lattice with Single Base algorithm.

Input: Two-Dimension Concept Lattice L , sequence to be added $Z(= (Z_o, Z_u))$, minimal support threshold t , pointer array P_o, P_u whose elements point to the highest level nodes

of each unique character in the base dimension and the additional dimension respectively.

Output: Updated lattice L , pointer array P_o, P_u , frequent multi-dimensional sequence set S .

```

1.selectset←SelectNodes( $L, P_o, Z_o$ ) ∩ SelectNodes( $L, P_u, Z_u$ );
//Find all valid nodes which have intersection with Z
2.For each  $H \in$  selectset in ascending length( $X_1(H)$ ) order Do
3.      If ( $X_1(H) \subseteq Z_o$  and  $X_2(H) \subseteq Z_u$ ) Then // If  $H$  is the
         ancestor of  $Z$ , update the cardinality of  $H$ 
4.          Add  $C(H)$  by 1;
5.          If  $C(H) \geq t$  Then
6.              If  $\exists H' \in S$  such that ( $X_1(H') = X_1(H)$  and
 $X_2(H') = X_2(H)$ ) Then //  $H$  is not in the frequent sequence set
 $S$ 
7.                   $S \leftarrow S \cup \{H\}$  ;
8.                  If ( $X_1(H) = Z_o$  and  $X_2(H) = Z_u$ ) Then Exit For
;
9.      Else // Otherwise, create a new node into  $L$ 
10.         If  $X_1(H) = Z_o$  Then
11.             interset ←  $X_2(H) \cap Z_u$ ;
12.             Create new node  $N = (C(H)+1, X_1(H), \text{interiset})$ ;
13.             If  $C(H)+1 \geq t$  Then  $S \leftarrow S \cup \{N\}$ ;
14.             FindParent( $H, N$ ); // Call subroutine
15.         Else
16.             interset ←  $X_2(H) \cap Z_u$ ; // Compute
intersection of additional dimensions
17.             common ←  $X_1(H) \cap Z_o$ ; // Compute intersection
of base dimensions
18.             Do
19.                 element ← common.get(); // Get an element
20.                 IF  $\exists H_k \in \text{mark}$  such that ( $X_1(H_k) = \text{element}$  and
 $X_2(H_k) = \text{interest}$ ) Then
21.                     Create  $N = (C(H)+1, \text{element}, \text{interiset})$ ;
22.                     If  $C(H)+1 \geq t$  Then  $S \leftarrow S \cup \{N\}$ ;
23.                     FindParent( $H, N$ );
24.                 While common ≠  $\emptyset$ 
25.                 For each distinct  $x_1 \in X_1(Z)$  Do
26.                     Update  $P_o(x)$  according to updated Lattice  $L$ 
27.                 For each distinct  $x_2 \in X_2(Z)$  Do
28.                     Update  $P_u(x)$  according to updated Lattice  $L$ 

```

Subroutine FindParent(H, N) is used to find the father node of N from node H and we do not list its code for the paper length limitation.

For the multi-dimensional sequence dataset Tm shown in table 2.1 with the minimum support set to 2, the procedure of TDCL-SB is described as Fig. 4.1.

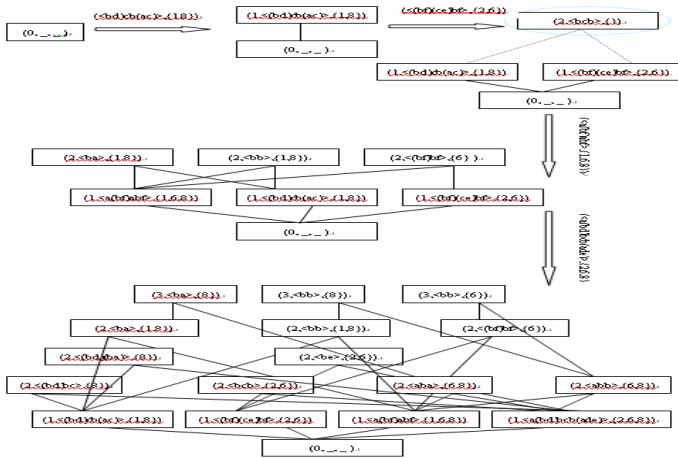


Fig. 4.1. Demonstration of the procedure of the algorithm TDCL-SB

5 Experimental Results and Performance Study

Our experiments are conducted on a 2G Hz Pentium IV PC with 256 MB main memory. As the source code of PSFP is not available, we implement the algorithm according to the description in [2]. In implementing TDCL-SB, we adopt the data structure provided in [6].

Our performance study shows that TDCL-SB is both scalable and efficient, outperforming PSFP algorithm in many aspects.

5.1 Experimental Results on the Synthetic Datasets

We use the standard program provided by IBM Almaden Research Centre to generate synthetic datasets. The parameters of the procedure are described in [7].

In the experiments, we set $N_S=5000$, $N_I=25000$, $N=1000$, $|C|=20$, $|T|=2.5$ and $|I|=1.25$. We test the scale-up performance by changing the number of customers $|D|$ from 1000 to 10000 with step 1000. We also compare the dependence of the two algorithms on the sequence length by changing the average length of maximum potential large sequences $|S|$ from 4 to 8. Therefore, we get 20 synthetic datasets and formulate them according to a uniform rule. For example, C20-T2.5-S4-I1.25-D1 implies $|D|=1000$, $|C|=20$, $|T|=2.5$, $|I|=1.25$, $|S|=4$.

After generating the synthetic datasets, we add several background or additional dimension values in front of each sequence record according to [2]. In our experiments, six compositions, 2_2, 3_3, 4_4, 5_5, 8_8 and 10_10 are used.

Figure 5.1 shows the scalability of the two algorithms over the number of customers. As the number of customers increases, the runtimes of the two algorithms scale up and TDCL-SB is more scalable.

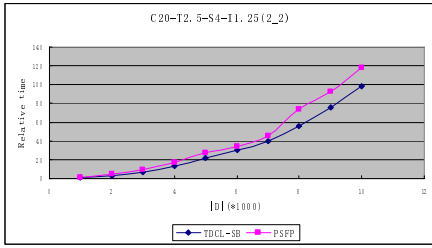


Fig. 5.1. Effect of customer size on runtime when |S| is set to 4

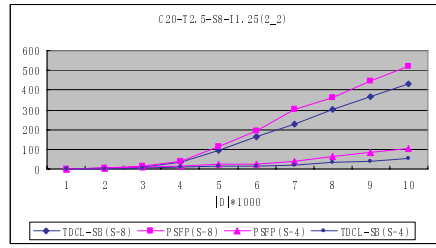


Fig. 5.2. Effect of the average length of maximum sequence on runtime

Figure 5.2 shows the scalability of the two algorithms over the average length of sequences |S|. As |S| increases, the runtimes of the two algorithms go up. Moreover, it can be observed that the increase of |S| from 4 to 8 has more impact on PSFP than TDCL-SB.

Figure 5.3 shows the influence of the different cardinalities of the background dimension and compositions of attribute values. We can see when the value compositions are relatively low, for example 2_2, 3_3 and 4_4, the performance of the two algorithms are close. However, when the value compositions are higher than 5_5, TDCL-SB outperforms PSFP significantly.

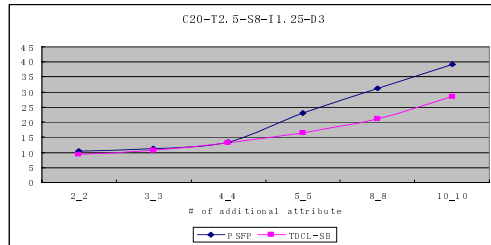


Fig. 5.3. Effect of value compositions of background dimensions on runtime

5.2 Application of TDCL-SB on Real-World Financial Data

In order to test the practicability of TDCL-SB, we use the transaction dataset of credit card customers of a commercial bank of China as the test dataset. In our experiments, we segment transactions according to the following rules: i) regard all transactions happening in ten days as one transaction; ii) take all transactions happening in a year as one sequence. We consider that a typical customer has four kinds of behaviors, i.e. depositing, drawing, over drafting and consuming. And two kinds of behaviors can happen simultaneously. Moreover, we select age, occupation, region and the card type including credit card and debit card as background dimensions. Some interesting patterns by running TDCL-SB on the above real-world dataset are obtained:

(1) Customers with credit cards whose ages are between 25 and 50 follow ‘(consuming overdraft)->depositing->overdraft’ pattern. Through investigation, we find the bank grants 30-50 days to its credit card customers as grace period and this pattern demonstrates the maturity of customers whose ages are between 25 and 50.

(2) ‘Depositing->drawing->depositing’ pattern is discovered in Jiangsu and Zhejiang Province in China. Through investigation, we find the cards following such a pattern are mostly business cards using by small or moderate corporations as salary and bonus accounts. This pattern suggests us to include business card and individual card as a new background dimension.

Moreover, the transaction datasets of card customers update very frequently to the extent that a number of new cards are granted to new customers every day. Taking the great size of the card customers, it is difficult, even for experienced expert, to set suitable min support threshold in advance. Inherited from concept lattice, TDCL-SB behaves incrementally toward the introduction of new records and variation of the min support, which means that it is possible to update the multi-dimensional concept lattice based on the structure formed by the last operation.

6 Conclusions and Future Work

In this paper, we introduce a more general concept about multi-dimensional sequential patterns mining and propose a new data model named multi-dimensional concept lattice. Then we place the emphasis on the two-dimensional concept lattice with single base and develop a new efficient algorithm named TDCL-SB. Our experiments demonstrate that TDCL-SB is both scalable and efficient. Its advantages over PSFP can be concluded in three respects: i) It employs a uniform model to mine sequential patterns and association patterns simultaneously; ii) The incremental property makes it more suitable for occasions where the target dataset accumulates over time or the end user is not sure about the min support; iii) It finds the most valuable maximum frequent multi-dimensional sequential patterns more directly.

We will further explore the application domain of our proposed n -dimensional concept lattice in two directions: i) Using n -dimensional concept lattice when there are two or more base dimensions involved; ii) Extending the technique to taxonomies described in [7] and generalizing the intension of result patterns.

References

1. R. Agrawal and R. Srikant. Mining sequential patterns. In Proc. 1995 Int.Conf. Data Engineering (ICDE'95), pages 3-14, Taipei, Taiwan, Mar.1995.
2. Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, Umeshwar Dayal: Multi-Dimensional Sequential Pattern Mining. CIKM 2001: 81-88.
3. 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 Proc. 2001 Int. Conf. Data Engineering (ICDE'01), pages 215-224, Heidelberg, Germany, April 2001.

4. J. Han, J. Pei and Y. Yin. Mining Frequent Patterns without Candidate Generation. In Proc. 2000 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'00), pages 1-12, Dallas, TX, May 2000.
5. Wille, R. Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In: Rival I ed, Ordered Set, 1982. 445-470.
6. Yang Jin, Wanli Zuo. Ordered Concept Lattice and WWW User Transversal Pattern Mining. Journal of Computer Research and Development, (in Chinese), 2003, 40(5).675-683.
7. R.Srikant, R.Agrawal. Mining Sequential Patterns: Generalizations and Performance Improvements. In: Proc. of the 5th International Conf on Extending Database Technology, (EDBT'96). Avignon, France, LNCS 1057, Springer, 1996. 3-17.
8. Dean van der Merwe, Sergei A. Obiedkov, Derrick G. Kourie. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004, Sydney, Australia, February 23-26, 2004,372-385.

Mining Time-Delayed Coherent Patterns in Time Series Gene Expression Data*

Linjun Yin, Guoren Wang, Keming Mao, and Yuhai Zhao

Northeastern University, China
wanggr@mail.neu.edu.cn

Abstract. Unlike previous pattern-based biclustering methods that focus on grouping objects on the same subset of dimensions, in this paper, we propose a novel model of coherent cluster for time series gene expression data, namely td-cluster (time-delayed cluster). Under this model, objects can be coherent on different subsets of dimensions if these objects follow a certain time-delayed relationship. Such a cluster can discover the cycle time of gene expression, which is essential in revealing the gene regulatory networks. This work is missed by previous research. A novel algorithm is also presented and implemented to mine all the significant td-clusters. Experimental results from both real and synthetic microarray datasets prove its effectiveness and efficiency.

1 Introduction

With the rapid advances of microarray technologies, large amounts of high-dimensional gene expression data are being generated, which in turn poses great challenges for existing computation. Clustering is one of the most important techniques as similar expression profiles imply a related function and indicate the same cellular pathway [11].

Traditional clustering algorithms work in the full dimensional space, which consider the value of each point in all the dimensions and try to group the similar points together. [8,9,10] are examples of full space clustering algorithms. Biclustering [4], however, does not have such a strict requirement. If some points are similar in several dimensions (a subspace), they will be clustered together in that subspace. This is very useful, especially for clustering in a high dimensional space where often only some dimensions are meaningful for some subset of points.

As a step forward, pattern-based biclustering algorithms [12,15,18] take into consideration the fact that genes with strong correlation do not have to be spatially close in correlated subspace. However, the existing pattern-based biclustering algorithms are only able to address pure shifting, scaling or inverting patterns, i.e., address shifting, scaling or inverting patterns on the same conditions. That is, after a single shifting, scaling or inverting, a pattern may coincide with another pattern. Fig. 1 shows the case of scaling patterns. In the figure, the four patterns are of the relationship: $P1 = P3 * 0.8 = P2 * 4/3 = P4 * 0.5$.

* Supported by National Natural Science Foundation of China under grant 60573089 and 60473074.

The previous patterns mentioned above ignore an additional relationship implicit in time series gene expression data. A gene may control or activate another gene downstream in a pathway; in this case, their expression profiles may be staggered, indicating a time-delayed response in the transcription of the second gene [17]. Since some genes act as regulators of other genes, by looking at the temporal expression patterns of genes we might be able to infer relationships between regulators and the genes they regulate, and explain how genes are regulated in the cell [2]. Moreover, we may find the source of a disease by identifying the gene which is the first promoter. We use an example to illustrate the patterns resulting from this relationship. Table 1(a) shows the running dataset that we will look at in this paper. Each row of the table corresponds to a gene (denoted as g_i) while each column corresponds to a certain experimental time (denoted as t_j) at which the gene expression profile (denoted as c_{ij}) is measured. For clarity certain cells have been left blank in the table, we assume that these are filled by some random expression values. The expression profiles of genes g_5, g_1, g_4 and g_9 of the running dataset in Fig. 2 are time-delayed scaling patterns: $c_{5j} = c_{1(j+1)} * 0.8 = c_{4(j+1)} * 4/3 = c_{9(j+3)} * 0.5$, where $t_j \in \{t_1, t_2, t_4\}$. In this paper, we are interested in mining this kind of patterns, which have received little attention so far. Current pattern-based models only validate the case when genes are coherent on the same subset of times i.e. the delayed time is equal to 0, which are just special cases of the time-delayed model.

Table 1. (a) Running dataset. (b) Some clusters.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
g_1		1.5	3.0		1.0		
g_2	2.0	2.0		2.0			
g_3			1.8	1.8		1.8	
g_4		0.9	1.8		0.6		
g_5	1.2	2.4		0.8			
g_6			2.5	2.5		2.5	
g_7		3.0	3.0		3.0		
g_8	1.5	1.5		1.5			
g_9				2.4	4.8		1.6

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
g_5	1.2	2.4		0.8			
g_1		1.5	3.0		1.0		
g_4		0.9	1.8		0.6		
g_9				2.4	4.8		1.6
g_2	2.0	2.0		2.0			
g_8	1.5	1.5		1.5			
g_7		3.0	3.0		3.0		
g_3			1.8	1.8		1.8	
g_6			2.5	2.5		2.5	

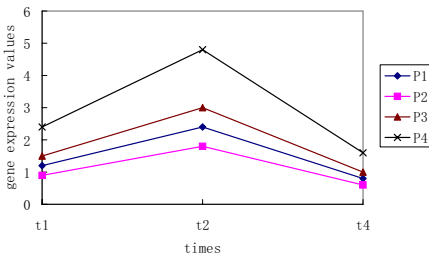


Fig. 1. Scaling patterns

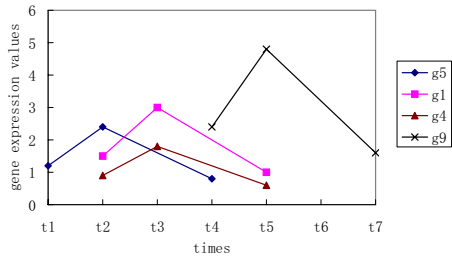


Fig. 2. Time-delayed scaling patterns

2 The Td-Cluster Model

This section describes the td-cluster model for mining time-delayed co-regulation patterns.

2.1 Definitions and Problem Statement

Let $G = \{g_1, g_2, \dots, g_n\}$ be a set of n genes and let $T = \{t_1, t_2, \dots, t_m\}$ be a set of m experimental time points with uniform time intervals. A two dimensional gene time-series microarray dataset is a real-valued $n \times m$ matrix $D = G \times T = \{c_{ij}\}$ ($i \in [1, n], j \in [1, m]$), whose two dimensions correspond to genes and times respectively. Each cell c_{ij} records the (absolute or relative) expression level of gene g_i at time t_j .

Let $T_s = \{t_{s_1}, t_{s_2}, \dots, t_{s_l}\}$ be a subset of T where $1 \leq s_1, s_2, \dots, s_l \leq m$. We call T_s a time sequence iff we have $s_1 < s_2 < \dots < s_l$ (i.e., $t_{s_1} < t_{s_2} < \dots < t_{s_l}$) and the length of T_s is l . Let $T_s = \{t_{s_1}, t_{s_2}, \dots, t_{s_l}\}$, $T_r = \{t_{r_1}, t_{r_2}, \dots, t_{r_l}\}$ be two time sequences with the same length l , and $t_{s_1} \leq t_{r_1}$. We say that there is a time-delayed relationship between T_s and T_r iff we have $r_k = s_k + d$ for all $k \in [1, l]$ where d is a constant, and T_r is delayed for d time intervals compared with T_s . Note that T_r is the same as T_s when d is equal to 0, then for unity we also view T_s and T_r as two time sequences with time-delayed relationship.

Definition 1. *Td-Cluster.* Let $C = \bigcup_{s=1}^u G_s \times T_s = \{c_{ij}\}$, where G_s is a subset of genes ($G_s \subseteq G$), and T_s is a subset of time points ($T_s \subseteq T$) and further a time sequence, then $G_s \times T_s$ specifies a submatrix of $D = G \times T$. C is a td-cluster if and only if: (1) $\forall T_s, T_r, 1 \leq s, r \leq u$, there is a time-delayed relationship between T_s and T_r , and (2) Let $G_s \times T_s$ and $G_r \times T_r$ be any two submatrices of D , where $1 \leq s, r \leq u$, we suppose T_r is delayed for d time intervals compared with T_s , $\forall g_a \in G_s, \forall g_b \in G_r, \forall t_p, t_q \in T_s$, let $r_a = \frac{c_{ap}}{c_{aq}}$ be the ratio of the expression level of gene g_a at time t_p and t_q , and let $r_b = \frac{c_{b(p+d)}}{c_{b(q+d)}}$ be the ratio of the expression level of gene g_b at time $t_{(p+d)}$ and $t_{(q+d)}$, we require that $\frac{\max(r_a, r_b)}{\min(r_a, r_b)} - 1 \leq \varepsilon$, where ε is a coherence threshold.

For example, Table 1(b) shows two td-clusters $C_1 = \{g_5\} \times \{t_1, t_2, t_4\} \cup \{g_1, g_4\} \times \{t_2, t_3, t_5\} \cup \{g_9\} \times \{t_4, t_5, t_7\}$ and $C_2 = \{g_2, g_8\} \times \{t_1, t_2, t_4\} \cup \{g_7\} \times \{t_2, t_3, t_5\} \cup \{g_3, g_6\} \times \{t_3, t_4, t_6\}$ embedded in Table 1(a). Apparently, their similarity can not be revealed by previous models.

In the td-cluster model, any two genes have a time-delayed scaling relationship on their corresponding time sequences. Moreover, the previous scaling pattern is just a special case of the td-cluster model when d is equal to 0. Although a td-cluster in Definition 1 represents a time-delayed scaling cluster, our definition can be easily generalized to cover time-delayed shifting, time-delayed inverting or other types of time-delayed patterns as well just by modifying the second condition of the td-cluster definition, which determines the types of td-cluster.

Let \mathcal{B} be the set of all td-clusters that satisfy the given homogeneity conditions, then $C \in \mathcal{B}$ is called a maximal td-cluster iff there doesn't exist another cluster $C' \in \mathcal{B}$ such that $C \subset C'$.

Problem Statement. Given: (1) ε , a coherence threshold, (2) min_t , a minimal number of time points, and (3) min_g , a minimal number of genes, the task of mining is to find all maximal td-clusters according to Definition 1 that satisfy all the given thresholds.

3 Algorithm

This section introduces our depth-first algorithm with some useful pruning strategies which effectively and efficiently mines all the significant td-clusters based on a TG-tree structure.

3.1 Construct Initial TG-Tree

To make our presentation clear and easy to follow, we first look at what the initial TG-tree is like. Then, we describe how it is constructed. Figure 3 shows the initial TG-tree constructed from Table 1. It contains the td-clusters for all pairs of time points (i.e., all time sequences with length two). The td-clusters are linked to the leaf nodes and there can be multiple td-clusters linked to the same cell. Each td-cluster $C = \bigcup_{s=1}^u G_s \times T_s$ is composed of a set of buckets and each bucket represents a submatrix $G_s \times T_s$. The buckets are sorted upwards by its time sequence T_s . And we call the first bucket baseline bucket as the time sequence of the first bucket (T_1) is composed of the time points in the path from the root to the node that the td-cluster C linked to. Each bucket contains a number that records the time intervals that T_s is delayed for compared with T_1 . Apparently, the number of the baseline bucket is 0. For example, the leftmost td-cluster in Fig. 3 is composed of three buckets. The time sequence of the baseline bucket (T_1) is $\{t_1, t_2\}$. And the time sequence of the second bucket is $\{t_2, t_3\}$ as the bucket number is 1.

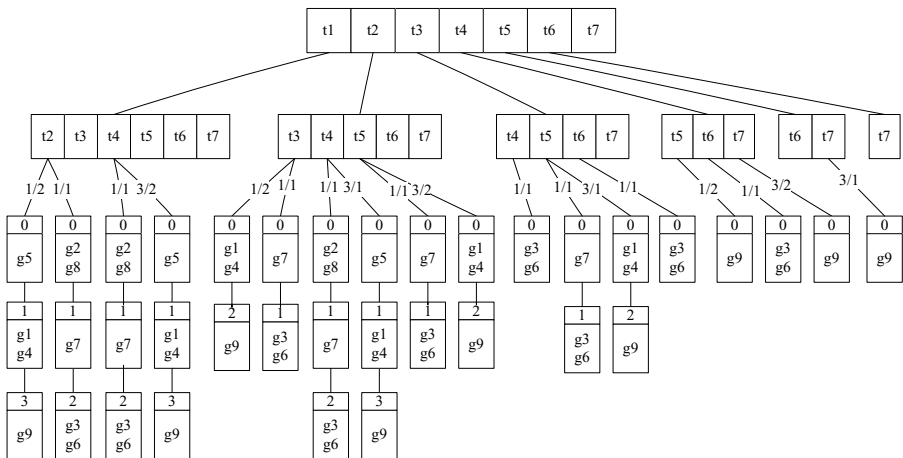


Fig. 3. Initial TG-tree

Let $C = \bigcup_{s=1}^u G_s \times T_s$ be a td-cluster and the length of T_s ($s \in [1, u]$) is two. Let $T_1 = \{t_p, t_q\}$ be the baseline time sequence, let g_a be any gene of G_s ($s \in [1, u]$) and suppose the corresponding time sequence T_s is delayed for d time intervals compared with T_1 . Let $r_a = \frac{c_a(p+d)}{c_a(q+d)}$ be the ratio of the expression level of gene g_a at time $t_{(p+d)}$ and $t_{(q+d)}$. A ratio range of td-cluster C is defined as the value scope of r_a , $[\min(r_a), \max(r_a)]$.

The construction of the initial TG-tree has the following steps:

Step 1. We begin with the time sequence $\{t_1, t_2\}$ which corresponds to path $t_1 t_2$ in Fig. 3 and find all of its baseline buckets. Each baseline bucket corresponds to a scaling gene set on $\{t_1, t_2\}$ and it is the first bucket of a td-cluster to be extended. Let $r_i^{12} = c_{i1}/c_{i2}$ be the ratio of the expression values of gene g_i in columns t_1 and t_2 , where $i \in [1, n]$. Using a sliding window approach (with window size: $r_i^{12} \times \varepsilon$ for each gene g_i) over the sorted ratio values, we can find all scaling gene sets (baseline buckets) for $\{t_1, t_2\}$, i.e., the genes in each window that is not subsumed by its preceding windows form a scaling gene set.

Note that the experimental result indicates that there are massive td-clusters with large overlap because of the large overlap of the windows when using the basic sliding window approach mentioned above (i.e. the left boundary of the window slides forward one gene at one time). Thus we propose two improved approaches here. One is that the left boundary of the window also slides forward one gene at one time but we merge the adjacent overlapping windows if the extended range is not more than 2ε .

The other is as follows. The left boundary of the window doesn't slide forward just one gene at one time, as shown in Fig. 4, the next window begins from the gene just out of its preceding window. Assume window 1 and window 2 are very close ($r_{i2}^{12} = r_{i3}^{12}$ or $r_{i2}^{12} \approx r_{i3}^{12}$) and the range of the two windows is not more than 2ε , then we merge them into an extended window. And assume window 3 cannot be merged with windows 1 and 2 (i.e., the extended range is too wide, say more than 2ε), then the genes in windows 1 and 2 form a baseline bucket. Similarly assume window 4 cannot be merged with window 3, then the genes in window 3 form a baseline bucket. Moreover, if the adjacent baseline buckets are close i.e. the ratio values near the boundary of the two baseline buckets are approximately equal, then to avoid missing any potential cluster we also add an overlapping patched window near the boundary and set the maximum range of the patched window equal to the larger of the two adjacent baseline buckets. The genes in the patched window form a baseline bucket too. For example, in Fig. 4, assume the first two baseline buckets are close ($r_{i5}^{12} = r_{i6}^{12}$ or $r_{i5}^{12} \approx r_{i6}^{12}$), then we add an overlapping patched window and set its maximum range 2ε . Note that there can be multiple positions near the boundary for the patched window, and we choose the one which leads to the minimal scope of the ratio values within the patched window. Also note that an added advantage of allowing extended windows is that it makes the method more robust to noise, since often the users may set a stringent ε condition, whereas the data might require a larger value.

Step 2. Generate baseline buckets for all time sequences $\{t_{(1+d)}, t_{(2+d)}\}$ one by one, where $d \in [1, m - 2]$. After generating the baseline buckets for each

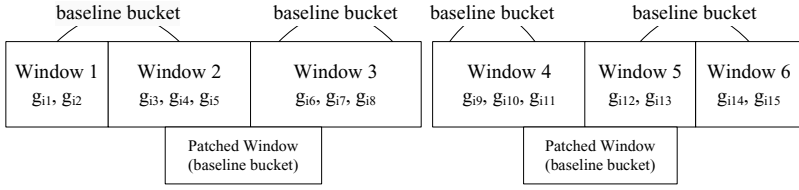


Fig. 4. The second improved sliding window approach

$\{t_{(1+d)}, t_{(2+d)}\}$, we need to use these buckets to backwards generate the bucket with number $(1 + d - i)$ for all time sequences $\{t_i, t_{(1+i)}\}$ before $\{t_{(1+d)}, t_{(2+d)}\}$, where $i \in [1, d]$. For example, the baseline bucket b of $\{t_3, t_4\}$ can be used to extend the td-cluster C of $\{t_1, t_2\}$ if the ratio ranges of b and C can be merged, i.e., the merged ratio range is within 2ϵ considering the noise in the raw data, and further we update the ratio range of $\{t_1, t_2\}$ with the merged range. As shown in Fig. 3, the baseline bucket of $\{t_3, t_4\}$ which links to the end of the path t_3t_4 can be used as a bucket with number 2 to extend the td-cluster with the ratio range $[1/1, 1/1]$ of $\{t_1, t_2\}$ and also as a bucket with number 1 to extend the td-cluster with the ratio range $[1/1, 1/1]$ of $\{t_2, t_3\}$. Note that we don't need to generate new buckets. Instead, we only need to keep pointers to the corresponding baseline buckets. Figure 3 shows the logical structure of the initial TG-tree and only baseline buckets exist in main memory.

Step 3. For all time sequences $\{t_1, t_i\}$ (with $i \in [3, m]$), repeat the process in step 1 and 2 just as $\{t_1, t_2\}$ does.

After finishing all the above steps, the initial TG-tree is constructed, as Fig. 3 shown.

3.2 Mine Td-Clusters by Developing TG-Tree

We develop the TG-tree in a depth-first way to mine all maximal td-clusters that satisfy all the given thresholds. Let $T_s = \{t_{s_1}, t_{s_2}, \dots, t_{s_l}\}$ be the current time sequence to be extended. We first find all time sequences with length two that can extend T_s by one time point. The method is as follows: we locate t_{s_l} in the root node, and t_{s_l} plus any time point in the node linked to it form a time sequence that can extend T_s by one time point. For example, as shown in Fig. 3, suppose T_s is $\{t_1, t_2\}$, then we locate t_2 in the root node. And the node linked to t_2 is composed of t_3, t_4, t_5, t_6, t_7 thus $\{t_2, t_3\}, \{t_2, t_4\}, \{t_2, t_5\}, \{t_2, t_6\}, \{t_2, t_7\}$ are the time sequences that can extend $\{t_1, t_2\}$ by one time point and the extended time sequences are $\{t_1, t_2, t_3\}, \{t_1, t_2, t_4\}, \{t_1, t_2, t_5\}$, etc.

Let $T_r = \{t_{s_l}, t_r\}$ be any time sequence that can extend T_s . Then we create a new node which contains the set of t_r and link the new node to the tail of T_s (cell t_{s_l}). For example, as shown in Fig. 5, suppose T_s is $\{t_1, t_2\}$, then the set of t_r is composed of t_3, t_4, t_5, t_6, t_7 . Thus we create a new node which contains t_3, t_4, t_5, t_6, t_7 and link the node to the tail of $\{t_1, t_2\}$. Note that before linking the new node, to avoid missing any potential cluster we should judge whether the td-clusters on T_s satisfy output conditions and output the valid ones to the result set.

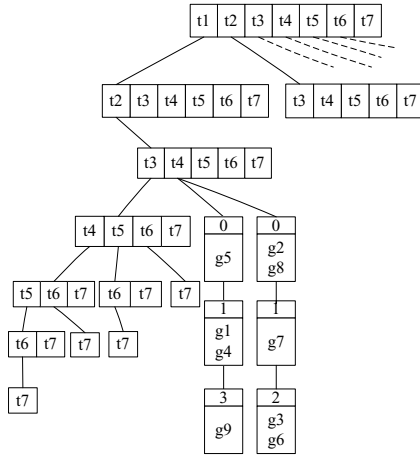


Fig. 5. TG-tree during development

Next we generate the td-clusters for each extended time sequence $T'_s = \{t_{s_1}, t_{s_2}, \dots, t_{s_l}, t_r\}$. Let C_s be any arbitrary td-cluster on T_s and let C_r be any arbitrary td-cluster on T_r . For each bucket b_s in C_s , if there exists a bucket b_r in C_r whose bucket number is the same as b_s i.e. the time sequences of b_s and b_r are delayed for the same time intervals compared with T_s and T_r respectively, then we get the intersection of the genes in b_s and b_r as the time sequence of b_r can extend the time sequence of b_s . A new bucket is generated which contains the intersection of genes if it is not empty. After all the buckets in C_s and C_r that satisfy the above condition finish the intersection operation, all the new buckets generated form a td-cluster on T'_s . For example, as shown in Fig. 5, $\{t_2, t_4\}$ is one of the time sequences that can extend $\{t_1, t_2\}$ and we get td-clusters on $\{t_1, t_2, t_4\}$ which link to the end of the path $t_1 t_2 t_4$. Finally we recursively extend each time sequence T'_s to continue mining valid td-clusters.

Note that the initial TG-tree prunes away much of the noise and irrelevant information. Moreover, the depth of the search is likely to be small, since microarray datasets have far fewer times than genes.

3.3 Pruning Rules

We next look at the pruning techniques to improve the performance of the basic algorithm introduced above.

Pruning rule 1. min_g pruning: For a td-cluster linked to a cell v , we prune it if it contains less than min_g genes, as further extension of the corresponding time sequence will only reduce the number of genes in the td-cluster. Moreover, we stop the search after v if all the td-clusters linked to it are pruned.

Pruning rule 2. min_t pruning:

2(a) For a time sequence $\{t_p, t_q\}$ with length two, let T_s be any arbitrary extended time sequence from $\{t_p, t_q\}$, then according to our algorithm, the

expression of T_s must be $\{t_{s_1}, \dots, t_p, t_q, \dots, t_{s_l}\}$. Thus the longest extended time sequence from $\{t_p, t_q\}$ is $T_{\max} = \{t_1, \dots, t_{(p-1)}, t_p, t_q, t_{(q+1)}, \dots, t_m\}$. If the length of T_{\max} is less than \min_t (i.e., $p + (m - q + 1) < \min_t$), then $\{t_p, t_q\}$ cannot lead to any coherent gene cluster having \min_t or more times, and thus all the td-clusters on it can be pruned when constructing the initial TG-tree and the search after it can also be pruned. For example, in Fig. 3, we can prune the search after $\{t_2, t_6\}$ in the case of $\min_t = 5$, as the longest extended time sequence from $\{t_2, t_6\}$ is $\{t_1, t_2, t_6, t_7\}$.

- 2(b) When constructing the initial TG-tree, we only need to generate the td-clusters on $\{t_p, t_q\}$ when the condition $q - p \leq m - \min_t + 1$ is met, applying pruning rule 2(a). And we can further prune the buckets in the td-clusters on these time sequences. Let C be a td-cluster on $\{t_p, t_q\}$ and let b be a bucket in C . Assume the bucket number of b is d , then the time sequence of b is $\{t_{(p+d)}, t_{(q+d)}\}$. Since the longest extended time sequence from $\{t_p, t_q\}$ is $T_{\max} = \{t_1, \dots, t_{(p-1)}, t_p, t_q, t_{(q+1)}, \dots, t_m\}$, the longest time sequence from $\{t_{(p+d)}, t_{(q+d)}\}$ extended along with $\{t_p, t_q\}$ is $T_{\max}^d = \{t_{(1+d)}, \dots, t_{(p-1+d)}, t_{(p+d)}, t_{(q+d)}, t_{(q+d+1)}, \dots, t_m\}$. If the length of T_{\max}^d is less than \min_t (i.e., $p + m - (q + d) + 1 < \min_t$), then $\{t_{(p+d)}, t_{(q+d)}\}$ cannot be extended to having \min_t or more times, and thus b can be pruned if its bucket number $d > m - \min_t + 1 - (q - p)$. For example, in Fig. 3, the time sequence of the bucket 3 in the leftmost td-cluster on $\{t_1, t_2\}$ is $\{t_4, t_5\}$, and thus the bucket can be pruned in the case of $\min_t = 5$, as the longest time sequence from $\{t_4, t_5\}$ extended along with $\{t_1, t_2\}$ is $\{t_4, t_5, t_6, t_7\}$.
- 2(c) During development based on the initial TG-tree, for a cell v in a node with height h , assume the number of times that are likely to occur after v is k . We prune the search after v if $h + k < \min_t$, as any td-cluster along the path cannot satisfy the requirement of \min_t . For example, in Fig. 5, we prune the search after cell t_4 in the root node in the case of $\min_t = 5$, as the number of times of all td-clusters in its subtree is less than 5.

4 Experiments

To evaluate the performance of our td-cluster algorithm, we perform experiments on both synthetic and real microarray datasets, on a 2.4-GHz Dell PC with 512 M memory running Window XP. For the real dataset we used a yeast gene expression data [5], available at http://yscdp.stanford.edu/yeast_cell_cycle/cellcycle.html. The dataset contains the expression levels of over 6000 genes under 17 timepoints, which are the normalized fluorescence between 0 and 160 minutes after cell cycle reinitiation from start. And after eliminating the negative expression levels, 6330 genes are included in our calculation.

We evaluate the efficiency of our algorithm on synthetic datasets, which are obtained with a data generator algorithm with three input parameters: number of genes (# genes), number of times (# times), and number of embedded clusters (# clusters). We set the default parameters of the data generator algorithm as

genes = 3000, # times = 20 and # clusters = 20. The synthetic dataset is initialized with random values ranging from 0 to 10. Then a number of # clusters perfect td-clusters of average dimensionality 6 and average number of genes equal to $0.01 * \# \text{ genes}$ are embedded into the initialized data, which are validated with regard to $\varepsilon = 0$.

4.1 Efficiency

Given the default parameter setting of the data generator algorithm above, we test the scalability of td-cluster with the basic sliding window approach and the two improved ones respectively by varying only one input parameter while keeping the other two as default. The average runtime of td-cluster when we vary the parameters invoked with $min_g = 0.01 * \# \text{ genes}$, $min_t = 6$, and $\varepsilon = 0.005$ is illustrated in Fig. 6. As we can observe, the runtime of the td-cluster algorithm with all the three sliding window approaches are approximately linear in terms of the number of genes (# genes). It shows worse scalability with the number of times (# times) for all of the three. This is because, the TG-tree is constructed on the times; the number of times has a more direct effect on the runtime. However typically, the number of times is much less than the number of genes in real microarray datasets. Figure 6(c) shows an approximately linear relationship between the runtime of the td-cluster algorithm and the number of clusters (# clusters).

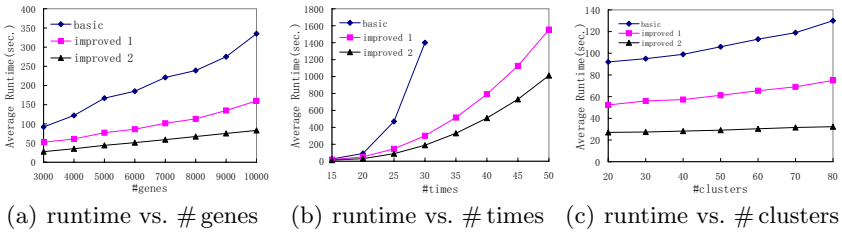


Fig. 6. Evaluation of efficiency on synthetic datasets

From Fig. 6 we can see the two improved sliding window approaches are apparently faster than the basic one. This is because the improved approaches massively reduce the number of windows with large overlap from the very beginning and hence massively reduce the following calculation.

4.2 Effectiveness

We ran the td-cluster algorithm on the 2D 6330×17 yeast dataset with $min_g = 40$, $min_t = 6$, $\varepsilon = 0.008$; 17 td-clusters are output in 46.5 seconds. Due to space limit, we only report the details of three td-clusters with six times each. Figure 7(a) illustrates the gene expression profiles for td-cluster 16. As we can see, our td-cluster algorithm can successfully identify time-delayed scaling patterns

satisfying the coherence threshold. In contrast, previous pattern-based biclustering algorithms [18] only allow pure scaling patterns and hence fail to identify the td-cluster.

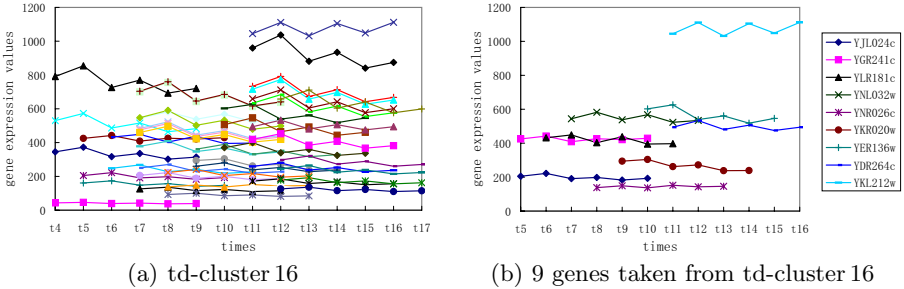


Fig. 7. One td-cluster

We apply the yeast genome gene ontology term finder (<http://db.yeastgenome.org/cgi-bin/GO/goTermFinder>) on each discovered clusters to evaluate their biological significance in terms of associated biological processes, cellular components and gene function respectively. Table 2 shows the top GO terms of the three categories and the GO terms with low p-values for td-clusters 2, 8 and 16, which have been overlooked by previous work. For example for cluster C_{16} , we find that the genes are mainly involved in vesicle-mediated transport. The tuple ($n=9, p=4.72E-05$) means that out of the 45 genes, 9 belong to this process, and the statistical significance is given by the p-value of $4.72E-05$. Figure 7(b) illustrates the expression profiles of the 9 genes, which are taken from Fig. 7(a). Note that only the most significant common terms are shown in Table 2; the other genes in the cluster share other terms, but at a lower significance. From the table, it is clear that the clusters are distinct along each category. The extremely low p-values suggest that the td-clusters are of significant biological meaning in terms of biological process, cellular component and gene function.

Table 2. Top GO terms of the discovered td-clusters

Cluster	# Genes	Process	Function	Cellular Component
C_2	42	tRNA pseudouridine synthesis ($n=2, p=5.54E-05$), pseudouridine synthesis ($n=2, p=0.00066$)	tRNA-pseudouridine synthase activity ($n=2, p=0.00022$), pseudouridine synthase activity ($n=2, p=0.00034$)	external encapsulating structure ($n=4, p=0.00308$), cell wall (sensu Fungi) ($n=4, p=0.00308$)
C_8	46	establishment of cell polarity ($n=4, p=0.00355$), establishment and/or maintenance of cell polarity ($n=4, p=0.00391$), cell organization and biogenesis ($n=16, p=0.00705$)	intramolecular transferase activity ($n=2, p=0.00261$)	intracellular ($n=40, p=2.91E-05$), membrane-bound organelle ($n=32, p=0.00036$), intracellular organelle ($n=33, p=0.00078$)
C_{16}	45	vesicle-mediated transport ($n=9, p=4.72E-05$), endocytosis ($n=3, p=0.00979$)	endopeptidase activity ($n=3, p=0.00656$)	cell ($n=40, p=0.00616$)

4.3 Effects of the Parameters

The mined td-cluster is validated with respect to three parameters, i.e., the minimum number of genes min_g , the minimum number of times min_t and the coherence threshold ε . We set their default values as $min_g = 40$, $min_t = 5$ and $\varepsilon = 0.01$. Then we test the effect of the parameters on the real GT data set by varying only one parameter while keeping the other two as default. Figure 8 show the effect of each parameter on the number of td-clusters.

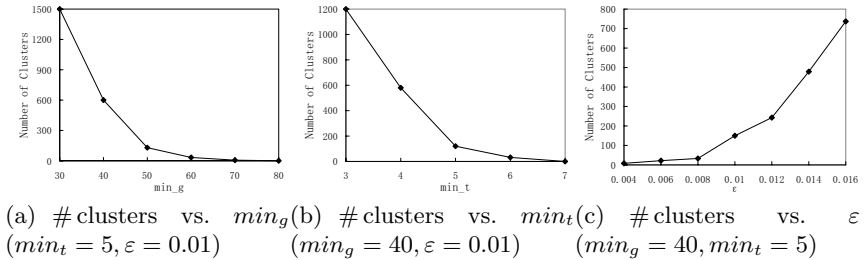


Fig. 8. Effects of the parameters on the number of clusters

Interestingly, the three curves in Fig. 8 share a similar characteristic. That is, there exist “knots” in the curves. Note that the first two curves drop sharply until “knots” are met, then the curves go stably to the right. While the third curve goes up stably until a “knot” is met, then the curve rises sharply. For example, we can see the “knots” of $min_g = 50$ in Fig. 8(a), $min_t = 5$ in Fig. 8(b) and $\varepsilon = 0.008$ in Fig. 8(c). These “knots” indicate that there exist stable and significant td-clusters in the real data set. They are highly correlated, involving a statistically significant number of genes and times. The “knots” also suggest the best settings of the parameters to avoid the coherent gene clusters formed just by chance.

5 Conclusion

In this work, we have proposed a td-cluster model for identifying arbitrary time-delayed scaling patterns from time series gene expression data. Moreover, our definition can be generalized to cover time-delayed shifting, time-delayed inverting or other types of time-delayed patterns as well. These kinds of patterns may help identify activators or some other type of components and infer causality from time series expression experiment. And we have developed a depth-first algorithm with some useful pruning strategies which effectively and efficiently mines all the significant td-clusters based on a TG-tree structure. Our experimental results prove that our td-cluster algorithm is able to discover a significantly number of biologically meaningful td-clusters missed by previous work.

References

1. C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In ACM SIGMOD Conference, 2000.
2. Z. Bar-Joseph. Analyzing time series gene expression data. *Bioinformatics*, 20 (16): 2493–2503, 2004.
3. T. Chen, V. Filkov, and S. S. Skiena. Identifying gene regulatory networks from experimental data. In *Recomb*, 1999.
4. Y. Cheng and G. M. Church. Biclustering of expression data. In 8th Int'l Conference on Intelligent Systems for Molecular Biology, 2000.
5. Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L. et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, 2: 65–73, 1998.
6. I. S. Dhillon, E. M. Marcotte, and U. Roshan. Diametrical clustering for identifying anti-correlated gene clusters. *Bioinformatics*, 19: 1612–1619, 2003.
7. M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Science, USA*, 95 (25): 14863–14868, 1998.
8. S. Erdal, O. Ozturk, D. Armbruster, H. Ferhatosmanoglu, and W. Ray. A time series analysis of microarray data. In 4th IEEE Int'l Symposium on Bioinformatics and Bioengineering, May 2004.
9. J. Feng, P. E. Barbano, and B. Mishra. Time-frequency feature detection for time-course microarray data. In 2004 ACM Symposium on Applied Computing, 2004.
10. V. Filkov, S. Skiena, and J. Zhi. Analysis techniques for microarray time-series data. In 5th Annual Int'l Conference on Computational Biology, 2001.
11. T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburttty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102: 109–126, 2000.
12. J. Liu, W. Wang, and J. Yang. Gene ontology friendly biclustering of expression profiles. In *Computational Systems Bioinformatics*, 2004.
13. S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1 (1): 24–45, 2004.
14. J. Qian, M. Dolled-Filhart, J. Lin, H. Yu, and M. Gerstein. Beyond synexpression relationships: Local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions. In *Journal of Molecular Biology*, 2001.
15. H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In ACM SIGMOD Conference, 2002.
16. B. -K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proc. of the 14th Intl. Conf. on Data Eng. (ICDE '98)*, 201–208, Orlando, February 1998.
17. H. Yu, N. Luscombe, J. Qian, and M. Gerstein. Genomic analysis of gene expression relationships in transcriptional regulatory networks. *Trends Genet*, 19 (8): 422–427, 2003.
18. L. Zhao and M. J. Zaki. Tricuster: An effective algorithm for mining coherent clusters in 3d microarray data. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 2005.

Mining Delay in Streaming Time Series of Industrial Process^{*}

Haijie Gu and Gang Rong

National Key Laboratory of Industrial Control Technology,
Zhejiang University, Hangzhou, 310027, P.R. China
{hjgu, grong}@iipc.zju.edu.cn

Abstract. Time delay is a general phenomenon in industrial process. Accurate evaluation on delay is important in data preprocessing when mine manufactory process data. As typical streaming time series, sensors' data of industrial process have attracted much attention recently. A new concept, *trend similarity search*, is proposed based on raw monotony between two industrial process variables. The new concept is for those two time series which are similar only in trend but dissimilar in shape, whereas similarity search may not do well in such condition. An algorithm **DelayMine** is also proposed to mine delay between two interrelated time series by trend similarity search. Moreover, the DelayMine is extended to online algorithm for processing streaming time series. The properties and performance of DelayMine is demonstrated through experiments both on systems with steady and time-varying delay.

1 Introduction

Data mining developed rapidly last decade, and applied successfully in many fields such as financial, retail, telecom and networks. However, its application in process industry encounters some troubles. One problem of mining the process data is time delay. For example, X and Y are two attributes (process variable), $X(t)$ is usually associated with $Y(t + \tau)$ but not $Y(t)$. The delay τ needs to be cleaned in the phase of data preprocessing.

It can be seen that in some cases there are raw monotony between two inter-related process variables, for example, the temperature of tank will rise when the input flow of hot water increases. As a result, we can get the delay by analyzing the trends between the two time series of interrelated variables. Similarity search has been the focus of time series research in the past decade [6,9,10]. The main three issues studied in similarity search are the similarity model, the data representation and the index structure. In our scenario, the two time series differ from each other obviously after moving delay part. In other words, the distances between them is large using whatever similarity models, such as Euclidian distance, time warping [7] or amplitude shifting. The internal reason is that they are not similar in shape but only in trend, as fig.1 shows. That results in our new concept, trend similarity search, which extracts the trend of each segment

^{*} This work is supported by National Natural Science Foundation of China (60421002).

in time series for searching. We propose an algorithm BasicDelayMine to mine the delay between two time series by trend similarity search.

Data streams have drawn much interest in recent years [1]. Sensors' data from a number of industrial process units are of not only time series but also data streams, thus they are called streaming time series in some studies [4,5]. One of the core issues in data streams compared to data sets is online mining of the changes [3]. In [5,8], approaches of aggregation on time dimension are proposed. In this work, the representation of each regression line differs from [8], and the aggregation formula is described for time granularity transformation. Furthermore, we extend BasicDelayMine to OnlineDelayMine using sliding window in order to catch the changes of delay.

Our contributions can be summarized as follows.

- Two widely existed features of the process variables referring to the same unit are assumed and described, which are raw monotony and nonperiodicity.
- Data mining technology is introduced into evaluation of the delay in industrial process.
- A new concept called trend similarity search distinguished from similarity search is proposed.
- Algorithms are developed to get the delay not only for time series set but also for streaming time series.

In the remainder of this paper, Section 2 explains the motivation and preliminary preparation of this work, Section 3 develops the algorithm BasicDelayMine, Section 4 extends the algorithm to data streams, Section 5 presents the results of experiments, Section 6 concludes this paper.

2 Background

2.1 Time Delay

Time delay is a common phenomenon in many processes due to material and energy transportation lag, measurement delay, and so on. In this paper the delay will be denoted by τ . It is assumed that the operation of variable X will impact the variable Y after τ , and we should relate $X(t)$ to $Y(t + \tau)$ when we try to find some patterns from process data. To steady delay systems, offline experiments are made before the running of plant. This method is risky if the conditions of experiments are not accordant to the real ones. In addition, the delay is changing more or less in fact. As a result, there is a practical method known as exhaustion that searches the best τ among an assumed interval. The best τ is the one that makes the total errors of the model from X to Y smallest. This method will be less of efficiency if the possible interval is large, because all the parameters of the model have to recompute at each τ .

We introduce data mining technology into evaluation of the delay in this work. The main idea is based on two features that widely exist among variables of the same unit. The precondition of this work is on the assumption that the process matches with the two features as follows:

Feature 1 (raw monotony). Increase of the input leads to increase (or decrease) of the output of the model in general, although the increasing (decreasing) process of output maybe exist vibrations. We call this feature as raw monotony.

Feature 2 (nonperiodicity). In general, process variables are nonperiodic.

Accounting to feature 1, the delay could be evaluated by comparing the trends of two time series. The answer is not unique if without feature 2, because any $(\tau + k \cdot T)$ may be the potential answer, where τ is the real delay, T is the period and $k \in N$.

2.2 Problem Description

Sensors' data sampled with fixed step are time series. The model is a function from input signal X to output signal Y . Signal X is an infinite time series denoted by $\langle x(0), x(1), \dots, x(k), \dots \rangle$, and the interval between $x(k)$ and $x(k + 1)$ is the sample time that is fixed and denoted by *samp*, so do Y . If Y is delayed d steps to X , then $\tau = d \cdot \text{samp}$.

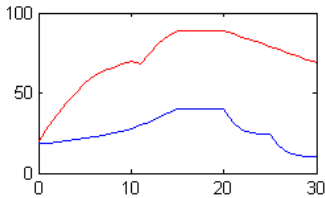


Fig. 1. Not similar in shape, but similar in trend

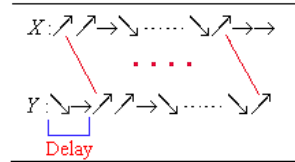


Fig. 2. Compute delay using trend similarity search

Our main idea to detect τ is shown in fig.2, where X is input and Y is output of the model. X is divided into several segments as the same as Y . We use \nearrow , \rightarrow and \searrow to represent *increasing*, *stable* and *decreasing* of each segment. The trend sequence of Y from the third segment is much similar to that of X from the beginning, thus the time of two segments is the delay obviously.

3 Proposed Method

3.1 Equispaced Linear Regression

The first step of evaluating delay is dividing time series into segments and catching the trend from each segment.

Segment approximation approaches have been widely investigated in similarity search of time series. The main approaches are Piecewise Aggregate Approximation (PAA), Adaptive Piecewise Constant Approximation (APCA)[9], Piecewise Linear Approximation (PLA)[10], Piecewise Quadratic Approximation (PQA), Symbolic Aggregate approximation (SAX)[11].

Piecewise Linear Regression(PLR)[12] is one kind of PLA. PLR is not suitable for evaluating delay due to different segmenting points of two series. We propose Equispaced Linear Regression(ELR) to approximate X and Y in which segments are divided into equal length. ELR have two important characters like PLR as follows:

- It's convenient for online combining two neighboring segments.
- The slope of regression line has the information of trend we cared.

Definition 1 (Equispaced Linear Regression (ELR)). *Given time series $U = \langle u(0), u(1), \dots, u(k), \dots, u(N) \rangle$, $u(k)$ is sampled at time t_k , where $t_{k+1} - t_k = samp$ ($k = 0, 1, 2, \dots, N - 1$). We set each segment combined with p points, assuming $N = m \cdot (p - 1)$ without losing generality. The m segments of the same length divided from U are denoted by s_1, s_2, \dots, s_m . We define $u(i : j) = \langle u(i), u(i + 1), \dots, u(j) \rangle$.*

Makes linear fit for m segments using the least square error method. The slope $\hat{\alpha}_i$ and the base $\hat{\beta}_i$ of each s_i are obtained from the following lemma.

Lemma 1. *The values of $\hat{\alpha}_i$ and $\hat{\beta}_i$ for each s_i are as follows:*

$$\hat{\alpha}_i = \bar{u} - \hat{\beta}_i \cdot \bar{t} \tag{1}$$

$$\hat{\beta}_i = \frac{\sum_{k=(p-1)(i-1)}^{(p-1)i} (t_k - \bar{t})(u(k) - \bar{u})}{\sum_{k=(p-1)(i-1)}^{(p-1)i} (t_k - \bar{t})^2} = \frac{\sum_{k=(p-1)(i-1)}^{(p-1)i} (t_k - \bar{t})u(k)}{\frac{1}{12}(p-1)p(p+1) \cdot samp^2} \tag{2}$$

where $\bar{t} = \frac{1}{p} \sum_{k=(p-1)(i-1)}^{(p-1)i} t_k = \frac{1}{2}(t_{(p-1)(i-1)} + t_{(p-1)i})$, $\bar{u} = \frac{1}{p} \sum_{k=(p-1)(i-1)}^{(p-1)i} u(k)$.

3.2 Trend Vector

We obtain slope $\hat{\alpha}_i$ of each segment by using ELR in 3.1. As shown in fig.1, the two time series are similar in trend but the slopes of corresponding part are unequal. Our strategy is that only care about three kinds of trends as increasing, stable and decreasing fuzzily, such a strategy also accords with feature 1 in section 2.1. Furthermore, the discrete values of slopes are saved in a vector called Trend Vector(TrV).

Definition 2 (Trend Vector). *The length of Trend Vector (TrV) is equal to the number of segments m . Each element in TrV is defined as follows:*

$$TrV(i) = \begin{cases} 1, & \text{if } \hat{\alpha}_i > max_steady, \\ 0, & \text{if } min_steady \leq \hat{\alpha}_i \leq max_steady, \\ -1, & \text{if } \hat{\alpha}_i < min_steady. \end{cases} \tag{3}$$

Where 1, 0 and -1 represent *increasing*, *stable*, *decreasing* separately, *max_steady* is the user specified critical slope between *increasing* and *stable*, *min_steady* is as well as the user specified critical slope between *stable* and *decreasing*, $i = 1, 2, \dots, m$.

3.3 Trend Sequence Index

We utilize sliding window technology to search in TrV_Y the similar trend subsequence with TrV_X . Assumed that the length of sliding window is w ($w < m$). Given Query Sequence Q with w -length in TrV_X , it starts *offset* away from the beginning of TrV_X . Thus Q can be denoted by $TrV_X(offset : (offset + w - 1))$, while sliding window W_j is the subsequence $TrV_Y((offset + j) : (offset + j + w - 1))$ in TrV_Y . W_j moves step by step in TrV_Y to find the one matched with Q most, where $j \in [0, m + 1 - w - offset]$. The degree of match is measured by $matchRatio = common_elements/w$, where *common_elements* is the total common elements in Q and W_j . The time complexity of such an index is $O((m + 2 - w - offset) \cdot w)$. The algorithm called BasicDelayMine is presented as Table 1.

It's assumed that W_j is matched with Q in the highest ratio, then the delay is evaluated by $\tau = j \cdot p \cdot samp$. There is a precondition of this method: N should be large enough.

Table 1. An algorithm to find the delay between two time series

Algorithm: BasicDelayMine

Input: (1) time series $X(0 : N)$ and $Y(0 : N)$

(2) user-specified parameters: p , slope thresholds, *offset* and w

Output: Delay τ , *matchRatio*

Method:

- 1: Let $m = INT(N/(p - 1))$, reset $N = m \cdot (p - 1)$; //INT gets aliquot part
 - 2: Compute slope $\hat{\alpha}_i(X)$ in m equispaced segments of $X(0 : N)$ using equation (1) and (2), compute $\hat{\alpha}_i(Y)$ in the same way;
 - 3: Compute TrV_X and TrV_Y using equation (3);
 - 4: Search the subsequence in TrV_Y which matches with Query Q in TrV_X best using sliding window;
 - 5: **Return** τ and *matchRatio*;
-

4 Extended Method in Data Streams

Changes of pattern are one of the key issues in data streams. The delay evaluating algorithm should be an online one to catch the changes of delay.

4.1 Time Granularity Transformation

We call the segment length p as time granularity. There are requirements to transform time granularity conveniently. First, computation can speed up when p increases. Second, algorithm is not strong enough to noise if p is too small while some trend information may lost if p is too large.

A method of aggregation on the time dimension was proposed in [8]. However, it can be seen that in our algorithm the start of current segment is just the end of last segment. For example, two neighboring segments are of time interval $[0,9]$ and $[10,19]$ in paper [8], while $[0,9]$ and $[9,18]$ in our algorithm, because in $[9, 10]$ there also exists trend information. Thus a new representation of regression line called BtEvSB($t_b, u_e, \hat{\alpha}, \hat{\beta}$) is proposed here, where t_b is the **B**eginning time, u_e is the **E**nd value (raw data, not fit data), $\hat{\alpha}$ is the **S**lope, $\hat{\beta}$ is the **B**ase.

It's assumed that s_a is the new segment combined from K segments s_1, s_2, \dots, s_K , and its BtEvSB is ($t_b^a, u_e^a, \hat{\alpha}_a, \hat{\beta}_a$). The BtEvSB of s_a can be calculated only from p , $samp$ and BtEvSBs of K segments. The proofs are not expanded here.

4.2 Online Algorithm

We should track the current delay to update the model. In many work of data streams, it is not just that remove the oldest data but remove those can't represent the new concept [2]. However, the current delay relies on current data in our scenario, the new data represent the new concept. Thus we utilize sliding window to realize online computation.

In section 3, the time series are of $(N + 1)$ -length. So we set the length of sliding window as $N + 1$ in OnlineDelayMine, and the sliding window moves only when a block of new $p - 1$ data coming. A new segment is born consisting of the last point of the window before moving and the new $p - 1$ data, while the oldest $p - 1$ data are dead. Then make a linear fit on the new segment and update Trend Vector and BtEvSB corresponding.

In order to catch the change of delay, comparing to BasicDelayMine a new index strategy is more suitable here that define the last w elements from back in TrV_Y as query sequence Q instead of the head in TrV_X . In that case, we search the best match sequence in the reverse order of TrV_X with the query sequence, that resulted in the newest data is always taken into consideration first.

5 Experimental Results

5.1 Steady Delay Experiment

The algorithm proposed was tested on a pilot plant with multiple tanks. The input variable X of the model is the *flow* of input liquid while the output variable Y is the *height* of liquid in tank. The model is nonlinear in our device. The delay of this system is steady in the same experimental condition and its value can be evaluated by offline experiments. We collected 1500 points data from sensors whose sample time was $0.5s$.

The value of τ is $36s$ evaluated by offline experiments on Step Response Model. We changed p , w and trend thresholds respectively in our experiments, some results of BasicDelayMine were shown in table 2, from which we can see that the algorithm is robust if the user-specified parameters in a suitable rang. Obviously, the last row presented a wrong result, because the trend of each segment in X was *stable* under such a trend threshold. We note that a suitable p

Table 2. Experimental results on the system whose delay is steady

Step p	$max_steady(X)$	$min_steady(X)$	$max_steady(Y)$	$min_steady(Y)$	w	τ	$matchRatio$
5	0.002	-0.002	0.02	-0.02	20	36s	0.80
10	0.002	-0.002	0.02	-0.02	20	36s	0.95
5	0.002	-0.002	0.02	-0.02	40	36s	0.85
5	0.015	-0.015	0.02	-0.02	20	36s	0.90
5	0.1	-0.1	0.02	-0.02	20	0s	0.70

is based on the speed of signal variation while a suitable threshold distinguishing trend is based on the amplitude of signal.

5.2 Time-Varying Delay Experiment

Furthermore, we tested our algorithm on the time-varying delay system. We designed an experiment that called hardware-in-the-loop simulation to make sure we know the real variation of delay in advance. A simulation part of time-varying delay was inserted into the signal transport part of the sensor that recorded **height** of liquid mentioned in section 5.1. The designed and evaluated value of delay were shown in fig.3. The sliding window of data was set as 160 points, and other parameters were set as row 1 in table 2. As a designed result, the delay was 48s from time of 350 and 40s from time of 600. As a result, it seemed that the detection of change was always late to some extent. In fact it was unavoidable because any identification method is based on a series of historical data under some principles. We considered a system evolving into another situation only when the new phenomena last some time. It was showed that we could catch the evolvement of delay from the new sensor data using our algorithm.

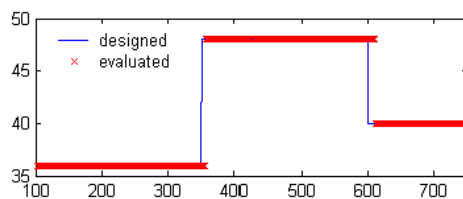


Fig. 3. Experimental results on the time-varying delay system. Both the designed value of delay and the evaluation result were presented.

6 Conclusions

In this paper, we proposed a new concept called trend similarity search differed from similarity search, which was based on two features widely existed in industrial process. Data mining technology was introduced into identification of delay in this work. We developed BasicDelayMine and OnlineDelayMine to mine delay between two relative time series in data sets and data streams respectively.

Future directions for research include a better search strategy on trend vector, a principle to measure the confidence on the evaluation result, and maybe a self adaptive formulation of parameter setting from the time series straight without human analysis. Furthermore, it's a challenge to analyze the delay of such time series models that are unmatched with the two features.

References

1. Gaber, M. M., Zaslavsky, A. Krishnaswamy, S.: Mining Data Streams: A Review. SIGMOD Record, 34(2),(2005) 18-26
2. Hulten, G., Spencer, L., Domingos, P.: Mining Time-Changing Data Streams. Proc. ACM SIGKDD (2001) 97-106
3. Kifer, D., Ben-David, S., Gehrke, J.: Detecting Change in Data Streams. Proc. VLDB (2004) 180-191
4. Gao, L., Wang, X.S.: Continually evaluating similarity-based pattern queries on a streaming time series. Proc. SIGMOD, (2002) 370-381
5. Palpanas, T., Vlachos, M., Keogh, E.J., Gunopulos, D., Truppel, W.: Online Amnesic Approximation of Streaming Time Series. Proc. ICDE, (2004) 338-349
6. Perng, C.S., Wang, H., Zhang, S.R., Parker, D.S.: Landmarks: A New Model for Similarity-Based Pattern Querying in Time Series Databases. Proc. ICDE, (2000) 33-44
7. Yi, B.-K., Jagadish, H., Faloutsos, C.: Efficient retrieval of similar time sequences under time warping. Proc. ICDE, (1998) 201-208
8. Chen, Y., Dong, G., Han, J., Wah, B.W., Wang, J.: Multi-Dimensional Regression Analysis of Time-Series Data Streams. Proc. VLDB, (2002) 323-334
9. Keogh, E.J., Chakrabarti, K., Mehrotra, S., Pazzani, M.J.: Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. Proc. ACM SIGMOD, (2001) 151-162
10. Keogh, E.J., Pazzani, M.J.: An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback. Proc. KDD, (1998) 239-243
11. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. Proc. DMKD, (2003) 2-11
12. Shatkay, H., Zdonik, S.B.: Approximate Queries and Representations for Large Data Sequences. Proc. ICDE, (1996) 536-545

Segmental Semi-Markov Model Based Online Series Pattern Detection Under Arbitrary Time Scaling*

Guangjie Ling^{1,2}, Yuntao Qian^{1,2}, and Sen Jia^{1,2}

¹ College of Computer Science, Zhejiang University, Hangzhou, 310027, P.R. China

² State key Laboratory of Information Security,

Institute of Software of Chinese Academy of Sciences, Beijing, 100049, P.R. China

balley_ling@163.com, ytqian@zju.edu.cn, zjujiase@hotmai.com

Abstract. Efficient online detection of similar patterns under arbitrary time scaling of a given time sequence is a challenging problem in the real-time application field of time series data mining. Some methods based on sliding window have been proposed. Although their ideas are simple and easy to realize, their computational loads are very expensive. Therefore, model based methods are proposed. Recently, the segmental semi-Markov model is introduced into the field of online series pattern detection. However, it can only detect the matching sequences with approximately equal length to that of the query pattern. In this paper, an improved segmental semi-Markov model, which can solve this challenging problem, is proposed. And it is successfully demonstrated on real data sets.

1 Introduction

In recent years, online series pattern detection technique has attracted increasing interest in time series data mining communities, as it plays an important role in many applications such as endpoint detection in plasma etch processes and pattern detection in medical data. Efficient online detection of similar patterns under arbitrary time scaling of a given time sequence (see Fig. 1) is a challenging problem in the real-time application field of time series data mining. For example, persons reproduce the same tune or motions at different speeds [1, 2], and many financial time series also contain such similar patterns [3]. Readers are referred to [4] for details.

Some methods based on sliding window have been proposed to solve this problem. Although their ideas are simple and easy to realize, their computational loads are very expensive. So model based methods are proposed. Recently, the segmental semi-Markov model [5, 6] is introduced into the field of online series pattern detection. However, it can only detect the matching sequences with approximately equal length to that of the query pattern [7-9]. In this paper, an improved segmental semi-Markov model, which can solve this challenging problem, is proposed. And it is successfully demonstrated on real data sets.

* This research is supported partly by Science and Technology Project of Zhejiang (2006C21001).

First, some symbols to be used throughout this paper are summarized in Table 1. Then the online pattern detection can be described as follows. Given a real-time time series D and a query time series Q which is acquired by prior knowledge (where $|D| \gg |Q|$), and a scaling factor $l, l \geq 1$, which represents the maximum allowable stretching and shrinking of Q by l and $1/l$ respectively, the matching sequences of Q in D for any scaling range specified by l are located.

Table 1. Summary of symbols

Symbols	Definitions
D	Real-time time series
Q	Query pattern
$ X $	Length of sequence X
$X[i]$	The i -th entry of sequence $X(1 \leq i \leq X)$
$X[i..j]$	Subsequence of X , including entries from the i -th to the j -th
l	Scaling factor, $l \geq 1$

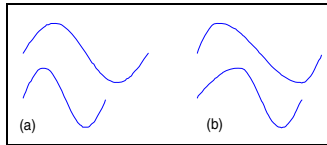


Fig. 1. Similar patterns under different time scaling. (a)Uniform scaling. (b)Arbitrary scaling.

2 Online Pattern Detection Methods Based on Sliding Window

Sliding window is a typical approach for online detection of similar patterns. The approach begins at the initial position of D , gets a window of minimum size W_{min} (where W_{min} is the lower scaling bound specified by $l, W_{min} = \lfloor |Q|/l \rfloor$), then checks whether $D[1..W_{min}]$ matches Q under some similarity measure. With the left side of the window anchored at $D[1]$, each subsequence $D[1..k]$ is scanned orderly in a similar manner to check if it matches Q for all $W_{min} \leq k \leq W_{max}$ (where W_{max} is the upper scaling bound specified by $l, W_{max} = \lceil |Q| \cdot l \rceil$). Repeat the same procedure with the window anchored at position $D[2]$, then $D[3]$ etc., until end of D .

There are many methods proposed to match similar patterns under time scaling. Keogh *et al* [2] use uniform scaling and Euclidean distance to match similar patterns. This method can only deal with the similar patterns under uniform scaling, as shown in Fig. 1(a). Similarly, the limitation also holds for the ‘‘CD-Criterion’’ technique [10].

Dynamic time warping (DTW) [11] distance compares sequences of different lengths by stretching them, so it can be used to measure the similar patterns under arbitrary time scaling. But some disadvantages have been found in practice, e.g., it may introduce fault matching patterns due to local over-scaling, and its time complexity is $O(|D| \cdot |Q|^3 \cdot (l^2 - 1/l^2))$, which is unsuitable for the real-time application.

Fu *et al* [4] utilize scaled and warped matching (SWM) and its corresponding lower bounding technique to look for similar patterns under arbitrary time scaling. Considering the left side of the window anchored at $D[i]$, the lower bounding technique starts by calculating the lower bounding distance between Q and all subsequence beginning with $D[i]$ in the range specified by l . If the distance exceeds the user-specified tolerance, we can be sure that there are no matching patterns of Q starting at $D[i]$, and the left side of the window can slide to $D[i + 1]$; otherwise, using SWM to check whether there exists subsequence similar to Q . We call this method SWM_LB for short. The pruning power P describes the effectiveness of lower bounding technique, which is defined as follows [4]:

$$P = \frac{\text{Number of objects that do not require full SWM}}{|D|}$$

And the time complexity of SWM_LB is $O(|D| \cdot ((l-1/l) + \rho) \cdot |Q|^2 + (1-P) \cdot (\rho |Q|^3 \cdot (l-1/l)))$, where ρ is the fraction of $|Q|$ (the time warping constraint $r = |Q| \cdot \rho$). Note that the larger P becomes, the more efficient the algorithm would be. As far as we know, SWM_LB is best for online series pattern detection in all sliding window based methods, so we empirically compare it to our approach in Section 5.

3 Segmental Semi-Markov Model

The basic theory of Hidden Markov Model (HMM) was proposed by Baum and his colleagues in the late 1960s and early 1970s [12]. For a HMM with the transition probability $P(s_t = j | s_{t-1} = i) = A_{ij}$, once in state i , the system will stay in it for d time units, where d has an implicit geometric distribution: $P(d) = A_{ii}^{d-1} (1 - A_{ii})$. During the stay in state i , the system generates d observations, which are conditionally independent and identically distributed.

The segmental semi-Markov model is an extension of the standard HMM. It was originally proposed in the speech recognition literature [5, 6], then Ge *et al* [8] introduced it into the field of online series pattern detection. The segmental semi-Markov model improves the standard HMM by introducing explicit state duration distributions [5] and segment observation models [6]:

- a. The duration d can have an explicit distribution that may be non-geometric, e.g., Gauss distribution, Poisson distribution.
- b. The observations of every state can have an explicit distribution to model the dependence among them.

4 Improved Segmental Semi-Markov Model

4.1 Model Construction

The segmental semi-Markov model is a good solution to detect the matching sequences with approximately equal length to that of the query pattern. However, the corresponding segmental observations of similar patterns under arbitrary time scaling may differ considerably, as shown in Fig. 2 (left), so it can not solve the problem mentioned in Section 1. In this section, we propose an improved segmental semi-Markov model which modifies the existing model from the following three aspects:

1. Introducing the offset distribution to replace the observation distributions.

Assume the subsequence of the i th segment is $Y = y_1, y_2, \dots, y_n$, and its corresponding sequence generated by linear regression function is $Y' = y'_1, y'_2, \dots, y'_n$. The offset R_i of the i th segment is defined as the root mean squared errors between Y and Y' :

$$R_i(y_1 \dots y_n, y'_1 \dots y'_n) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2} \quad (1)$$

The offset R_i describes the fitting degree between the subsequences of the i th segment and its regression line. Form Equation (1) we know that, for any similar patterns, the offset R_i is independent of other segmental observations. Hence, the difference between the observations of corresponding segments is allowable (see Fig. 2).

R_i is governed by $P(R_i | \theta_{R_i})$, where θ_{R_i} is the set of parameters for the distribution.

We use Gaussian distribution to describe the form of $P(R_i | \theta_{R_i})$:

$$P(R_i | \theta_{R_i}) \propto \begin{cases} N(\mu_{R_i}, \sigma_{R_i}^2), & R_i \geq \mu_{R_i} \\ N(R_i, \sigma_{R_i}^2), & R_i < \mu_{R_i} \end{cases} \quad (2)$$

with parameters $\theta_{R_i} = \{\mu_{R_i}, \sigma_{R_i}^2\}$.

2. Introducing the amplitude (Y coordinate) difference distribution.

Without the segmental observation distributions, the shape of a segment can not be modeled, so we introduce the amplitude difference distribution to perform this task.

Assume the time (X coordinate) of the endpoint of the i th segment is t , and the number of data points in the segment is d_i . Let us call the amplitude difference of the i th segment ΔY_i , and we define it as $\Delta Y_i = y_t - y_{t-d_i+1}$, as shown in Fig. 2(right).

ΔY_i is governed by $P(y_t - y_{t-d_i+1} | \theta_{y_i})$, where θ_{y_i} is the set of parameters for the distribution. The actual form of $P(y_t - y_{t-d_i+1} | \theta_{y_i})$ depends on the specific application.

Usually it would be the following Gaussian distribution:

$$P(y_t - y_{t-d_i+1} | \theta_{y_i}) \propto N(\mu_{y_i}, \sigma_{y_i}^2) \quad (3)$$

with parameters $\theta_{y_i} = \{\mu_{y_i}, \sigma_{y_i}^2\}$.

3. *Introducing an extra state—pre-pattern state, and proposing a method to compute its classification and transition probabilities.*

The pre-pattern state (state 0) models the data before the query pattern. Introducing this state is mainly to meet the needs of the online pattern detection, and it is first mentioned by Ge and Smyth [7]. However, it is difficult to determine the classification probability that any data belongs to the pre-pattern state ([7] does not specify it). Here we propose a method to solve this problem. We first specify the probabilities of the data belonging to states $1 \dots K$ (assume the model has K states), and then recalculate their values according to Equation (4), where p_i^{query} is the probability of the data belonging to state i , which is the endpoint of the query pattern's i th segment:

$$p_i = \frac{P_i}{p_i^{query}} \quad i = 1 \dots K \quad . \tag{4}$$

Then compute the probability of the data belonging to the pre-pattern state:

$$p_0 = 1 - \sum_{i=1}^K p_i \quad . \tag{5}$$

Finally we normalize the probabilities:

$$p_i = \frac{p_i}{\sum_{j=0}^K p_j} \quad i = 0, 1, \dots, K \quad . \tag{6}$$

For transition probability of pre-pattern state, we set $A_{0,0} = 0$ and $A_{0,1} = 1$.

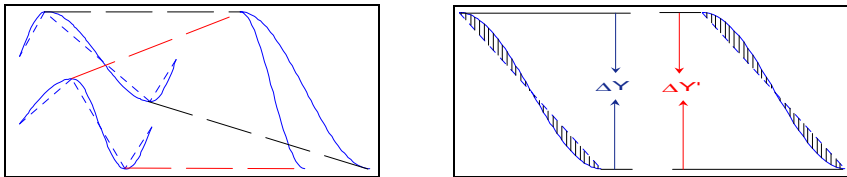


Fig. 2. (left) The second segmental observations of the two similar patterns differ considerably. (right) Two curves from the two similar patterns' second segments have equal amplitude difference, that means $\Delta Y = \Delta Y'$, and both fit their regression lines well.

4.2 Modeling the Query Pattern

First dividing Q into K segments, then estimating the parameters as follows. For transition matrix A , we set $A_{i,i+1} = 1$, and $A_{i,j} = 0$ if $j \neq i + 1$ except $A_{K,0} = 1$. Given scaling factor l , the state duration distribution $P(d_i)$ is defined as following distribution:

$$P(d_i) \propto \begin{cases} \frac{1}{\left(\lceil d_i^{query} \cdot l \rceil - \lfloor d_i^{query} / l \rfloor\right)}, & \lfloor d_i^{query} / l \rfloor \leq d_i \leq \lceil d_i^{query} \cdot l \rceil \\ 0, & \text{otherwise} \end{cases} \quad . \tag{7}$$

where d_i^{query} is the length of Q 's i th segment. The offset distribution is set to be the form as Equation (2), and its parameters μ_{R_i} and $\sigma_{R_i}^2$ are set to be R_i^{query} and $0.2R_i^{query}$ respectively, where R_i^{query} is the offset of the i th segment of Q . For the amplitude difference distribution, we usually use Equation (3) to model the pattern with μ_{y_i} being ΔY_i^{query} and $\sigma_{y_i}^2$ being $0.2\Delta Y_i^{query}$, where ΔY_i^{query} is the amplitude difference of the i th segment of the query pattern.

4.3 Online Pattern Detention

Now we can apply the improved segment semi-Markov model to detect arbitrary scaling similar patterns. Let us call the detection algorithm ISSMM for short. According to the model, for D at each time t , the algorithm first calculates the quantity $\hat{p}_i^{(t)}$ that represents the probability of the data belonging to each state i , $1 \leq i \leq K$. The recursive function for calculating $\hat{p}_i^{(t)}$ is

$$\hat{p}_i^{(t)} = \max_j \left(\max_{d_i} \left[\hat{p}_j^{t-d_i} A_{ji} \right] P(d_i | \theta_{d_i}) P(y_t - y_{t-d_i+1} | \theta_{y_i}) P(R_i | \theta_{R_i}) \right). \quad (8)$$

Then computes the probability of pre-pattern state and finally normalize the results. The state i and the time $t - d_i$ for the maximum value $\hat{p}_i^{(t)}$ are recorded in $PREV(i, t)$, then we can trace back from $PREV(i, t)$ through the table $PREV$ to get the most likely state sequence. Fig. 3(left) summarizes this procedure in pseudo-code. The algorithm chooses the state with maximum value as the state of the data. If the state is K , we declare that one similar pattern has been found, as shown in Fig. 3(right).

Note that it takes constant time to calculate $P(d_i | \theta_{d_i})$ and $P(y_t - y_{t-d_i+1} | \theta_{y_i})$; and in order to calculate $P(R_i | \theta_{R_i})$, it must take $O(d_i)$ time to calculate the offset R_i first. So according to Equation (8) we can deduce that the time complexity of ISSMM is $O(|D| \cdot |K| \cdot |Q|^2 \cdot (l^2 - 1/l^2))$, which is lower than other methods mentioned in Section 2 when $|Q| \gg |K|$.

5 Experiment Results

In this section, we perform our experiments on two real data sets (available from <http://www.cs.ucr.edu/~eamonn>), which are normalized with mean being 0. Both ISSMM and SWM_LB are used to detect the similar patterns in the same time series.

1. Results on Motion Capture data set

The Motion Capture data set was distilled from several hours of recording with Vicon (an optical motion capture system), using 124 sensors [2]. We randomly select a sequence from the data set to use as the query pattern, and then randomly choose 10 other similar sequences and 10 dissimilar sequences to form a time series acting as D , see Fig. 4. The scaling factor l is set to 1.2. Table 2 shows the comparative results.

<pre> function $s_1 s_2 \dots s_t = AMLSS(y_1 y_2 \dots y_t)$ 1. for each state i ($1 \leq i \leq K$) 2. Compute $\hat{p}_i^{(t)}, PREV(i, t)$; 3. $\hat{p}_i^{(t)} = \frac{\hat{p}_i^{(t)}}{p_i^{query}}$ 4. end for 5. $\hat{p}_0^{(t)} = 1 - \sum_1^K \hat{p}_i^{(t)}$; 6. normalize $\hat{p}_i^{(t)}$ 7. $j = \arg \max_i(\hat{p}_i^{(t)})$; 8. return; </pre>	<pre> procedure $DETECT(y_1 y_2 \dots y_t \dots)$ 1. $t = 1$; 2. $s_1 s_2 \dots s_t = AMLSS(y_1 y_2 \dots y_t)$; 3. if ($s_t == K$) 4. declare 'found'; 5. else 6. $t = t + 1$; 7. goto 2; 8. end if </pre>
--	--

Fig. 3. Pseudo-code for ISSMM algorithm. (left) Pseudo-code for AMLSS(finding the most likely state sequence $s_1 s_2 \dots s_t$ for data sequence $y_1 y_2 \dots y_t$). (right) Pseudo-code for DETECT.

Because the similar patterns differ from the other patterns considerably, the lower bounding technique helps a lot to save running time for SWM_LB, and the pruning power P reaches to 0.827. Nevertheless, ISSMM still beats SWM_LB in terms of running time, though the precision of SWM_LB is as good as that of ISSMM.

2. Results on Gun Problem data set

The Gun Problem data set comes from the video surveillance domain. The data set has two classes, and all instances were created using one female actor and one male actor in a single session. The two classes are Gun-Draw and Point, as shown in Fig. 5.

We conduct our first experiment on this data set as follows. We randomly select a sequence from the Gun-Draw class to use as the query pattern, see Fig. 6(top). From Fig. 5, we see the amplitude difference of different actors may differ a lot in the 2nd and 8th segments. So we use the following uniform distribution instead of the Gaussian distribution to model the query pattern for these two segments (see Fig. 6(top)):

$$P(\Delta Y_i) \propto \begin{cases} \frac{1}{2\Delta Y_i^{query} - 0.5\Delta Y_i^{query}}, & 0.5\Delta Y_i^{query} \leq \Delta Y_i \leq 2\Delta Y_i^{query} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Then we randomly choose 10 other Gun-Draw and 10 Point sequences, performed by the same actor performing the query pattern, to form a long time series acting as D , see Fig. 6(middle). The scaling factor l is set to 2.5. Table 3 shows the results.

The overall motions of both classes differ subtly, so the lower bounding technique is less efficient, and the pruning power P is only 0.085. ISSMM beats SWM_LB in both precision and speed.

We conduct our second experiment on the data set as follows. We use the same query pattern as the one used in the first experiment. However, we randomly pick out 10 other Gun-Draw sequences performed by both actors, where half by each, and

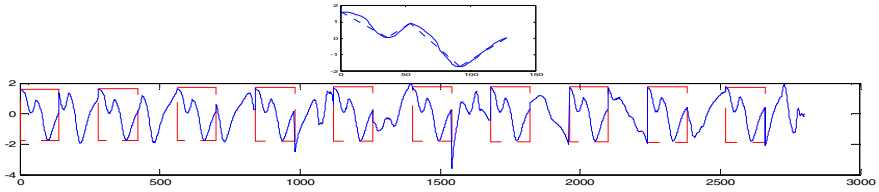


Fig. 4. The experiment data on Motion Capture data set. (top) The query pattern represented by the solid curve is divided into 4 linear segments. (bottom) The time series to be detected, and the occurrences of the similar patterns are tagged by the dashed rectangles.

Table 2. The experiment results on Motion Capture data set

	Fault detection rate	Missing detection rate	Running time (second)
ISSMM	0	0	343.33
SWM_LB	0	0	428.94

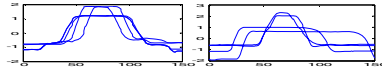


Fig. 5. (left)Some examples from Gun-Draw data. (right)Some examples from Point data.

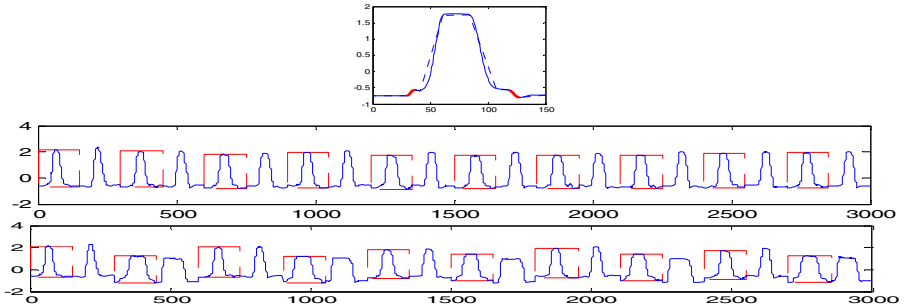


Fig. 6. The experiment data on Gun Problem data set. (top) The query pattern represented by the solid curve is divided into 9 linear segments. Especially the 2nd and 8th segments are marked in bold line. (middle) The time series to be detected in the first experiment, and the occurrences of the similar patterns are tagged by the dashed rectangles. (bottom) The time series to be detected in the second experiment, and the occurrences of the similar patterns are tagged by the dashed rectangles.

Table 3. The first experiment results on Gun Problem data set

	Fault detection rate	Missing detection rate	Running time (second)
ISSMM	0	20%	2417.8
SWM_LB	33.3%	20%	27992

Table 4. The second experiment results on Gun Problem data set

	Fault detection rate	Missing detection rate	Running time (second)
ISSMM	9%	0	2042.4
SWM_LB	25%	70%	16461

similarly we pick out 10 Point sequences. Then we concatenate these 20 sequences to form a long time series acting as D , see Fig. 6(bottom). The scaling factor l is also set to 2.5. Table 4 shows the comparative results.

Owing to arbitrary time scaling, the amplitudes of Gun-Draw patterns performed by different actors differ sharply after normalization, but their shapes are similar. SWM_LB miss all the Gun-Draw patterns performed by the other actor who is different from the one performing the query pattern, while ISSMM can detect all of them.

6 Conclusion and Future Work

In this paper, based on the existing segmental semi-Markov model, we modify it in several aspects. The improved model is applied to online detect arbitrary scaling similar patterns. And it is successfully demonstrated on real data sets.

In future work, we will consider using the model in the noisier environment to widen the application scope of the model.

References

1. Meek, C., Birmingham, W.: The Dangers of Parsimony in Query-By-Humming Applications. In Proc. of International Symposium on Music Information Retrieval. Baltimore, USA (2003) 51–56
2. Keogh, E., Palpanas, T., Zordan, V.B., Gunopulos, D., Cardle, M.: Indexing Large Human-Motion Databases. In Proc. of 30th International Conference on Very Large Data Bases. Toronto, Canada (2004) 780–791
3. Mandelbrot, B.: Fractals and Scaling in Finance: Discontinuity, Concentration, Risk. New York :Springer-Verlag, (2000)
4. Fu, A.W., Keogh, E., Lau, L.Y.H., Ratanamahatana, C.A.: Scaling and Time Warping in Time Series Querying. In Proc. of the 31st VLDB. Trondheim, (2005) 649-660
5. Ferguson, J.D.: Variable Duration Models for Speech. In Proc. Symposium on the Application of Hidden Markov Models to Text and Speech. (1980) 143-179
6. Ostendorf, M., Digalakis, V.V., Kimball, O.A.: From HMM's to Segment Models: a Unified View of Stochastic Modeling for Speech Recognition. IEEE Transactions on Speech and Audio Processing. (1996) 4(5):360-378
7. Ge, X.P., Smyth, P.: Deformable Markov Model Templates for Time-Series Pattern Matching. In Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discover and Data Mining. Boston, USA (2000) 81-90
8. Ge, X.P., Smyth, P.: Hidden Markov Models for Endpoint Detection in Plasma Etch Processes. Technical Report, Dep. Of ICS, UCI. Available from <http://citeser.nj.nec.com/>
9. Jia, S. Qian, Y.T., Dai, G.: An Advance Segmental Semi-markov Model Based Online Series Pattern Detection. In Proc. of the 17th International Conference on Pattern Recognition. Vol.3. Cambridge, UK. (2004) 634–637

10. Argyros, T., Ermopoulos, C.: Efficient Subsequence Matching in Time Series Databases under Time and Amplitude Transformations. In Proc. Of 3rd ICDM. USA (2003) 481-484
11. Rabiner, L., Rosenberg, A., Levinson, S.: Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition. IEEE Transactions on Acoustics, Speech and Signal Processing. (1978) 26(6):575-582
12. Baum, L.E., Petrie, T.: Statistical Inference for Probabilistic Functions of Finite State Markov Chains. Annals of Mathematical Statistics. (1966) 37(6):1554-1563

Diagnosis of Inverter Faults in PMSM DTC Drive Using Time-Series Data Mining Technique

Dan Sun, Jun Meng, and Zongyuan He

College of Electrical Engineering, Zhejiang University
Hangzhou, Zhejiang, 310027, China
sundan@zju.edu.cn

Abstract. This paper investigates a Time-Series Data Mining (TSDM) Technique based fault diagnostic method for short-switch and open-phase faults in a standard 6-switch inverter fed permanent magnet synchronous motor (PMSM) direct torque control (DTC) drive system. For diagnosing the operating condition of an inverter, the reconstructed phase space (RPS) theory is applied to obtain the special feature consisting in the trajectories of phase currents for healthy and faulty operating conditions. The fuzzy C-mean (FCM) algorithm is used to build a fuzzy membership function, an FCM based ANFIS (FCM-ANFIS) is designed to classify different fault patterns. The proposed method has been studied by simulation using MATLAB; which proves that different operating conditions of PMSM DTC drive can be discovered clearly without background knowledge.

1 Introduction

Standard 6-switch 3-phase voltage source inverter is currently in common use in a majority of AC variable-speed drives, especially for permanent magnet synchronous motor (PMSM) direct torque control (DTC) drives [1]. In the last few decades, there have been efforts to study and diagnose faults in AC motor drives fed by inverters in order to avoid unplanned standstill or breakdown, and further, to make possible to run an emergency protection or tolerant operation in case of faults.

For a reliable fault diagnosis, it is extremely required in detecting and classifying fault elements. Various approaches to the fault detection and diagnosis of the inverter have been proposed. In terms of the signature to identify fault elements in an inverter, the phase voltages [2], d-q phase currents [3] and mean phase currents [4] are often used. Some researches introduced detection and diagnosis methods such as model-based techniques [5], intelligent systems [6] and so on.

Recently, a novel method for fault monitoring and diagnosis based on Time-Series Data Mining (TSDM) technique has evolved in [7] for induction motors. This technique, based on a time delay embedding process, can reveal and extract hidden and inherent patterns (characteristics) in the voltages and currents used for fault identification. This technique can be utilized effectively to detect and distinguish the fault patterns in motor drives by analyzing the extracted fault signatures of these faults compared with the healthy performance signatures. However, only motor side fault diagnosis using TSDM technique has been proposed at present and, no literature

can be found for TSDM based fault diagnosis of the inverter in motor drive systems. [8] introduced a C-ANFIS based clustering method for inverter side fault classification with the mean current vector for fault diagnosis, however, it needs more known knowledge about the drive system to contrast with TSDM technique.

In this work, TSDM technique is employed to diagnose short-switch and open-phase faults in the inverter fed PMSM DTC system, and a Fuzzy C-mean clustering method based neural network is used to classify the fault modes. The proposed method is studied by simulation using MATLAB, which convincingly demonstrate the soundness and robustness of the RPS based FCM-ANFIS method for reliable and efficient fault diagnosis.

2 Characteristic Discovery of Inverter Faults in PMSM DTC

A PMSM DTC drive system is a mixture of electrical, electronic, and mechanical components as shown in Fig.1.

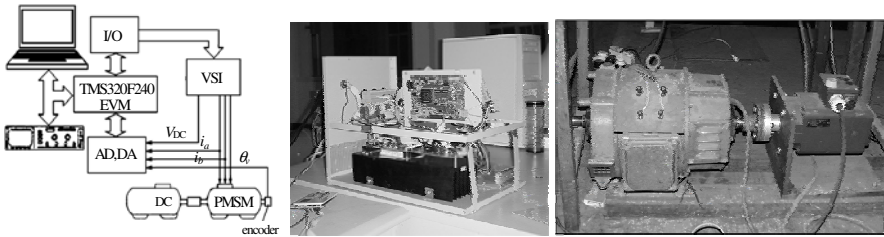


Fig. 1. Hardware setup of PMSM DTC drive, constituted by rectifier, inverter, PMSM and a DC motor served as the load

Faults might occur in different parts of the drive system. In this paper, the investigated faults are mainly in the inverter and the controller, these are, breakdown of switching device, missing drive signal for PWM and so on. However, the often happened short-switch and open-phase fault in phase A are considered only in this study, and the other faults can be analyzed similarly.

The modeled and simulated PMSM DTC drive parameters are: number of pole pairs: $P_n=3$, stator resistance: $R_s=0.56\Omega$, permanent magnet flux linkage: $\psi_{PM}=0.2Wb$, DC-link voltage: $u_{DC}=300V$, rated current: $I=15.8A$ and rated speed: $\omega_b=2000r/min$; waveform of phase current i_b under healthy, short-switch and open-phase fault occurring at $t=0.06s$ are shown in Fig.2.

The data consist in these waveforms can be taken as the time series data of the phase current. Therefore, a TSDM theory based FCM-ANFIS technique can be used to deal with these data and discover different patterns from them. The data sets used for diagnosis are selected in a short time ($t=0.06s$ to $t=0.09s$) as shown in Fig.2, due to that in the real time system operation, faults can not be allowed to last for a long time.

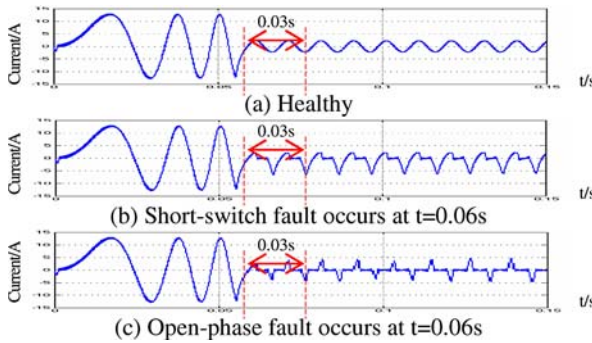


Fig. 2. Phase current i_b of PMSM DTC drive system in different operation condition

3 Time-Series Data Mining Based FCM-ANFIS Technique

The TSDM based FCM-ANFIS technique proposed in this study, is used to deal with the healthy and faulty time series data with the reconstructed phase space (RPS) theory firstly, and then with the FCM method to predict the distribution of fuzzy membership functions by universe of discourse partition, which will improve the speed and accuracy of ANFIS training process. Finally, the designed FCM-ANFIS is utilized to classify different drive operating mode.

3.1 Time-Series Data Mining Technique

TSDM technique is based on the time-delay embedding process, which transforms the time series data (time-domain waveform) into a different processing state space, named as reconstructed phase space (RPS) [9].

Given the time series data as:

$$I = [i_n], n = 1, \dots, N \tag{1}$$

where n is the time index, the RPS matrix I is defined by its row vectors whose elements are time-lagged versions of the original time series data and can be expressed as follows:

$$I = \begin{bmatrix} i_{1+(d-t)\tau} & \cdots & i_{1+\tau} & i_1 \\ i_{2+(d-t)\tau} & \cdots & i_{2+\tau} & i_2 \\ \vdots & & \ddots & \\ i_N & \cdots & i_{N+(d-2)\tau} & i_{N+(d-1)\tau} \end{bmatrix} \tag{2}$$

where i_n is the original time series data, N is the number of observations (number of data samples in a waveform), d is the embedding dimension and τ is the time lag. It is important to point out that each row vector of I is a single point in the RPS. Here, the time lag τ is determined by using the first minimum of the auto mutual information function, from which the embedding dimension d can be estimated by using the false nearest neighbor method [10].

It had been proved by Takens that the RPS is topologically equivalent to the original state space of the system and, therefore the full dynamics of the original system are accessible in this RPS. Based on this characteristic, the hidden pattern, which is captured from the RPS, has to be residing in the original system, which cannot be easily detected from a frequency spectral analysis, or other examinations of the time domain profile of the physical phenomenon’s waveform under consideration.

3.2 Fuzzy C-Mean Algorithm

Fuzzy c-mean (FCM) is an unsupervised clustering algorithm that has been applied successfully to a number of problems involving feature analysis, clustering and classifier design.

The FCM minimizes an objective function J_m , which is the weighted sum of squared errors within groups and is defined as:

$$J_m(U, V; X) = \sum_{k=1}^n \sum_{i=1}^c u_{ik}^m \|x_k - v_i\|_A^2, 1 < M < \infty \tag{3}$$

where $V=(v_1, v_2, \dots, v_c)$ is a vector of unknown cluster prototype center $v_i \in R^p$. The value of u_{ik} represents the grade of membership of data point x_k in set $X=(x_1, x_2, \dots, x_n)$ to the i_{th} cluster. The inner product defined by a norm matrix A defines a measure of similarity between a data point and the cluster prototypes. A nondegenerate fuzzy c -partition of X is conveniently represented by a matrix $U=[u_{ik}]$.

It has been shown that if $\|x_k - v_k\| > 0$ for all i and k , then (U, V) may minimize J_m only, when $m > 1$ and

$$v_i = \frac{\sum_{k=1}^n (u_{ik})^m x_k}{\sum_{k=1}^n (u_{ik})^m} \quad \text{for } 1 \leq i \leq c; \tag{4}$$

$$u_{ik} = 1 / \sum_{j=1}^c (\|x_k - v_i\|_A^2 / u_{ik}^m \|x_k - v_j\|_A^2)^{1/m-1} \quad \text{for } 1 \leq i \leq c, 1 \leq k \leq n; \tag{5}$$

Among others, J_m can be minimized by the Picard iteration approach. This method minimizes J_m by initializing the matrix U randomly (or predefined) and computing the cluster prototypes (Eq. (4)) and the membership values (Eq. (5)) after each iteration. The iteration is terminated when reaching a stable condition. This can be defined, for example, when the changes in the cluster centers or the membership values at two successive iteration steps is smaller than a predefined threshold value. The FCM algorithm always converges to a local minimum or a saddle point. A different initial guess of u_{ij} may lead to a different local minimum. Finally, to assign each data point to a specific cluster, the defuzzification is necessary, e.g., by attaching a data point to a cluster, for which the value of the membership is maximal.

3.3 FCM-ANFIS Technique

The grid partition based ANFIS has a fuzzy rules number exponential increasing problem when the inputs number is increased. In this paper, such problem is solved

by adopting the FCM clustering strategy. The data clustering is to identify initial parameters and structure based on the scatter partition. Consequently, the technique can be used to reduce the dimension of models as well as training time since the number of fuzzy rules equals the number of membership functions regardless of the dimension of inputs. This modified ANFIS approach is named as FCM-ANFIS.

4 Implementation of TSDM Based FCM-ANFIS Method

The current waveforms shown in Fig.2 include features of the operating condition of the drive. These data sets are arranged to a single time series data set for diagnosis. After generating the RPS by using these time series data, next step is to categorize the faults into different classes (healthy, short-switch and open-phase) using FCM-ANFIS method. In summary, there are several stages in the process; which can be depicted as a block diagram, shown in Fig. 3. In this figure, the first stage is the data sensing, followed by a RPS part to extract different features and, then, an ANFIS-FCM identification method is developed for faults classification.

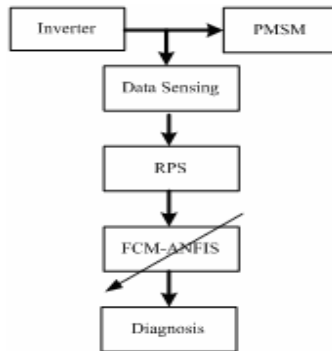


Fig. 3. Block diagram of stages in diagnosis process

4.1 RPS Theory Based Characteristic Discovery and Clustering Analysis

According to RPS theory, $\tau=1/4T_s$ and $d =2$ are selected. Three trajectories for healthy, open-phase and short-switch faults are obtained in Figs.4-6, separately, from which it can be seen that these trajectories are intuitionistic and totally different, the trajectory in Fig. 4(a) is an ellipse, a cross in Fig.5 (a) and a triangle in Fig.6 (a).

The clustering method is then used in this study to cluster three operation conditions for predicting the output of the FCM-ANFIS, different results are shown in Figs.4-6 as well. It can be seen that 4 cluster centers is obtained when the PMSM is healthy, 3 centers is obtained when short-switch fault happened, and only 1 center is got when open-phase fault happened. The numbers of clustering center are all different when the operating condition of the PMSM DTC drive is different.

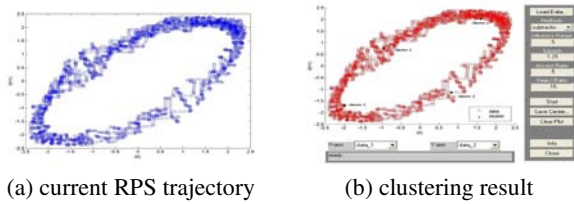


Fig. 4. Healthy operating condition

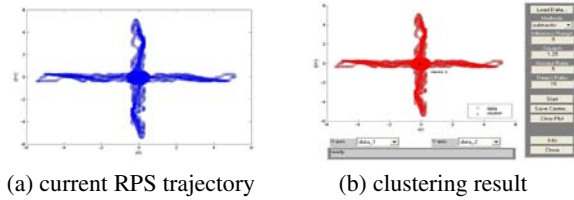


Fig. 5. Short-switch fault operating condition

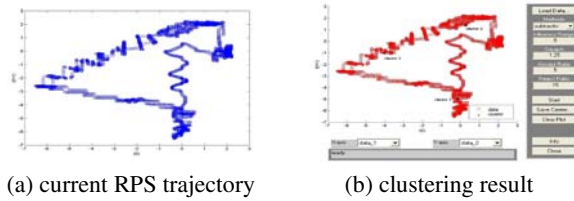


Fig. 6. Open-phase fault operating condition

4.2 FCM-ANFIS Technique

ANFIS has two functions to generate fuzzy inference system: `genfis1` and `genfis2`; `genfis1` is based on grid partition, which is not good for real time data, `genfis2` is based sub clustering method, where the radial of clustering can not be adjusted, so that the number of membership function has to be increased in order to depict complicated trajectory. In order to improve the generation of ANFIS, a program named as “`genfis4`” was written with the function of generating an initial FIS by using FCM method. The MATLAB program is coded as follows:

```
genfis4
function fismat =
    genfis4(Xin,Xout,cluster_n,xBounds,options)
    %GENFIS4 Generate an FIS structure from data using
    fcm clustering.
    [numData,numInp] = size(Xin);
    [numData2,numOutp] = size(Xout);
```

```

.....
data=[Xin Xout];
[center,U,obj_fcn]=fcm(data,cluster_n);
.....
%relation between U and sigmas
U1=U';
U2=sum(U1);
U3=U2./numData;
U4=U3.*20;
maxX=max(Xin);
minX=min(Xin);
sigmas=(0.5.*(maxX-minX))/sqrt(8.0);

% FIS type (FCM-ANFIS)
.....
% Set the input membership function types
.....
% Set the input membership function parameters
.....
% Set the membership function pointers in the rule
list
.....
% Set the antecedent operators and rule weights in
the rule
    fismat.rule(j).weight=1;
    fismat.rule(j).connection=1;
end

```

A FCM-ANFIS model is designed for motor operating mode classification as shown in Fig.7.

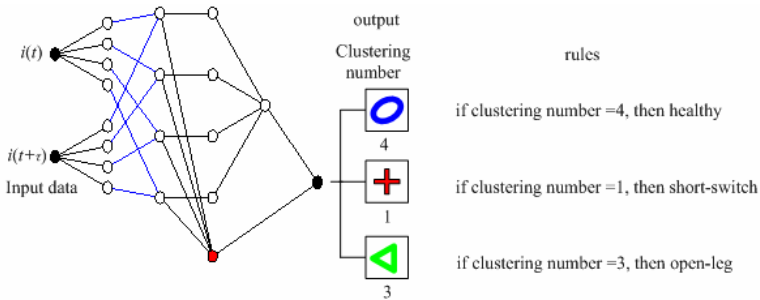


Fig. 7. FCM-ANFIS model structure and inference rules. The FCM-ANFIS has two input unit, which are selected as $i(t)$ and $i(t+\tau)$, and the output is set up as the clustering number, 4, 1, 3, which denotes healthy, short-switch and open-phase, respectively.

5 Conclusion

In this work, a novel fault diagnosis technique was proposed to directly discover the inverter faults of a PMSM DTC drive system from real-time time series data. A

reconstructed phase space method was employed by analyzing only a single phase current time series data under both healthy and faulty conditions without background knowledge of the drive system. It was found that some distinguishable fault patterns can be discovered and the clustering numbers are also totally different. Consequently, a hybrid FCM based ANFIS model could be designed and utilized to diagnose different inverter faults and the inference rules could be given as well to classify different operating conditions of PMSM DTC drive. Besides, the technique is easy to be implemented by hardware.

Acknowledgement

The authors would like to thank the National Natural Science Foundation of China for the financial support (Project No. 50507017, 60574079) and the Zhejiang Provincial Natural Science Foundation of China for the financial support (Project No. 601112).

References

1. Zhong, L., Rahman. F.: Analysis of direct torque control in permanent magnet synchronous motor drives. *IEEE Transactions on Power Electronics*, Vol. 12, Issue 3, (1997) 528–536
2. Riheira, R.L., Jacobina, C.B., Silva, E. R. C.: Fault detection of open-switch damage in voltage-fed PWM motor drive systems. *IEEE Transactions on Power Electronics*, vol. 18, Issue 2, (2003) 587–593
3. Ye, Z. Wu, B.: Simulation of electrical faults of three phase induction motor drive system. *Power Electronics Specialist Conference*, vol. I. (2001) 75–80
4. Diallo, D.; Benbouzid, M.E.H.; Hamad, D. Fault Detection and Diagnosis in an Induction Machine Drive: A Pattern Recognition Approach Based on Concordia Stator Mean Current Vector, *IEEE Transactions on Energy Conversion*, Vol. 20, Issue 3, (2005) 512–519
5. Klima, J. Time and frequency domain analysis of fault-tolerant space vector PWM VSI-fed induction motor drive. *IEE Proceedings-Electric Power Applications*, Vol. 152, Issue 4, (2005) 765–774
6. Peugeot, R. Courtine, S. and Rognon, J.P.: Fault Detection and Isolation on a PWM Inverter by Knowledge-Based Model. *IEEE Transaction on Industry Application*, vol. 34, Issue 6, (1998) 1318–1326
7. Bangun, J., Povinelli, R., herdash, N. and Brown, R.: Diagnostics of Eccentricities and Bar/End-Ring Connector Breakages in Polyphase Induction Motors Through a Combination of Time-Series Data Mining and Time Stepping Coupled-State-Space Techniques. *IEEE Transactions on Industry Applications*, Vol. 39, Issue 4, (2003) 1005–1013
8. Park Jang-Hwan, Kim Dong-Hwa, Kim Sung-Suk: C-ANFIS based fault diagnosis for voltage-fed PWM motor drive systems. *IEEE Annual Meeting of the Fuzzy Information*, Vol. 1, (2004) 379–383
9. Abarbaue, H. D. I.: *Analysis of Observed Chaotic Data*. New York: Springer, (1996)
10. Kanb, H., Schreiber, T.: *Nonlinear Time Series Analysis*. Cambridge. Cambridge University Press (1997)

Applications of Data Mining Time Series to Power Systems Disturbance Analysis^{*}

Jun Meng, Dan Sun, and Zhiyong Li

College of Electrical Engineering,
Zhejiang University
Hangzhou, Zhejiang, 310027, China
junmeng@zju.edu.cn

Abstract. In the last decade there has been an explosion of interest in mining time series data, introducing new algorithms to index, classify, cluster and segment time series. In this paper we use fractal theory and reconstructed phase space to analysis the special time series –power systems disturbance signal. After analyzing the feasible method of time series data mining-fractal theory and reconstructed phase space, which is used for the analysis of power disturbance signals. Eight common happed disturbances are considered in this paper, the simulation results show that fractal method can detect the transient disturbance, accurately locate the time when it occurred. Reconstructed phase space can classify the different type of disturbance. It is concluded that two methods are all efficient and intuitionistic for detection and diagnosis the fault of power system, which presents a new concept for power disturbance analysis.

1 Introduction

Popular feature extraction techniques for time series include the Discrete Wavelet Transform (DWT) and the Discrete Fourier Transform (DFT). The signal is projected into the frequency domain (DFT) or a tiling of the time frequency plane (DWT). [1]. Fractal theory [2-3] based dimension measurement and reconstructed phase space present new route for linear time series analysis [4-6]. Fractal theory was applied firstly on fractal image compress and computer image generation and processing, the application area has been extended to earthquake forecast data analysis, market economy prognosticate and share market analysis. It has been used in power system recently. [7-9].

The initial purpose of reconstructed phase space was to recover the strange attractor of chaos system in high-dimension phase space, and has been developed as an efficient method for analyzing complicated time series. [10-11].

Fractal dimension can reflect the characteristic of measured signal to a certain extent, and characteristic extraction is very important to identification of signal, therefore, fractal theory can be used to analyze the complexity of system signal and obtain valuable information in measured time series for analyzing different working

^{*} The work was supported by National Natural Science Foundation of China. (No. 60574079 and 50507017), Supported by Zhejiang Provincial Natural Science Foundation of China. (No. 601112)

condition of the system and provide the basis for faulty situation. However, there are not many studies of using fractal theory to pre-process the information in complicated system available nowadays.

Power system is a typical complicated system, it is found recently that these are chaos phenomena in power system under certain circumstance [12-13]; power disturbance signal shows the power quality of the consumer side, with the rising requirement of the power quality, analysis and identification of different power disturbance signal became the basic of obtaining disturbance source and the precondition of determining improved measure.

There are many different types of power disturbances, and each has different features considering applied time, variation and spectrum components. The analysis is more complex if two or more disturbances occur at the same time. Mathematical transform based method are mainly used to analyze the power disturbance nowadays, FFT is the most popular on among them, which can extract the spectrum characteristics efficiently, however, it can not sufficiently describe the time-vary unstable signal because of its part contradiction between time-frequency domain. Wavelet transform has good time-frequency feature so that it can well analyze weak and chop signal [14-16], however, the efficiency and accuracy of wavelet transform depend on the selection of wavelet mother function, and it should be implemented in different dimension, so that it has heavy computational burden and it is sensitive to the noise. Moreover, there are other methods such as continuous wavelet transform and short-time FFT combined S transformation, [17,18], d-q transformation[19]、 Hilbert-Huang transform used to extract the signal characteristic. Mathematical transform based methods has contradiction of accuracy and computational burden, and time-frequency based methods often has frequency band aliasing, leak effect and not enough frequency partition. [20].

This paper introduced fractal and reconstructed phase space to the detection and analyzing of power disturbance. The simulation results show that fractal can detect the occurrence of power disturbance, accurately locate its occurrence time and ending time; reconstructed phase space can classify different power disturbance signal by describing the power signal time series; test results show that there two methods are efficiently and intuitionistic with light computational burden.

2 Basic Theories

2.1 Fractal Theory

According to Barnsley, fractal dimension is an index used to describe the thickness of a set in its space. The classic definition of fractal dimension is capacity dimension, also called as box dimension: suppose set F is a limitary non-empty subset, $N(\varepsilon)$ is the minimum number of the minimum set has a diameter of ε to cover F , then

$$D_B = \lim_{\varepsilon \rightarrow 0} \frac{\ln N(\varepsilon)}{\ln(1/\varepsilon)} \quad (1)$$

Capacity dimension considered only the number needed, but not the number of fractal set element in the minimum subset. Substituting $N(\epsilon)$ in (1) by information equation $I(\epsilon)$, the definition of information dimension is obtained as:

$$D_I = \lim_{\epsilon \rightarrow 0} \frac{\ln I(\epsilon)}{\ln(1/\epsilon)} \tag{2}$$

Where

$$I(\epsilon) = \sum_{i=1}^{N(\epsilon)} -P(\epsilon, i) \ln(P(\epsilon, i)) \tag{3}$$

$P(\epsilon, i)$ is the probability of the element in the set belongs to subset i .

Another definition of fractal dimension is correlation dimension:

$$D_C = \lim_{\epsilon \rightarrow 0} \frac{\ln C(\epsilon)}{\ln \epsilon} \tag{4}$$

Where $C(\epsilon)$ is correlation function, defined as:

$$C(\epsilon) = \lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{i,j=1}^N H(\epsilon - \|x_i - x_j\|) \tag{5}$$

H is Heaviside function.

correlation dimension is easy to be implemented in computer, as to power signal time series, disturbance in the signal can be detected by calculating a extended correlation dimension. fractal number is defined as:

$$F_m = \frac{\sqrt{\sum_{i=1}^{N-l} \|x_{i+k} - x_i\|^2}}{\sqrt{\sum_{i=1}^{N-l} \|x_{i+l} - x_i\|^2}} \tag{6}$$

F_m is the m^{th} subset fractal number of the set cover N data points, k and l is sampling interval for $l > k$. Test indicates that l is selected as the non integral time of k , and number of subset is also the non integral time of then number of signal period.

2.2 Reconstructed Phase Space Theory

For the single variable time series x_1, x_2, \dots, x_N , a time delay parameter τ and a embedded dimension m can be introduced to construct a m -dimension phase space:

$$X_i = [x_i, x_{i+\tau}, \dots, x_{i+(m-1)\tau}]^T \tag{7}$$

Where $i = 1, 2, \dots, L, L = N - (m - 1)\tau$

From (7), the reconstructed phase space trajectory matrix is obtained:

$$\begin{aligned} X_1 &= [x_1, x_{1+\tau}, \dots, x_{1+(m-1)\tau}]^T \\ X_2 &= [x_2, x_{2+\tau}, \dots, x_{2+(m-1)\tau}]^T \\ &\vdots \\ X_L &= [x_L, x_{L+\tau}, \dots, x_{L+(m-1)\tau}]^T \end{aligned} \tag{8}$$

For the periodical power signal, $x(t) - x(t + \tau)$ is selected to reconstruct signal trajectory. Delay time τ is a #n important parameter in reconstructed phase space, the extended degree of phase trajectory along diagonal is different if τ is different. Because there is no literature available discussing the selection of τ , therefore, 4 different vale of τ are selected here to compare the reconstructed trajectory of different disturbances with the standard sine wave.

3 Fractal Theory Based Time Series Analysis

A series data can be obtain by sampling a variable or objection in a fixed time interval in a system, that data series is called time series, which is expressed as:

$$\begin{aligned} Z(t) &= \{x_1(t), x_2(t), \dots, x_n(t)\} \\ t &= t_0 + p\Delta t, \quad p = 0, 1, \dots, N \end{aligned}$$

Where, n is the number of the observed variables; t is time; t0is starting time for observation; Δt is sampling time interval; N is the length of the data; x_1, x_2, \dots, x_n might be the vale of temperature, pressure, speed, etc.

3.1 Fractal Dimension of Time Series

Dimensionality curse and dimensionality reduction are two issues that have retained high interest for data mining, machine learning, multimedia indexing, and clustering. [18]

3.1.1 Higuchi Method [21]

Let us have the time series $X(i)(i = 1, \dots, N)$ Then, the value $L_m(k)$ can be calculated for $m = 1, \dots, k$

$$L_m(k) = \frac{1}{k} \left\{ \left(\sum_{i=1}^{\lfloor \frac{N-m}{k} \rfloor} |X(m+ik) - X(m+(i-1)k)| \right) \frac{N-1}{k \lfloor \frac{N-m}{k} \rfloor} \right\} \tag{9}$$

The averaging of $L_m(k)$ will give:

$$L(k) = \frac{1}{k} \sum_{m=1}^k L_m(k) \tag{10}$$

If the curve has a fractal property:

$$L(k) \sim k^{-D} \tag{11}$$

where D is a fractal dimension of the curve.

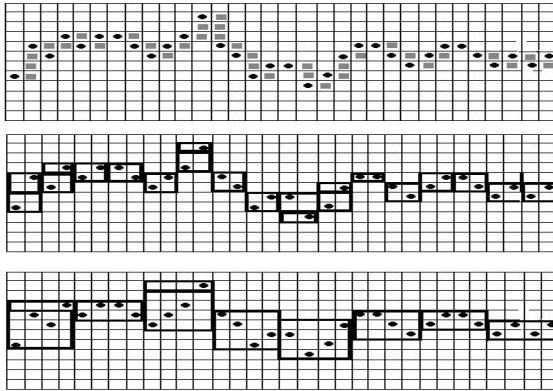


Fig. 1. Using Higuchi method calculate fractal dimension of time series [22]

3.2 Correlation Dimension of Time Series

Correlative dimension is always used in reconstructed phase space, F. Takens indicated the mathematical method of reconstructed phase space in his paper “Detecting strange attractors in turbulence”.

there is a time series $\{x(t_i)\}$, $i = 1, 2, \dots, N$, if \mathcal{E} is the distance of two internal points, then correlative $C(\mathcal{E})$ is:

$$D_R = \lim_{\mathcal{E} \rightarrow 0} \frac{\ln C(\mathcal{E})}{\ln(1/\mathcal{E})} \tag{12}$$

Where

$$C(\mathcal{E}) = \frac{1}{N^2} \sum_{i,j=1}^N H(\mathcal{E} - |x_i - x_j|) = \sum_{i=1}^N p_i^2 \tag{13}$$

Where $H(s)$ is Heaviside function[23],

- $H(s)=1$, when $S>0$;
- $H(s)=0$, when $S<0$.

Correlative dimension is obtained as follows:

From (12) it can be seen that, DR value is the slope of double logarithm (log-log plot of) $C(\mathcal{E})$ versus \mathcal{E} . For a given \mathcal{E} , a $C(\mathcal{E})$ can be estimated. therefore, $C(\mathcal{E})$ is obtained by calculating the distance of internal points $r_{i,j} = \|x_i - x_j\|$, and counting the number $N_r(\mathcal{E})$ of satisfying $r_{i,j} < \mathcal{E}$ ($i, j = 1, 2, \dots, N$), and then, $C(\mathcal{E}) = N_r(\mathcal{E}) / N^2$. Finally, DR value is obtained by calculating double logarithm of data composed by \mathcal{E}_k and $C(\mathcal{E}_k)$, following a least square linear fit.

$\mathcal{E}_0, \mathcal{E}_0^2, \mathcal{E}_0^3, \dots, \mathcal{E}_0^k$ ($\mathcal{E}_0 > 0, k > 0$) is used to express a series \mathcal{E} in the real time operation. For any time series k_{\min} and k_{\max} can be obtained to make $\mathcal{E}_0^{k_{\min}}$ to be the distance of minimum measured time series, $\mathcal{E}_0^{k_{\max}}$ to be the maximum one (as Fig 2).

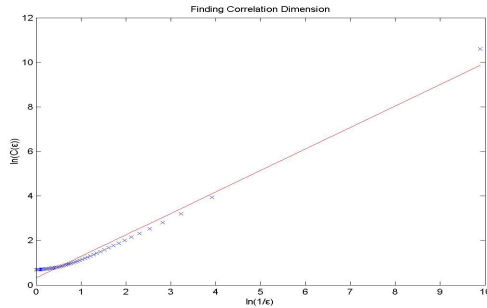


Fig. 2. The results of calculating of DR

3.3 Identification of Time Series Signal Using Fractal Dimension

Fractal theory offers new ration guideline for signal analysis, that is to discribe the complexity of signal using its special dimension theory, and for the identii#fication porpous. It can be said that sampling fractal number is an index to weight the chaos degree of system signal in physical sense, when the signal is chaos, the sampling fractal dimension is big, and vice versa. The fractal number Dalso shows the complexity of signal, that is the frquency components. when D is big, the part fluctuation is big, correlation between neibour point of signal is small, which means there are more high frequent components, and and vice versa. There are many manipulate variable and disturbances in complicate system, observing the manipulate variable data waveform sampled from the complicate syste, it is not dificult to find abnormity in the waveform, and the edge is not smooth. After analyzing the waveform, it can be seen that it is difficult to distinguish different dsignal using traditional methods, and other normal signal analyzing methods are not working well either. Considering the natual fractal discription of fractal theroy, combinig fractal

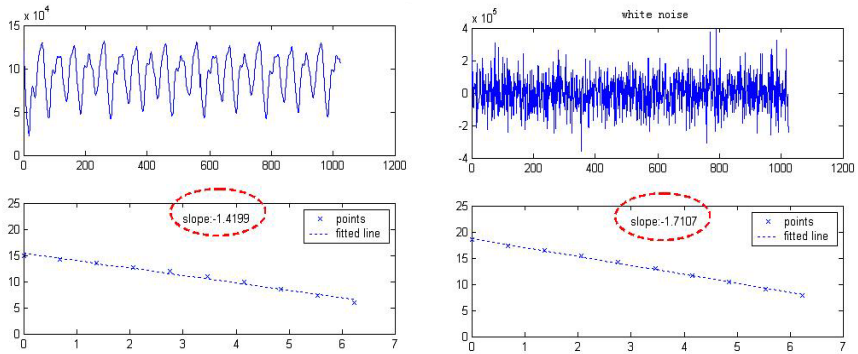


Fig. 3. The different time series' fractal dimension

and signal analysis, using fractal number theory, classifying different signal by the complexity of signal, and finally to identify different signal, as shown in Fig 3.

4 Application Case

4.1 Signal Need to Be Detected

According to IEEE 22th [24] standard, there are ten power disturbance, eight of them are considered in this paper: voltage sags, voltage swells, voltage interruptions, voltage spikes or transients, voltage fluctuations or flickers, voltage notches, harmonics and inter harmonics.

4.2 Study Results of Fractal Dimension

The test results are shown in Figs.4-11, which are eight power disturbance simulated waveform and its fractal number. Simulation time is 200ms, sampling frequency is 4.8 kHz, and number of subset is 48.

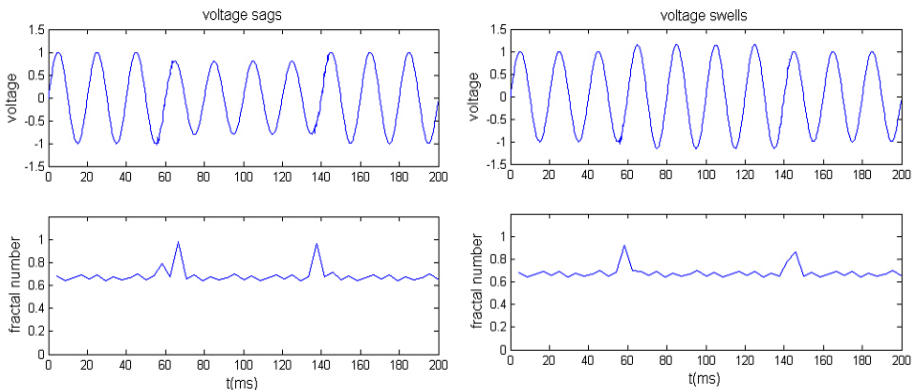


Fig. 4. and Fig. 5. Waveform and fractal number of voltage sags and voltage swells

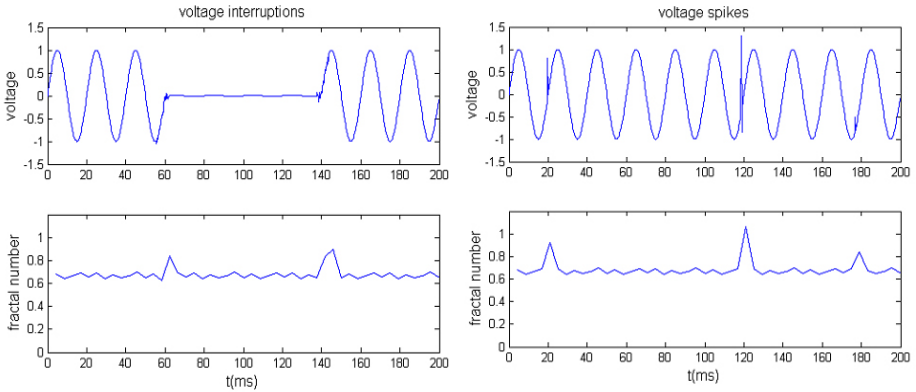


Fig. 6. and Fig. 7. Waveform and fractal number of voltage interruptions and voltage spikes

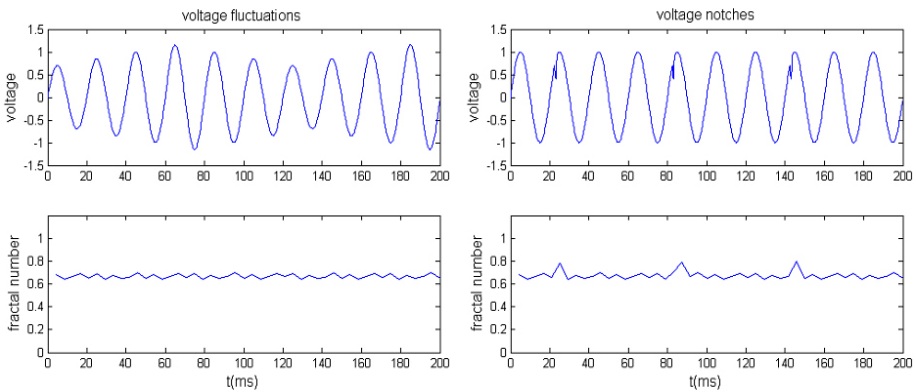


Fig. 8. and Fig. 9. Waveform and fractal number of voltage fluctuations and voltage notches

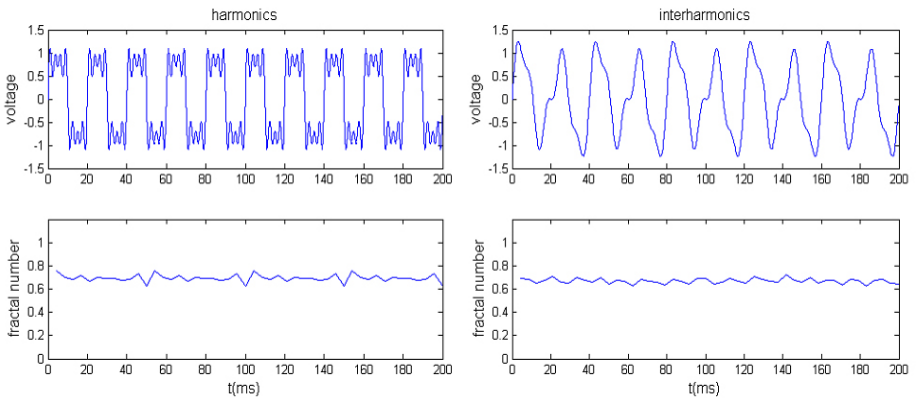


Fig. 10. and Fig. 11. Waveform and fractal number of voltage harmonics and voltage interharmonics

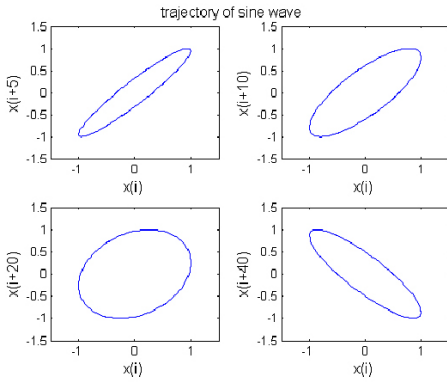


Fig. 12. Trajectory of sine wave

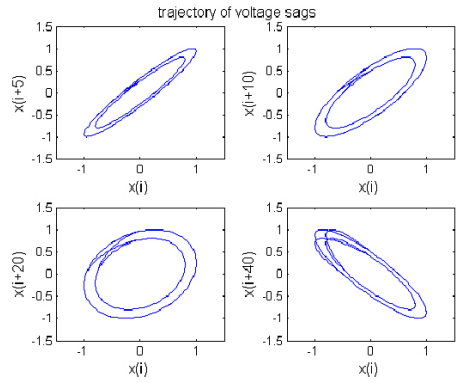


Fig. 13. Trajectory of voltage sags

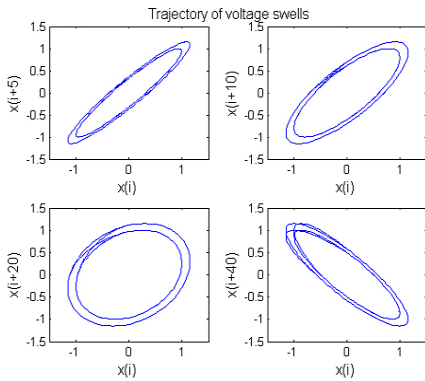


Fig. 14. Trajectory of voltage swells

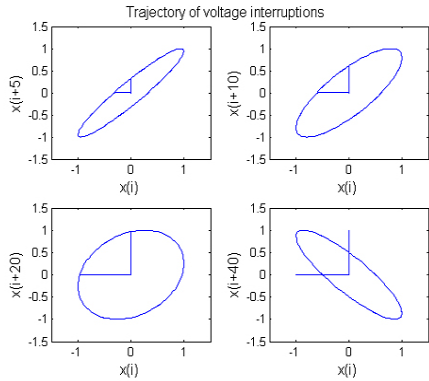


Fig. 15. Trajectory of voltage interruptions

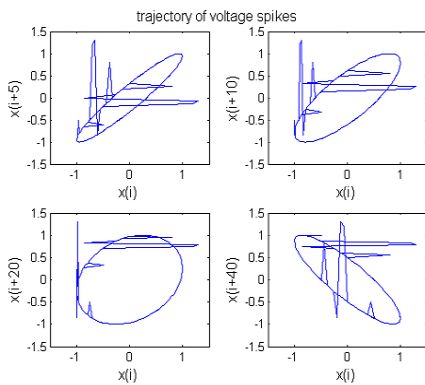


Fig. 16. Trajectory of voltage spikes

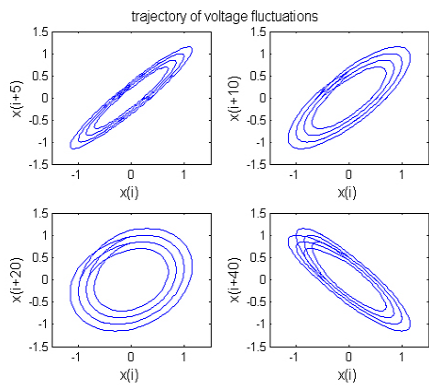


Fig. 17. Trajectory of voltage fluctuations

4.3 Study Results of Reconstructed Phase Space

The waveforms of disturbances are the same as fractal based study results, simulation time is 200ms, sampling frequency is 4.8kHz, delay time is selected as 5, 10, 20, and 40, respectively. Fig.12 is the trajectory of standard 50Hzsine wave, Figs.13-20 are trajectories of disturbance.

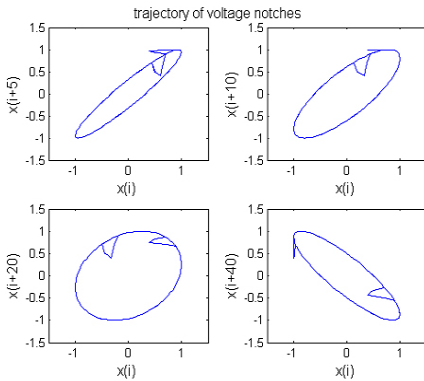


Fig. 18. Trajectory of voltage notches

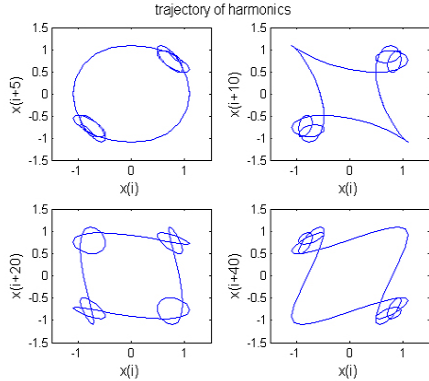


Fig. 19. Trajectory of voltage harmonics

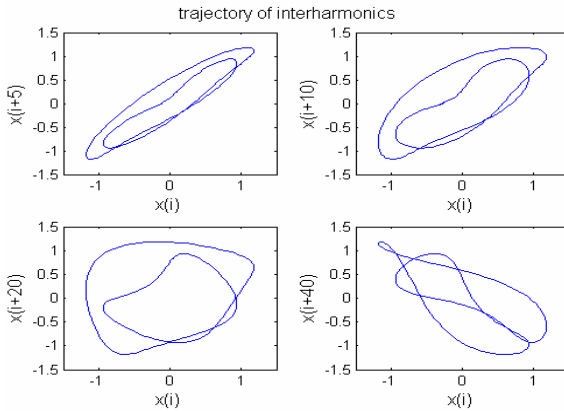


Fig. 20. Trajectory of voltage interharmonics

5 Conclusions

This paper introduced fractal theory and reconstructed phase space to power disturbance analysis, firstly attempted to analyze the power signal with disturbance as a nonlinear time series. The preliminary test results indicate that both methods can detect disturbance signal efficiently with small computational burden.

- (1) Fractal theory can accurately locate different disturbance occurrence time and the lasting time, which is significant to judge the power quality of an area.
- (2) Using reconstructed phase space to describe trajectory of disturbance signal used looking for chaos attractor in nonlinear dynamics for reference. Study results verified that different disturbance with variable features can be detected efficiently using reconstructed phase space method, and automatically identify and classify disturbances if combined this method with artificial method such as expert system, etc..
- (3) With the development of intelligent inspect instrument, original real time data of the signal can be easily collected on line, which can be analyzed as nonlinear time series. Therefore, reconstructed phase space and fractal method show strong develop potential. The next step is to further consummate these methods and improve their practicality, properly selection of parameters such as sampling frequency, delay time, and how to filter the noise signal in the data collected on the spot.

References

1. Morchen, Fabian. Time series feature extraction for data mining using DWT and DFT(2003)
2. Mandelbrot B.B. The fractal Geometry of Nature. San francisco: Freeman, (1982)
3. Caetano Traina Jr., Agma Trainal Leeyay Wu, Christos Faloutsos. Fast feature selection using fractal dimension (2000)
4. ChiaChou Yeh, Povinelli, R.J., Mirafzal, B., Demerdash, N.A.O.. Diagnosis of stator winding inter-turn shorts in induction motors fed by PWM-inverter drive systems using a time-series data mining technique. Power System Technology, 2004. PowerCon 2004. 2004 International Conference on Volume 1, 21-24 Nov. 2004 (2004) 891 - 896
5. Feng, X., Huang, H., A fuzzy-set-based Reconstructed Phase Space method for identification of temporal patterns in complex time series. Knowledge and Data Engineering, IEEE Transactions on Volume 17, Issue 5, May 2005 . (2005) 601 – 613
6. Keogh Eamonn , Lin Jessica. Finding Unusual Medical Time Series Subsequences: Algorithms and Applications. IEEE Transactions on Information Technology in Biomedicine : Accepted for future publication Volume PP, Issue 99, (2006)1 – 1
7. Gao Kai, Tan Kexiong, Li Fuqi. Pattern recognition of partial discharges based on fractal features of the scatter set[J]. Proceedings of the CSEE. (2002) 22-26.
8. Wang Jing, Shu Hongchun, Chen Xueyun. Fractal exponent wavelet analysis of dynamic power quality[J]. Proceedings of the CSEE. (2004) 40-45.
9. Umeh K C, Mohamed A, Mohamed R, *et al.* Characterizing nonlinear load harmonics using fractal analysis[C]. Proceedings of the 2004 International Symposium on Circuits and Systems. 2004, Vol.5, (2004) 932- 935.
10. Espinoza M, Joye C, Belmans R *et al.* Short-term load forecasting, profile identification, and customer segmentation: a methodology based on periodic time series[J]. IEEE Trans. on Power Systems. (2005) 1622-1630.
11. Sun Yaming, Zhang Zhisheng. A new model of STLF based on the fusion of PSRT and chaotic neural networks[J]. Proceedings of the CSEE. (2004) 44-48.
12. Ajjarapu V, Lee B. Bifurcation theory and its application to nonlinear dynamical phenomena in an electrical power system. IEEE Transactions on Power Systems (1992) 424-431.

13. Chiang H D, Liu C C Chaos in a simple power system. *IEEE Transactions on Power Systems* (1993) 1407—1417.
14. Poisson O, Rioual P, Meunier M. Detection and measurement of power quality disturbances using wavelet transform[J]. *IEEE Trans. on Power Delivery* (2000) 1039-1044.
15. Chen Xunxiang. Wavelet-based measurements and classification of short duration power quality disturbance[J]. *Proceedings of the CSEE*. (2002) 1-6.
16. Mei Xue, Wu Weilin. Power quality classification based on wavelet and artificial neural network[J]. *Journal of Zhejiang University(Engineering Science)*. (2004) 1383-1386.
17. Dash P K, Panigrahi B K, Panda G. Power quality analysis using S-transform[J]. *IEEE Trans. on Power Delivery* (2003) 406-411.
18. Zhan Yong, Cheng Haozhong, Ding Yifeng, *et al.* S-Transform based classification of power quality disturbance signals by support vector machines[J]. *Proceedings of the CSEE*. (2005) 51-56.
19. Xu Yonghai, Xiao Xiangning, Song Y H. Automatic classification and analysis of the characteristic parameters for power quality disturbances[C]. *Power Engineering Society General Meeting* (2004).
20. Shu Hongchun, Wang Jing, Chen Xueyun. Multiscale morphology analysis of dynamic power quality disturbances[J]. *Proceedings of the CSEE* (2004) 63-67.
21. T. Higuchi. Approach to an irregular time series on the basis of the fractal theory. *Physica D*, Vol 31, (1988) 277-283
22. Boshoff, H.F.V. A fast box counting algorithm for determining the fractal dimension of sampled continuous functions. *Communications and Signal Processing, 1992. COMSIG '92.*, *Proceedings of the 1992 South African Symposium on 11 Sept.* (1992) 43 - 48
23. Parker T.S, Chua L.O. *Practical numerical algorithms for chaotic systems*. Springer-Verlag Press (1989)
24. IEEE standards coordinating committee 22 on power quality. *IEEE recommended practice for monitoring electric power quality[S]*. 1995.IEEE Std 1159-1995. (1995)
25. Zhiyong Li and Weilin Wu, *Detection and Identification of Power Disturbance Signals Based on Nonlinear Time Series*, *Proceedings of the 6th World Congress on Control and Automation*, June 21 - 23, 2006, Dalian, China, 7646-7650, (2006)

Mining Compressed Sequential Patterns^{*}

Lei Chang, Dongqing Yang, Shiwei Tang, and Tengjiao Wang

Department of Computer Science & Technology, Peking University, Beijing, China
{changlei, dqyang, tsw, tjwang}@pku.edu.cn

Abstract. Current sequential pattern mining algorithms often produce a large number of patterns. It is difficult for a user to explore in so many patterns and get a global view of the patterns and the underlying data. In this paper, we examine the problem of how to compress a set of sequential patterns using only K *SP-Features* (Sequential Pattern Features). A novel similarity measure is proposed for clustering SP-Features and an effective SP-Feature combination method is designed. We also present an efficient algorithm, called CSP (Compressing Sequential Patterns) to mine compressed sequential patterns based on the hierarchical clustering framework. A thorough experimental study with both real and synthetic datasets shows that CSP can compress sequential patterns effectively.

1 Introduction

Sequential pattern mining, introduced in [2], has become an increasingly important data mining task, due to the significant advances in the biological study as well as other fields. There have been a lot of efficient algorithms proposed to mine sequential patterns [2][5][12]. However, the real bottleneck of frequent pattern mining is not at the efficiency but at the interpretability [7][9]. The large number of patterns discovered hamper the analysis of the patterns, and, it is extremely difficult for users to explore in thousands of patterns, in order to find the potential rules hidden in the data. For example, in the dense biological dataset Snake [11], with a relatively high support 0.95, there will be 25,967 frequent sequential patterns discovered.

A general proposal for the interpretability problem is to find a concise representation of the sequential patterns. Agrawal et al. [2] proposed mining maximal sequential patterns. Maximal pattern mining is a lossy compression method, because the support information of subpatterns can not be recovered from only the complete set of maximal patterns. On the other hand, some lossless methods have also been designed. Closed sequential patterns [8][11] keep all the information (including the expression [7] and the support) of the original set of frequent sequential patterns. However, due to the rigid definition of closed sequential pattern, there are also a large number of closed patterns to be discovered (4,521 for Snake dataset at support 0.95), which is still difficult for a user to handle, and the redundancy in a closed pattern set is significant.

^{*} This work is supported by the National Natural Science Foundation of China under Grant No. 60473051.

A similar problem occurs in the frequent itemset mining environment. Afrati et al.[1] proposed using K frequent(or border) itemsets to approximate a collection of frequent itemsets. The result can be thought as a generalization of maximal frequent itemsets. Xin et al.[7] presented new algorithms RPglobal and RPlocal for compressing frequent itemsets, and Yan et al.[9] studied the profile based model to summarize a set of frequent itemsets. However, to the best of our knowledge, no effective methods, which produce more compact results than using closed sequential patterns without losing support information, have been developed for compressing sequential patterns.

In this paper, a novel structure SP-Feature is designed to represent sequential patterns compactly, and effective methods for efficiently computing SP-Features are developed. More specifically, this paper makes the following contributions. (1) A novel structure SP-Feature is proposed to represent a set of sequential patterns, and it keeps succinct information to facilitate SP-Feature clustering and SP-Feature combination algorithm. (2) A new similarity measure is presented which not only reflects the sequential information of patterns, but also carries the support information of the patterns covered by SP-Features. (3) We present an effective algorithm CSP for compressing sequential patterns, which has linear time complexity. We also provide several optimization techniques to reduce the algorithm's memory usage and improve its efficiency. (4) A new pattern support restoration approach is proposed using probabilistic method. (5) Experimental results show that CSP algorithm can compress sequential patterns effectively.

2 Sequential Pattern Feature

In this section, we first give the formal definitions of some novel concepts introduced in this paper. Then, we describe the pattern combination method. Finally, we discuss how to restore the support of a pattern without consulting the original dataset.

Definition 1 (Pattern Profile). Let α be a frequent sequential pattern in a sequence database \mathcal{D} . The **pattern profile** of α , denoted as $\mathcal{P}(\alpha)$, is computed by first computing the profile[4] of \mathcal{D}_α ($\mathcal{D}_\alpha = \{s | s \supseteq \alpha \wedge s \in \mathcal{D}\}$) with item substitution function

$$S_{xy} = \begin{cases} \omega, & \text{if } x = y \wedge x \in I(\alpha), \\ 1, & \text{if } x = y \wedge x \notin I(\alpha), \\ 0, & \text{if } x \neq y, \end{cases} \quad (1)$$

where $\omega(\omega \geq 1)$ is an item-importance factor and $I(\alpha)$ is the set of distinct items which occurs in α , then keeping only the columns in each of which the maximal frequency is greater than the significance threshold $\theta(0 \leq \theta \leq 1)$. Frequently, we also call the pattern profile of α as the pattern profile of the sequence set \mathcal{D}_α .

The reason we use an item-importance factor $\omega(\omega \geq 1)$ in the computation of a pattern profile is that by giving more importance(i.e., weight) to the items

in the pattern, it makes the *MIF vector*(see Definition 2) of the pattern profile reflect the pattern’s item composition and frequency information more precisely. Since there are always noises in a sequence database[10], we include a significance threshold θ in our definition to remove the columns with insignificant frequencies from pattern profiles.

Definition 2 (MIF Vector). Let \mathcal{P} be the pattern profile of pattern α in a sequence database \mathcal{D} , N be the number of the columns of \mathcal{P} . The **MIF vector**(*master item-frequency vector*) of \mathcal{P} , denoted by $MIF(\mathcal{P})$, is the item-frequency pair vector, $\langle item_1:f_1, item_2:f_2, \dots, item_N:f_N \rangle$, where $item_i$ and f_i , $1 \leq i \leq N$, are the most frequent item in column i of \mathcal{P} , and the frequency that $item_i$ appears in column i respectively. The sequence $\langle item_1, item_2, \dots, item_N \rangle$, composed of the items of $MIF(\mathcal{P})$, is called the **master pattern** of \mathcal{P} or the **master pattern** of the sequence set \mathcal{D}_α , denoted by $MP(\mathcal{P})$ or $MP(\mathcal{D}_\alpha)$.

Definition 3 (Primary Appearance of a Subsequence). Let $\langle item_1 : f_1, item_2 : f_2, \dots, item_N : f_N \rangle$ be the MIF vector of a pattern profile \mathcal{P} , $s_k = \langle item_{k_1}, item_{k_2}, \dots, item_{k_m} \rangle$ be a subsequence of $MP(\mathcal{P})$, and $o_i = \langle item_{k_1} : p_1, item_{k_2} : p_2, \dots, item_{k_m} : p_m \rangle$ be an appearance of s_k in $MP(\mathcal{P})$, where p_j ($1 \leq j \leq m$) is the position of $item_{k_j}$ of that appearance in $MP(\mathcal{P})$. o_i is called the **primary appearance** of s_k in $MP(\mathcal{P})$, if o_i maximizes the sum of frequencies at each position of $\langle p_1, p_2, \dots, p_m \rangle$ in the MIF vector, i.e. $\sum_j f_{p_j}$, $1 \leq j \leq m$.

Note that we can also use $\prod_j f_{p_j}$, $1 \leq j \leq m$, instead of $\sum_j f_{p_j}$, $1 \leq j \leq m$, in the definition of the *primary appearance* of s_k .

Definition 4 (SP-Feature of a Pattern). Let α be a sequential pattern in a sequence database \mathcal{D} . The **SP-Feature**(*sequential pattern feature*) of α , denoted as $SPF(\alpha)$, is a length-3 vector $\langle \chi, \rho, \varphi \rangle$. $\chi = \alpha$ is the representative pattern. ρ is the pattern profile of α , $\rho = \mathcal{P}(\alpha)$. φ is the cardinality of \mathcal{D}_α divided by the cardinality of \mathcal{D} , i.e., $\varphi = \frac{|\mathcal{D}_\alpha|}{|\mathcal{D}|}$, and φ is called the **support** of the SP-Feature.

The definition of SP-Feature can be extended to a set of sequential patterns.

Definition 5 (SP-Feature of a Pattern Set). Let $\mathbb{P} = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ be a set of sequential patterns in \mathcal{D} . $\mathcal{D}_{SPF(\mathbb{P})}$ denotes the set of sequences $\bigcup_i \mathcal{D}_{\alpha_i}$, $1 \leq i \leq m$. The **SP-Feature**(*sequential pattern feature*) of \mathbb{P} , denoted as $SPF(\mathbb{P})$, is a length-3 vector $\langle \chi, \rho, \varphi \rangle$. χ is the representative pattern. ρ is the pattern profile of $\mathcal{D}_{SPF(\mathbb{P})}$. φ , called the **support** of the sequential pattern feature, is the cardinality of $\mathcal{D}_{SPF(\mathbb{P})}$ divided by the cardinality of \mathcal{D} , i.e., $\varphi = \frac{|\mathcal{D}_{SPF(\mathbb{P})}|}{|\mathcal{D}|}$. We also say that the sequences in $\mathcal{D}_{SPF(\mathbb{P})}$ **support** the SP-Feature. χ , ρ and φ are also written as $\chi(SPF(\mathbb{P}))$, $\rho(SPF(\mathbb{P}))$ and $\varphi(SPF(\mathbb{P}))$, respectively.

A SP-Feature represents a set of sequential patterns succinctly. The SP-Feature of a pattern set can be computed by calling ComSPF (**C**ombine two **SP-Features**) algorithm iteratively. ComSPF combines two SP-Features by merging the corresponding components of the SP-Features. The intuition behind the algorithm

is apparent, but the method is somewhat intricate. Due to limited space, please see [3] for the detailed algorithm description.

Algorithm 1. (*ComSPF*) Combine two SP-Features

Input: (1) Two SP-Features \mathcal{SPF}_1 and \mathcal{SPF}_2 ; (2) A sequence database D .

Output: (1) A boolean value indicating the success/failure of the combination.

(2) If successful, the result SP-Feature \mathcal{SPF}_{out} .

Method:

1. Select \mathcal{SPF}_i such that $i = \operatorname{argmax}_i \varphi(\mathcal{SPF}_i), i \in \{1, 2\}$;
2. $\mathcal{SPF}_j =$ the other input SP-Feature;
3. Set $\rho(\mathcal{SPF}_{out}) = \rho(\mathcal{SPF}_i)$;
4. for each sequence $s \in \mathcal{D}_{\mathcal{SPF}_j} - \mathcal{D}_{\mathcal{SPF}_i}$
5. Use s to update $\rho(\mathcal{SPF}_{out})$;
6. if $\chi(\mathcal{SPF}_i) \not\subseteq \mathcal{MP}(\rho(\mathcal{SPF}_{out}))$
7. return false;
8. else $o_i =$ the primary appearance of $\chi(\mathcal{SPF}_i)$ in $\mathcal{MP}(\rho(\mathcal{SPF}_{out}))$;
9. if $\chi(\mathcal{SPF}_j) \not\subseteq \mathcal{MP}(\rho(\mathcal{SPF}_{out}))$
10. return false;
11. else $o_j =$ the primary appearance of $\chi(\mathcal{SPF}_j)$ in $\mathcal{MP}(\rho(\mathcal{SPF}_{out}))$;
12. Compute $\chi(\mathcal{SPF}_{out})$ by combining o_i and o_j ;
13. Set $\varphi(\mathcal{SPF}_{out}) = \frac{|\mathcal{D}_{\mathcal{SPF}_{out}}|}{|D|}$;
14. return true;

Here, we discuss how to approximately estimate the support of a pattern using only the information contained in the corresponding MIF vector without consulting the original dataset.

Lemma 1 (Support Estimation). Let F be the SP-Feature of a set of sequential patterns $\mathbb{P} = \{\alpha_1, \alpha_2, \dots, \alpha_m\}$ in a sequence database \mathcal{D} , $\langle i_1:f_1, i_2:f_2, \dots, i_n:f_n \rangle$ be the MIF vector of $\rho(F)$, and N_k be the number of appearances of $\alpha_k (1 \leq k \leq m)$ in $\mathcal{MP}(\rho(F))$. $A_j = \langle \alpha_k^1:p_1^j, \alpha_k^2:p_2^j, \dots, \alpha_k^l:p_l^j \rangle$ is the j th $(1 \leq j \leq N_k)$ appearance of α_k in $\mathcal{MP}(\rho(F))$, where l is the length of α_k , $\alpha_k^x (1 \leq x \leq l)$ denotes the x th item of α_k , and p_x^j is the position where α_k^x of the j th appearance appears in $\mathcal{MP}(\rho(F))$. The corresponding frequency vector of A_j in $\mathcal{MIF}(\rho(F))$ is $\langle f_1^j, f_2^j, \dots, f_l^j \rangle$. Assume that, in $\mathcal{MIF}(\rho(F))$, $f_y (1 \leq y \leq n)$ is approximately equal to the probability that item i_y appears at the profile column y . The support of α_k can be estimated by using

$$\hat{\sigma}(\alpha_k) = \varphi(F) \times (1 - \sum_{j=1}^{N_k} (1 - f_1^j \times f_2^j \times \dots \times f_l^j)). \tag{2}$$

Proof. Proved in [3]. □

3 Compressing Method

In this section, we give a formal definition of the compression problem and an effective algorithm is designed to address the compression problem.

Definition 6 (Sequential Pattern Compression Problem). *Given a set of frequent sequential patterns $\mathbb{P} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$, the **sequential pattern compression problem** is to find K SP-Features to represent the patterns in \mathbb{P} .*

To address the *sequential pattern compression problem*, a general proposal is to first group the set of patterns according to a certain similarity measure, then calculate a SP-Feature for each group. Since we can construct a SP-Feature for each pattern which contains only that pattern, the similarity between patterns can be measured by the similarity between SP-Features.

Algorithm 2. (CSP) *Compressing Sequential Patterns*

- Input:** (1) A set of frequent sequential patterns $\mathbb{P} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$.
 (2) A sequence database D . (3) The number of SP-Features K .
 (4) A distance threshold τ .

Output: A set of SP-Features \mathbb{F} .

Method:

1. for each pattern $\alpha \in \mathbb{P}$
2. Compute $SPF(\alpha)$, and add it to \mathbb{F} ;
3. Compute a triangular distance matrix for the $SPFs$ in \mathbb{F} ;
4. while $|\mathbb{F}| > K$ and $MinimalDistance < \tau$
5. Find SPF_i and SPF_j such that $i, j = argmin_{i,j} d(SPF_i, SPF_j)$;
6. $\langle bRet, SPF_{out} \rangle = ComSPF(SPF_i, SPF_j)$;
7. if $lbRet$
8. Set $d(SPF_i, SPF_j) = MaxDistanceValue$;
9. continue;
10. Remove SPF_i and SPF_j from \mathbb{F} ;
11. Add SPF_{out} to \mathbb{F} ;
12. for each $SPF_k \in \mathbb{F}$ ($SPF_k \neq SPF_{out}$)
13. Set $d(SPF_{out}, SPF_k) = f(d(SPF_i, SPF_k), d(SPF_j, SPF_k))$;

In this paper, we propose using the similarity measure defined below to measure the similarity between two SP-Features.

$$sim(f_1, f_2) = corr(f_1, f_2) - dist(\chi(f_1), \chi(f_2)). \tag{3}$$

The similarity between two SP-Features f_1 and f_2 consists of two components. $corr(f_1, f_2)$ measures the correlation (or similarity) between \mathcal{D}_{f_1} and \mathcal{D}_{f_2} , and $dist(\chi(f_1), \chi(f_2))$ measures the distance between the two representative patterns of f_1 and f_2 .

$$corr(f_1, f_2) = \frac{|\mathcal{D}_{f_1} \cap \mathcal{D}_{f_2}|}{|\mathcal{D}_{f_1} \cup \mathcal{D}_{f_2}|}. \tag{4}$$

$$dist(\chi(f_1), \chi(f_2)) = 1 - \frac{length(lcs(\chi(f_1), \chi(f_2)))}{min_{length}(\chi(f_1), \chi(f_2))}, \tag{5}$$

where $lcs(\chi(f_1), \chi(f_2))$ is the *longest common subsequence* of the two representative patterns.

Using the similarity measure defined above, a sequential pattern compression algorithm CSP (Compressing Sequential Patterns) is developed based on hierarchical clustering framework. The result of hierarchical clustering is a dendrogram which allows users to explore the sequential patterns at each level of granularity. The f function in Line 13 can be AVG, MAX, or MIN. Due to the limited space, please see [3] for the detailed algorithm description and the complexity analysis.

The efficiency of CSP algorithm can be improved by two optimization techniques. First, we can consider using closed sequential patterns, instead of frequent sequential patterns, as input. It is easy to show that whether we use closed sequential patterns as input or frequent sequential patterns as input, CSP algorithm will complete with similar results (proved in [3]).

Second, we propose keeping only top- H ($H \geq 1$) item-frequency pairs for each profile column. It is because the critical information of a pattern profile is in its MIF vector, and a pattern profile often keeps a lot of tiny frequencies, which can be ignored without impairing the compression quality significantly. It can be easily shown that keeping only top- H item-frequency pairs also accelerates the calculation of pairwise alignment.

4 Experimental Results

In this section, we perform a thorough evaluation of the CSP algorithm on real and synthetic datasets. The algorithm was coded in C++ and compiled by g++ in cygwin environment. All experiments were done on a 2.8GHz Intel Pentium-4 PC with 512MB memory, running Windows Server 2003.

We use restoration error, proposed in [9], to measure the compression quality and choose the set of input patterns as the testing case. Due to space limitation, we refer readers to [3] for more experimental results (including the sensitivity to parameters and the effects of optimization techniques).

Snake[11]: This dataset is about a family of eukaryotic and viral DNA binding proteins. It contains 192 Toxin-Snake protein sequences and 20 unique items. There are 4,521 closed sequential patterns (25,967 frequent sequential patterns) generated with $\text{min_sup}=0.95$. Figure 1 shows the restoration error over CSP and CSP-fast algorithms with the significance threshold at 0 and 0.15 respectively. In CSP-fast, we do not use a clustering method to construct the binary guide tree used by multiple sequence alignment, instead, we simply generate a random guide tree. Overall, the 4,152 closed patterns can be compressed into 20 or less SP-Features with good quality and the restoration error at 20 profiles is less than 4%.

Gazelle: This dataset is a web click-stream dataset. It contains totally 33,305 sessions, and each session consists of a series of page views. This dataset is a sparse dataset with 1,037 distinct items, and its average length is only 2.50. There are 3,787 closed sequential patterns generated with $\text{min_sup} = 0.0003$. Figure 2 shows the restoration error. The restoration error at $K=300$ for CSP ($\theta = 0.05$) is about 40%. Overall, the compression quality for Gazelle is much worse than that

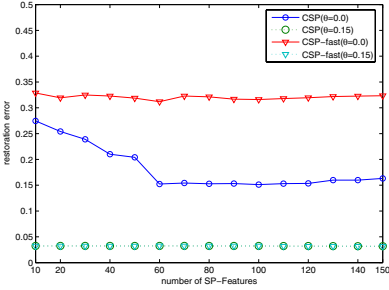


Fig. 1. Snake dataset

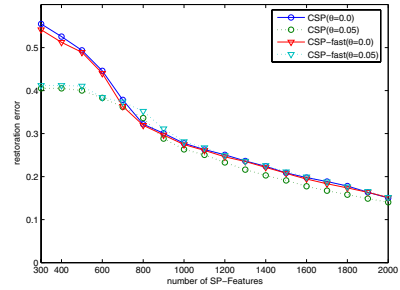


Fig. 2. Gazelle dataset

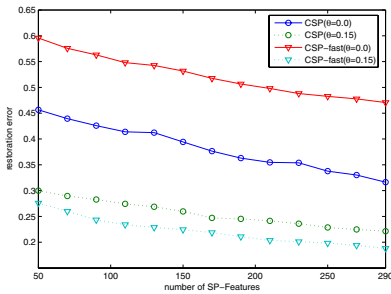


Fig. 3. NCBI dataset

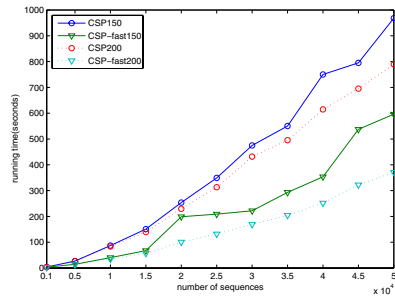


Fig. 4. Running time w.r.t database size

of Snake. The difference is due to the underlying distribution of patterns. There is much redundancy in patterns of Snake, while Gazelle contains random and short click sequences of web users, and the patterns of Gazelle is short(average length=2.06) and sparse. Thus, when two patterns have quite different item composition and item order, it is really not good to combine them into one.

NCBI: This dataset contains 1,074 protein sequences that were extracted from the web site of NCBI(National Center for Biotechnology Information) by the conjunction of (1)sequence length range = [30:40] and(2)publication period=[2003 /1/1, 2005/12/31]. There are 1,117 closed sequential patterns generated with $\text{min_sup}=0.41$. Interestingly, at $\text{min_sup}=0.41$, there are also the same number of frequent sequential patterns generated. Mining closed sequential patterns can not give any compression to this dataset. NCBI contains sequences of tens of families, and its average length(35.53) is also less than that of Snake, so it is sparser than Snake, but denser than Gazelle. It is expected that the compression quality will lie between that of Snake and that of Gazelle. Figure 3 confirms our conjecture.

Running time:We produce synthetic datasets using Rose data generator[6]. We tested the running time of our pattern compression methods by varying the number of sequences from 1,000 to 50,000. We obtained top-1000 closed patterns

for each dataset, and compressed them into 150 and 200 SP-Features respectively. Figure 4 shows that the running time of CSP and CSP-fast increases linearly with the number of sequences, and CSP-fast outperforms CSP by about a factor of two.

5 Conclusions and Future Work

In this paper, we examined the issues for compressing sequential patterns. The SP-Feature is employed to represent a collection of sequential patterns succinctly. A novel similarity measure is proposed for clustering SP-Features and effective SP-Feature combination method is designed. We also developed a new SP-Feature clustering algorithm CSP based on hierarchical clustering framework. A thorough experimental study with both real and synthetic datasets has been conducted to show that CSP can compress sequential patterns effectively and efficiently. In the future, we will examine how to compress graph and other structured patterns.

References

1. Afrati F., Gionis A., Mannila H.: Approximating a Collection of Frequent Sets. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004)12-19
2. Agrawal R., Srikant R.: Mining Sequential Patterns. Proceedings of the Eleventh International Conference on Data Engineering. (1995)3-14
3. Chang L, Yang D, Tang S, Wang T.: Mining Compressed Sequential Patterns. Technical Report PKUCS-R-2006-3-105, Department of Computer Science & Technology, Peking University. (2006).
4. Gribskov M, McLachlan A, Eisenberg D.: Profile analysis: Detection of distantly related proteins. Proceeding of National Academy Science. (1987)4355-4358
5. Pei J., Han J., Mortazavi-Asl B., Pinto H., Chen Q., Dayal U., Hsu M.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. Proceedings of International Conference on Data Engineering. (2001)215-224
6. Stoye J., Evers D., Meyer F.: Rose: generating sequence families. Bioinformatics. 14(2),(1998)157-163
7. Xin D., Han J., Yan X., Cheng H.: Mining Compressed Frequent-Pattern Sets. Proceedings of International Conference on Very Large Data Bases. (2005)709-720
8. Yan X., Han J., Afshar R.: CloSpan: Mining Closed Sequential Patterns in Large Datasets. Proceedings of SIAM International Conference on Data Mining. (2003)
9. Yan X., Cheng H., Han J., Xin D.: Summarizing Itemset Patterns: A Profile-Based Approach. Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2005)314-323
10. Yang J., Wang W., Yu S.P., Han J.: Mining Long Sequential Patterns in a Noisy Environment. Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. (2002)406-417
11. Wang J., Han J.: BIDE: Efficient Mining of Frequent Closed Sequences. Proceedings of International Conference on Data Engineering. (2004)79-90
12. Zaki M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning. 42(1/2),(2001)31-60

Effective Feature Preprocessing for Time Series Forecasting

Jun Hua Zhao¹, ZhaoYang Dong¹, and Zhao Xu²

¹ The School of Information Technology and Electrical Engineering, The University of Queensland, St Lucia, QLD 4072, Australia
{zhao, zdong}@itee.uq.edu.au

² Centre for Electric Technology (CET), Ørsted*DTU, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
zx@oersted.dtu.dk

Abstract. Time series forecasting is an important area in data mining research. Feature preprocessing techniques have significant influence on forecasting accuracy, therefore are essential in a forecasting model. Although several feature preprocessing techniques have been applied in time series forecasting, there is so far no systematic research to study and compare their performance. How to select effective techniques of feature preprocessing in a forecasting model remains a problem. In this paper, the authors conduct a comprehensive study of existing feature preprocessing techniques to evaluate their empirical performance in time series forecasting. It is demonstrated in our experiment that, effective feature preprocessing can significantly enhance forecasting accuracy. This research can be a useful guidance for researchers on effectively selecting feature preprocessing techniques and integrating them with time series forecasting models.

1 Introduction

Time series forecasting is a major challenge in many real-world applications, such as stock price analysis [1], electricity price forecasting [2] and flood forecasting [3]. Time series forecasting is to predict the values of a continuous variable (called *response variable*) with a forecasting model based on historical data. Given its valuable potential in many applications, time series forecasting has been a hot research topic since the past decade. There are many time series models developed for this purpose, including ARIMA [2] and Fuzzy-neural autoregressive models [4]. Neural Networks can also be used to approximate time series data [5]. Garch model [6] is proven to have good performance especially in handling the volatility in some time series such as stock prices. Recently, Support Vector Machine (SVM) has also been applied in time series forecasting and achieved satisfactory results [7].

Given a specific forecasting model, feature preprocessing is essential for improving forecasting accuracy. *Feature preprocessing* [8] is the process of selecting the relevant factors of a response variable, and/or constructing new features based on these factors. The following two reasons make feature preprocessing very important for time series forecasting. Firstly, training the forecasting model with irrelevant features can greatly degrade the forecasting accuracy, and also decrease the training

speed to an intractable level; moreover, when there are too many available features, manually determine the relevant features of the response variable with statistical plots will become impossible. Therefore, effective methods are required to automatically choose the most relevant feature set. This process is known as *feature selection* [9]. Secondly, the correlations between different features can seriously affect the performance of the forecasting model. *Feature extraction* [10] techniques can be employed to construct the new features that are mutually independent, and reduce the noise in data.

Feature selection/extraction have been extensively studied in statistics [11], machine learning [12] and data mining [13], and widely applied to many areas. In [14-17], several feature selection techniques are proposed as a preprocessing tool selecting the relevant feature subset. Feature extraction techniques can be applied to generate independent features which still maintain the relationships between original features and response variable. These techniques include principal component analysis (PCA) [18] and independent component analysis (ICA) [19]. Signal processing techniques such as wavelet decomposition and reconstruction [20], have also been applied to reduce the noise in training data. Some of these techniques have been applied in time series forecasting problem [20-22]. However, no systematic research has been performed yet to study the general performance of different feature preprocessing techniques. Without empirical studies, it is difficult to determine the most effective and proper feature preprocessing techniques for time series forecasting.

In this paper, the authors attempt to conduct a comprehensive study of existing feature preprocessing techniques, and evaluate their empirical performance in time series forecasting. We will firstly give a brief introduction to the general procedure of feature preprocessing and several existing feature selection/extraction techniques. A comprehensive empirical study will be given to compare the performance of different techniques on real-world datasets. The main contribution of this paper is to present a comprehensive study on how to choose the most suitable feature preprocessing techniques based on the characteristics of data and the requirements of time series analysis.

The rest of the paper is organized as follows. The definitions of time series forecasting and feature selection/extraction are presented in Section 2. A systematic introduction to feature preprocessing, including the general procedure and several widely used techniques are given subsequently. In Section 4, Support Vector Machine is briefly reviewed as the forecasting model for our case studies. In Section 5, a comprehensive empirical study of feature preprocessing is given. Finally, Section 6 concludes the paper.

2 Problem Formulation

Before we discuss time series forecasting and its feature preprocessing, clear definitions of forecasting and feature selection/extraction should be firstly given. Time series forecasting techniques are studied in several different areas, including statistics, machine learning, and data mining. Generally, forecasting is considered as a *supervised learning* problem [23], and can be solved by regression [23] or time series techniques [23]. The formal definition of forecasting is given as follows:

Definition 1: Given a historical time series dataset $S = \{(X_1, y_1), (X_2, y_2) \dots (X_t, y_t)\}$, where y_t is the *response variable* to be forecast at time t , and $X_t = (x_{t1}, x_{t2}, \dots, x_{tm})$ represents the values of relevant factors. We assume that a function dependency $f : S \rightarrow y_{t+n}$ exists, where y_{t+n} denotes the response variable n time units later than time t . Then the *forecasting* problem is to find a proper f' approximating f , and use f' to forecast the values of the response variable in the future.

Because training the forecasting model with the original feature set may not reach the optimal forecast accuracy, feature selection and extraction techniques are proposed to obtain a better feature set. Their definitions are given as follows:

Definition 2: Given a historical time series dataset $S = \{(X_1, y_1), (X_2, y_2) \dots (X_t, y_t)\}$, and denote $X = \{x_1, x_2, \dots, x_m\}$ as the original feature set. *Feature selection* is the process of generating an optimal feature subset $X' \subseteq X = \{x_1, x_2, \dots, x_m\}$, based on which f' will have the optimal forecast accuracy on the future dataset.

A simple illustration of feature selection is given in Fig.1. Clearly, among the many features available, only a small number of relevant features are identified through feature selection.

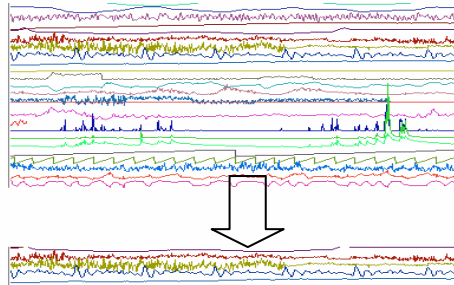


Fig. 1. An example of feature selection

Definition 3: Given the historical time series dataset $S = \{(X_1, y_1), (X_2, y_2) \dots (X_t, y_t)\}$, and denote $X = \{x_1, x_2, \dots, x_m\}$ as the original feature set. *Feature extraction* is a process to generate the optimal feature set $X' = (x'_1(x_1, \dots, x_m), x'_2(x_1, \dots, x_m) \dots x'_l(x_1, \dots, x_m))$, based on which f' will have the optimal forecast accuracy on the future dataset. Here each new feature in X' is a function of the original features.

Obviously, feature selection, which selects a subset of the original feature set, is a special case of feature extraction. Feature extraction transforms the original feature set into another feature space. However, the optimal forecast accuracy is the objective of both feature selection and extraction.

3 Feature Preprocessing Techniques

In this section, the authors will first introduce the basic procedure of feature preprocessing before discussing some well-known feature preprocessing techniques. We will also discuss how to choose the appropriate feature preprocessing techniques to suit the specific requirements of time series data analysis.

3.1 Procedure of Feature Preprocessing

Feature preprocessing usually includes four steps as follows [9]:

- **Candidate set generation.** Candidate set generation is a search procedure that uses a certain search strategy to produce candidate feature sets. Here the candidate feature sets can be the subsets of the original feature set (feature selection), and also can be the transformations of the original feature set (feature extraction). Feature preprocessing techniques can be classified according to the different search strategies in this step.
- **Candidate set evaluation.** An evaluation criterion should be applied to determine whether each candidate feature set is better than the previous best one. If the new feature set is superior, it replaces the previous best feature set. According to the different evaluation models employed in the step of candidate set evaluation, feature preprocessing techniques can also be categorized as the *filter model*, *wrapper model* and *hybrid model* [9].
- **Stopping criterion.** The process of candidate set generation and evaluation is repeated until a given stopping criterion is satisfied.
- **Results validation.** The selected optimal feature set usually needs to be validated by prior knowledge or different tests with synthetic and/or real-world data sets.

The complete feature preprocessing procedure and the relationship between the four major steps are given in Fig. 2.

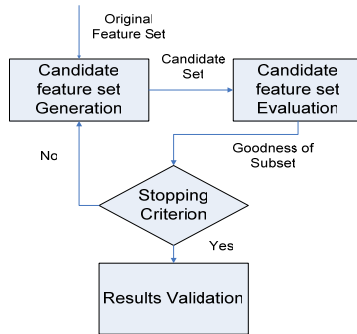


Fig. 2. Feature preprocessing procedure

3.2 Some Well-Known Feature Preprocessing Techniques

In this subsection, several existing feature preprocessing techniques will be discussed in details. These techniques employ different search strategies in candidate set

generation, and different evaluation criteria in candidate set evaluation. We will discuss several combinations of search strategy and evaluation criterion respectively. Their effectiveness in time series forecasting will be demonstrated in experiments section.

The first step of feature preprocessing is to generate candidate feature sets. To guarantee identifying the optimal candidate feature set, a straightforward method is to perform an exhaustive/complete search on all possible feature subsets. However, this is computationally expensive especially when dealing with a large number of original features. Consequently, several heuristic search strategies are proposed as follows:

- **Complete Search** [9]. As mentioned above, complete search is usually intractable, although it guarantees the global optimum. This search strategy is seldom used in real-world applications.
- **Best-first Search.** Best-first strategy searches the space of feature subsets by *greedy hill climbing* [24], and employs a backtracking facility to avoid the local optima. It may start with the empty set and search forward, or with the full set of attributes and search backward, or start at any point and search in both directions. Because of its heuristic strategy, there is no guarantee of the global optimum.
- **Greedy-stepwise Search.** This search strategy performs greedy forward or backward search, starting from an arbitrary point in the feature space [25]. It can produce a ranked list of attributes by traversing the feature space from one side to the other and recording the order that attributes are selected. This strategy is also locally optimal.
- **Genetic Search.** Genetic search employs the simple genetic algorithm to locate the global optimum [26]. Theoretically this strategy has the capability of locating the global optimum. Meanwhile its search speed is not significantly slower than best-first and greedy-stepwise methods.
- **Random Search.** Random search generates each candidate set without any deterministic rule [9]. The users can predetermine the percentage of the feature space to be explored. Theoretically this strategy is globally optimal if a large percentage is set, which usually makes the search process intractable.
- **Ranker.** When a predetermined evaluation criterion can be employed to evaluate each attribute individually, we may simply rank the features according to their values given by the criterion. This strategy is the fastest search strategy. However, when the features are correlated, results of ranker strategy are usually unreliable.
- **Feature transformation.** Different from the above search strategies, feature transformation generates new feature sets according to the characteristics of data. For example, *principle component analysis (PCA)* [18] chooses enough eigenvectors to account for some percentages of the variance in original data (e.g. 95%). Attribute noise can be filtered by transforming original features to the principal component (PC) space. Other feature transformation algorithms include the *Fourier transformation* [27], *wavelet transformation* [20], *independent component analysis (ICA)* [19], etc.

After the candidate feature sets are generated, a number of evaluation criteria can be used to evaluate the quality of feature sets. These criteria include:

- **Distance measure** [9]. Distance measures are also known as *separability*, *divergence*, or *discrimination* measures. For different values of a response

variable, a feature X is considered more relevant than another feature Y if X produces a greater difference between the conditional probabilities than Y. A famous feature selection technique employing distance measure is *relief* [16].

- **Information measure** [15]. Information measures calculate the *information gain* [15] of each feature. The information gain from a feature X is defined as the difference between the prior uncertainty and expected posterior uncertainty considering X. Features with large information gain are considered as good features. There are other information measures, such as *gain ratio* [28].
- **Dependency measure**. Also known as *correlation* measures or *similarity* measures, dependency measure evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features are preferred if they are highly correlated with the response variable and have low inter-correlation. *CFS* [14] is a popular dependency measure.
- **Consistency measure** [9]. Given different values of a response variable, consistency measures aims at searching a minimum number of features, which have a predictive ability similar to the full set of features. Two instances are inconsistent if they have the same feature values but different values of the response variable.

The above four measures belong to the *filter* model [9]. They evaluate the quality of feature sets according to predetermined measures. These measures are independent of a certain forecasting model. However, filter models may not lead to the highest forecast accuracy, because they do not guarantee to be consistent with accuracy criteria. To tackle this shortcoming, the following two models can be used:

- **Wrapper model** [17]. Different from the filter model, the wrapper model integrates a specific forecasting model (e.g. SVM) with the feature preprocessing process. After each candidate feature set is generated, wrapper model trains the forecasting model with this feature set and obtains the forecasting accuracy. This accuracy will then be used as the quality criterion of feature sets in the second step of feature preprocessing. The wrapper model usually outperforms the filter model, because it guarantees to generate the feature set leading to highest forecasting accuracy. However, the wrapper model requires a training time much longer than the filter model.
- **Hybrid model** [9]. Hybrid model employs different criteria of both filter and wrapper models at different search stages to overcome the limitations of filter model and wrapper model.

Given the feature preprocessing techniques discussed above, users may select different methods according to different purposes and requirements. First, several evaluation measures, such as the distance, information and consistency measures, are designed to handle discrete response variables. Therefore, discretization [23] should be performed on data before these measures can be applied in time series forecasting. Second, feature extraction methods should be applied considering the quality of data. For example, PCA can be applied when original features are highly correlated, while noisy data can be filtered with wavelet transformation.

4 Support Vector Machine

Support Vector Machine (SVM) will be used in the experiments as the forecasting model. We give a brief introduction to SVM for completeness.

SVM is a new machine learning method developed by Vladimir Vapnik et al at Bell Laboratories [29]. This method received increasing attention in recent years because of its excellent performance in both classification and regression. It has been proven that SVM has excellent performance in time series forecasting problems [7].

The simplest form of SVM is the *linear regression*, which can be used for linear training data. For the training data $\{(X_1, y_1), \dots, (X_l, y_l)\} \subset R^n \times R$, we can use Vapnik's ϵ -insensitive loss function

$$|y - f(X)|_\epsilon := \max\{0, |y - f(X)| - \epsilon\} \tag{1}$$

to estimate a linear regression function

$$f(X) = \langle W \bullet X \rangle + b. \tag{2}$$

with precision ϵ . This linear regression problem can be solved by minimizing

$$\frac{\|W\|^2}{2} + C \sum_{i=1}^m |y_i - f(X_i)| \tag{3}$$

It is equivalent to a constrained optimization problem of (4)-(5):

$$\text{Minimize} \quad \frac{\|W\|^2}{2} + C \sum_{i=1}^m |\xi_i + \xi_i^*| \tag{4}$$

$$\begin{aligned} \text{Subject to} \quad & \langle W \bullet X_i \rangle + b - y_i \leq \epsilon + \xi_i \\ & y_i - \langle W \bullet X_i \rangle + b \leq \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \tag{5}$$

The *Lagrange multipliers method* can be used to solve this optimization problem.

In most of real-world problems, the relationship between y and X is not linear. One method to deal with non-linear data is to use a map function $\Phi(X): R^n \mapsto H$ to map the training data from input space into some high dimensional feature space where the data become linear. SVM can then be applied in the feature space. Note that the training data used in SVM are only in the form of dot product, therefore, after the mapping the SVM algorithm will only depend on the dot product of $\Phi(X)$. If we can find a function that can be written in the form of $K(X_1, X_2) = \langle \Phi(X_1), \Phi(X_2) \rangle$, the mapping function $\Phi(X)$ will not need to be explicitly calculated in the algorithm. $K(X_1, X_2)$ is a *kernel function* or *kernel*. Radial basis kernel [29] is used in this paper:

$$K(X, Y) = \exp\left(-\frac{\|X - Y\|^2}{2\sigma^2}\right) \tag{6}$$

5 Experiment

We conduct a comprehensive experiment to study the effectiveness of several feature preprocessing techniques in the time series forecasting problem. The electricity price

dataset of the *National Electricity Market (NEM)* of Australia is used for the experiment. The market operator NEMMCO operates this competitive electricity market and publishes the historical and real-time data of NEM *regional reference price (RRP)* at its website. Electricity price forecasting is known as a challenging time series forecasting problem because of the high volatility of electricity prices [6]. The data of electricity prices are therefore used as the experiment data in this paper.

In the experiment, 15 feature preprocessing techniques will be firstly applied on the NEM price data of Jan, 2004, and generate 15 feature sets. These 15 feature sets will be employed to train 15 SVM regression models separately. Finally these 15 SVM models will be tested with the NEM price data of Jan, 2005. The forecast accuracy achieved by these 15 models represents the quality of the feature sets produced by corresponding feature preprocessing techniques.

Mean absolute percentage error (MAPE) is a commonly used criterion of forecast accuracy, due to its robustness and simplicity. MAPE is defined as:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \times 100 \right| \quad (7)$$

In our experiment, MAPE will also be used as a measure of forecast accuracy and feature set quality.

The real price data of NEM, which originally include 48 features, are used in the experiment. We firstly train a SVM model with all 48 features. The data of Jan, 04 is chosen as the training data, while the data of Jan, 05 is the test data. The forecasting result is shown in Fig. 4.

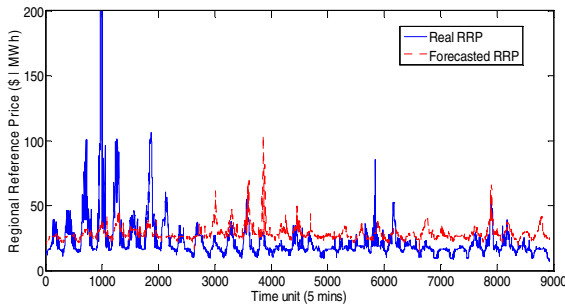


Fig. 4. Price forecast with original feature set

As shown in Fig.4, the forecast with the original feature set has large errors. The MAPE is 42.84%, which is not satisfactory.

In the second step of the experiment, 15 different feature preprocessing techniques are applied to generate 15 different feature sets. 15 SVM models are then trained separately with these 15 feature sets. Real market data of Jan, 04 and Jan, 05 are still used as the training and testing data. The accuracy achieved is listed as below:

Table 1. Forecast accuracy of 15 feature preprocessing techniques

Index	Search/Generation Strategy	Evaluation Criterion	Number of features generated	MAPE (%)
1	Best-First	CFS	2	24.51
2	Genetic Search	CFS	14	15.34
3	Random Search	CFS	11	25.24
4	Greedy Stepwise	CFS	2	24.51
5	Ranker	Chi-square measure [25]	12	14.04
6	Best-First	Consistency measure	2	24.51
7	Greedy Stepwise	Consistency measure	10	29.04
8	Genetic Search	Consistency measure	10	23.05
9	Ranker	Gain ratio	15	22.19
10	Ranker	Information Gain	13	20.27
11	Ranker	OneR Attribute Evaluation [25]	10	26.62
12	N.A.	PCA	13	13.17
13	Ranker	Symmetrical Uncertainty [25]	18	22.47
14	Ranker	Relief	15	12.78
15	Genetic Search	Wrapper Model	28	9.46

As shown in Table 1, every feature preprocessing technique leads to a smaller MAPE than that of the original feature set. This means that including irrelevant features in the forecasting model can significantly decrease the forecasting accuracy.

Among the 15 techniques, wrapper model + genetic search is the best approach with a MAPE < 10%. Moreover, relief + ranker, PCA, chi-square measure + ranker and CFS + genetic search also have good performance, which are within 15%.

The average accuracy of different search/generation strategies is listed in Table 2. Obviously, genetic search has a better performance than other alternatives. A surprising phenomenon is that, random search performs poorly, although it is theoretically global optimal. An explanation is that, to speed up the random search, we usually set a small percentage of the feature space to be explored. For example, in our experiment, we only search $1/10^6$ of the feature space. However, the random search costed more than 24 hours on a PC with Pentium IV CPU and 512M memory. Since only a small region of the feature space has been searched, random search usually cannot locate the global optimum.

Table 2. Average accuracy of different search/generation strategies

Search/Generation Strategy	Average number of features generated	Average MAPE (%)
Best-First	2	24.51
Genetic Search	17	15.95
Random Search	11	25.24
Greedy Stepwise	6	26.78
Ranker	14	19.73

The results of Table 2 are also visualized in Fig. 5. As is shown, methods that generate fewer features usually produce larger errors. This can be explained as, besides the irrelevant and noisy features, some relevant features are also removed by those feature preprocessing methods, which accounts for the performance degradation. Therefore, the optimal feature preprocessing technique should remove all irrelevant features while still keeping the features with useful information.

The average accuracy of different evaluation criteria is also listed in Table 3. The Wrapper model is the best evaluation criterion, although it is also the most time-consuming. Within the filter models, Chi-square measure and Relief have better performance. The MAPE of PCA is also at a low level, which means that severe feature correlations exist in the price data.

The results of Table 3 are illustrated in Fig. 6. Similarly, wrapper model, which has the largest feature number, has the smallest MAPE.

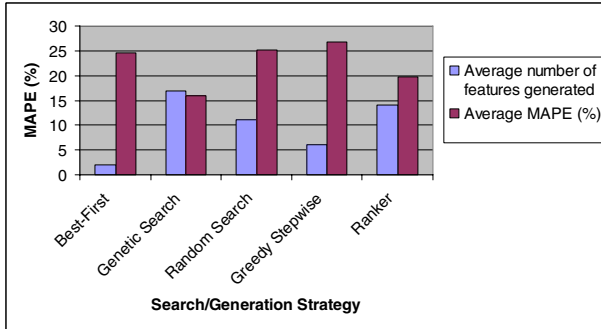


Fig. 5. MAPE and feature numbers of different search/generation strategy

To understand the importance of feature preprocessing, the price forecasts using wrapper model + genetic search, relief + ranker and PCA, are plotted in Figs. 7-9.

Table 3. Average accuracy of different evaluation criteria

Evaluation criterion	Average number of features generated	Average MAPE (%)
CFS	7	22.4
Chi-square measure	12	14.04
Consistency measure	7	25.53
Gain Ratio	15	22.19
Information Gain	13	20.27
OneR Evaluation	10	26.62
PCA	13	13.17
Symmetrical Uncertainty	18	22.47
Relief	15	12.78
Wrapper	28	9.46

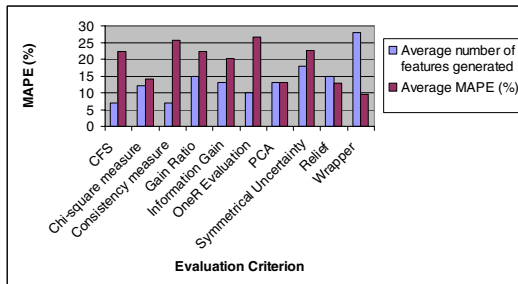


Fig. 6. MAPE and feature numbers of different evaluation criteria

Comparing Fig. 4 with Figs. 7-9, we can clearly observe that the performance of SVM is significantly improved after effective feature preprocessing techniques are integrated. This clearly demonstrates that feature preprocessing is an important part of time series forecasting. Moreover, selecting a proper feature preprocessing method is also essential to further enhance the forecasting accuracy, which is the major contribution of this paper.

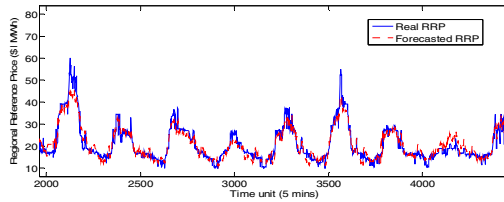


Fig. 7. Price forecast given by wrapper model + genetic search

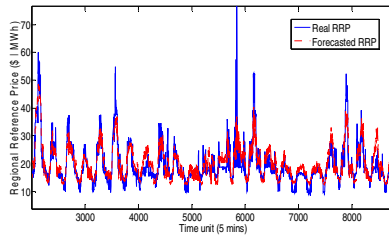


Fig. 8. Price forecast given by relief + ranker

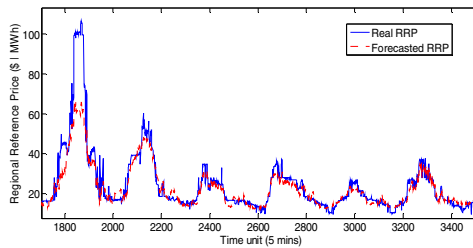


Fig. 9. Price forecast given by PCA

6 Conclusion

A systematic study of feature preprocessing techniques is presented in this paper. The authors discuss the importance of feature preprocessing in time series forecasting, as well as selecting proper feature preprocessing techniques in specific forecasting problems. Comprehensive experiments are conducted to demonstrate the effectiveness of feature preprocessing techniques with real-world electricity price data. According to the case studies, proper feature preprocessing techniques can significantly enhance

the performance of time series forecasting models. It is shown in the experiment that, considering the forecasting accuracy, genetic search is the best search strategy, while wrapper model is the best evaluation criterion. Combining these techniques with a forecasting algorithm such as Support Vector Machine, a more accurate time series forecast can be provided. The main contribution of this paper exists in providing a comprehensive guidance on properly selecting effective feature preprocessing techniques and integrating them with forecasting models.

References

1. T. Kolarik, G. Rudorfer: Time Series Forecasting Using Neural Networks. Proceedings of the international conference on APL (SIGAPL'94), pages 86-94, Antwerp, Belgium, 1994.
2. J. Contreras, R. Espinola, F.J. Nogales and A.J. Conejo: ARIMA Models to Predict Next-Day Electricity Prices, IEEE Transactions on Power Systems, Vol.18, No.3, pp.1040-1020, Aug 2003.
3. D.C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole: Research Challenges in Environmental Observation and Forecasting Systems. Proceedings of the 6th annual international conference on Mobile computing and networking, pages 292-299, Boston, Massachusetts, USA.
4. T. Niimura and H.S. Ko: A Day-ahead Electricity Price Prediction Based on a Fuzzy-neural Autoregressive Model in a Deregulated Electricity Market. Proc. the 2002 International Joint Conf. on Neural Networks (IJCNN'02), Vol.2, No. 12-17, pp.1362-1366, May 2002.
5. J.-J. Guo and P.B. Luh: Selecting Input Factors for Clusters of Gaussian Radial Basis Function Networks to Improve Market Clearing Price Prediction, IEEE Trans on Power Systems, Vol.18, No.2, pp.665-672, May 2003.
6. H.S. Guirguis, and F.A. Felder: Further Advances in Forecasting Day-Ahead Electricity Prices Using Time Series Models, KIEE International Transactions on PE, Vol 4-A No. 3 pp. 159-166, 2004.
7. K.-R. Nuller, A. J. Smola, G. Ratsch, b. Scholkopf, J. kohlmorgen, V. Vapnik: Predicting time series with support vector machine, in Proc. of ICANN'97, Springer LNCS 1327,1997. pp. 999-1004. 1997
8. Butler, K.L.; Momoh, J.A: Detection and classification of line faults on power distribution systems using neural networks. Proc. of the 36th Midwest Symp. on Circuits and Systems, vol. 1, 16-18 Aug. 1993 pp 368 - 371
9. H. Liu, L. Yu: Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 4, April, 2005.
10. Nealand, J.H.; Bradley, A.B.; Lech, M: Discriminative feature extraction applied to speaker identification, 6th Int. Conf. on Signal Processing, 2002 Vol. 1, 26-30 Aug. 2002 pp: 484 - 487.
11. A.C. Tamhane, D.D. Dunlop: Statistics and Data Analysis: from Elementary to Intermediate. Upper Saddle River, NJ: Prentice Hall, c2000.
12. A.L. Blum and P. Langley: Selection of Relevant Features and Examples in Machine Learning, Artificial Intelligence, vol. 97, pp. 245-271, 1997.
13. M. Dash, K. Choi, P. Scheuermann, and H. Liu: Feature Selection for Clustering-a Filter Solution, Proc. Second Int'l Conf. Data Mining, pp. 115-122, 2002.

14. M.A. Hall: Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning, Proc. 17th Int'l Conf. Machine Learning, pp. 359-366, 2000.
15. M. Ben-Bassat: Pattern Recognition and Reduction of Dimensionality, Handbook of Statistics-II, P.R. Krishnaiah and L.N. Kanal, eds., pp. 773-791, North Holland, 1982.
16. K. Kira and L.A. Rendell: The Feature Selection Problem: Traditional Methods and a New Algorithm, Proc. 10th Nat'l Conf. Artificial Intelligence, pp. 129-134, 1992.
17. R. Kohavi and G.H. John: Wrappers for Feature Subset Selection, Artificial Intelligence, vol. 97, nos. 1-2, pp. 273-324, 1997.
18. W.L. Martinez, A.R. Martinez: Computational Statistics Handbook with MATLAB. Boca Raton: Chapman & Hall/CRC, c2002.
19. Oja, E.; Kiviluoto, K.; Malaroui, S: Independent component analysis for financial time series. IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC..
20. B.L. Zhang and Z.Y. Dong: An Adaptive Neural-wavelet Model for Short-Term Load Forecasting, Electric Power Systems Research, Vol. 59, pp.121-129, 2001.
21. H. Zhang, L.Z. Zhang, L. Xie, and J.N. Shen: The ANN of UMCP forecast based on developed ICA, IEEE International Conference on Electric Utility Deregulation, Reconstruction, and Power Technologies (DRPT 2004) April 2004, Hongkong.
22. Petter Skantze, Andrej Gubina, and Marija Ilic: Bid-based Stochastic Model for Electricity Prices: The Impact of Fundamental Divers on Market Dynamics. Report produced by Energy Laboratory, MIT, [Online] Available: <http://fee.mit.edu/public/el00-004.pdf>.
23. J. W. Han, M. Kamber: Data Mining: Concepts and Techniques, San Francisco, Calif.: Morgan Kaufmann Publishers, 2001.
24. H. Liu and H. Motoda: Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic, 1998.
25. I.H. Witten, E. Frank: Data mining: practical machine learning tools and techniques (2nd edition). San Francisco, Calif.: Morgan Kaufman, 2005.
26. J. Yang and V. Honavar: Feature Subset Selection Using a Genetic Algorithm, Feature Extraction, Construction and Selection: A Data Mining Perspective, pp. 117-136, 1998, second printing, 2001.
27. Garcia, G.N.; Ebrahimi, T.; Vesin, J.-M: Support vector EEG classification in the Fourier and time-frequency correlation domains. Neural Engineering, 2003. Conference Proceedings. First International IEEE EMBS Conference on, 20-22 March 2003 Page(s):591 - 594.
28. T. Sakai: Average gain ratio: a simple retrieval performance measure for evaluation with multiple relevance levels. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, July 2003.
29. V. Vapnik: The Nature of Statistical Learning Theory, Springer Verlag, New York, 1995.

On Similarity of Financial Data Series Based on Fractal Dimension^{*}

Jian-rong Hou¹, Hui Zhao², and Pei Huang¹

¹ School of Management, Shanghai Jiaotong University
Shanghai, China, 200052
hzhao@sei.ecnu.edu.cn

² Software Engineering Institute, East China Normal University
Shanghai, China, 200062
zhaohui@fudan.edu.cn

Abstract. Financial time series show the non-linear and fractal characters in the process of time-space kinetics evolution. Traditional dimension reduction methods for similarity query introduce the smoothness to data series in some degree. In the case of unknowing the fractal dimension of financial non-stationary time series, the process of querying the similarity of curve figure will be affected to a certain degree. In this paper, an evaluation formula of varying-time *Hurst* index is established and the algorithm of varying-time index is presented, and a new determinant standard of series similarity is also introduced. The similarity of curve basic figure is queried and measured at some resolution ratio level. In the meantime, the fractal dimension in local similarity is matched. The effectiveness of the method is validated by means of the simulation examples.

1 Introduction

The similarity query of time series has found important applications in discovering the similar price behavior of stocks^[1]. At present, the similarity pattern query about time series is the research hotspot in knowledge discovery^[2]. Because there are more sampling points of original time series and more series in the series database, the research emphasis is how to quicken the process of querying the similarity to solve the problem of realizing the best series pattern matching.

The common method on similarity query is the dimensionality reduction technique (the dimension is defined as the number of time and space sampling which distinguishes from the fractal dimension in this paper). The representative research work include: the *F-Index* method based on discrete *Fourier* transformation (DFT) by *Agrawal Rakesh*^[2], the *Karhuen Loeve*(K-L) transformation method by *Wu Dannie*^[3], *Keogh Eamon* presented the linear division method, in which the complex curve subsection is represented as the straight line. The probability is used to query similarity after the series is represented again by the *Keogh Eamonn* method^{[4],[5]}. The recent

^{*} Supported by the National Natural Science Foundation of China under Grant No. 70371042, and Post-doctor Science Foundation of China under Grant No. 200303310.

research about dimension reduction is the time series similarity matching based on wavelet transformation presented by *Chan Franky* and *Zheng Chen*^{[6],[7]}. They used Euclidean distance standard and L-shift Euclidean distance standard as the judgement standard of series similarity respectively. The details are eliminated from the curve in order to distil the basic shape of series curve. Ordinarily, the time series reflected from an object's evolution of time-space kinetics is of non-linearity and fractal character (ragged irregularity of series). Almost all time series are stochastic non-stationary time series in nature^[8], similarity of which means similarity in statistic features; Aforementioned dimension reduction methods about similarity query introduce the smoothness to data series in some degree that the important features of time series about non-linearity and fractal are destroyed. Thus the local error of similarity matching increases.. In addition, previous work on similarity of sequence data mainly considers finding global patterns^{[9],[10]}. In this paper we think that the similarity of random non-stationary time series data shows the local similarity much more and a time series may be similar to the local shape of other sophisticated series. The similarity of basic curve figure is queried and measured at some resolution ratio level. In the meantime the fractal dimension in local similarity is matched. The fractal determinate dimension value educed by scale relation can not depict the space-time kinetics process of object evolution completely yet, except that it can reflect the self-similarity construction rule of static structure. We put forward a new concept---varying-time dimension function $D(t)$ in this paper in order to describe the phenomena of evolution along with time more sufficiently and completely. At the same time we note that *FBM* models stochastic process and its correlative increment process are ordinarily non-stationary because the *Hurst* index in the stochastic process with local self-similarity is varying-time. Wavelet analysis has been proven to be a very efficient tool in dealing with non-stationary and self-similarity^[11]. Thus, wavelet transformation will act a leading actor in the process of evaluating varying-time index.

The main work in this paper is organized as follows: the local self-similarity stochastic process definition of non-linear time series based on statistic self-similarity is presented in section 2. The original *FBM* model is rebuilt by introducing varying-time *Hurst* index. It utilizes *Daubechies* wavelet to transform the local self-similarity process and establishes an evaluation expression of *Hurst* index by the least square method. The algorithm of gaining the varying-time *Hurst* index is introduced in section 4. The effectiveness of the method is validated by simulation examples in the last section.

2 Wavelet Evaluation of Hurst Index Based on Financial Non-stationary Time Series

The following stochastic process $Y(t)$ of integral form can be regarded as a generalized fractal *Browian* motion (GFBM) of *FBM*, which includes a varying-time fractal parameter $H(t)$.

$$Y(t) = \int_{-\infty}^0 \left[(t-u)^{H(t)-\frac{1}{2}} - (-u)^{H(t)-\frac{1}{2}} \right] dB(u) + \int_0^t (t-u)^{H(t)-\frac{1}{2}} dB(u) \quad (1)$$

where real number $t \geq 0$, $B(t)$ is standard *Brownian* motion. $H(t) \in (0,1)$.

Let $Y(t)$ be a stochastic process with zero mean value. If its covariance $\Gamma_t(s_1, s_2)$ satisfies the following expression

$$\Gamma_t(s_1, s_2) - \Gamma_t(0,0) = -q(t)|s_1 - s_2|^{2H(t)} \{1 + o(1)\}, \quad (|s_1| + |s_2| \rightarrow 0) \quad (2)$$

where $q(t) \geq 0$, then $Y(t)$ is called a stochastic process with local self-similarity. For the given $|s_1 - s_2|$, local self-relativity of process $Y(t)$ will also wear off when $H(t)$ decreases from one to zero. When varying-time index is smooth, the covariance function $\Gamma_t(s_1, s_2)$ in *GFBM* model (1) satisfies formula (2). So $Y(t)$ presents local self-similarity behavior.

Let $\psi(x)$ be the mother wavelet. $WY(a, t)$ is the wavelet transformation about self-similarity process $Y(t)$ at scale a and position t . Well then,

$$WY(a, t) = a^{-\frac{1}{2}} \int \psi\left(\frac{u-t}{a}\right) Y(u) du = a^{\frac{1}{2}} \int \psi(x) Y(t + ax) dx$$

It is deduced by the formula (2) and the above formula that

$$\begin{aligned} E\left[|WY(a, t)|^2\right] &= a^{-1} \iint \psi\left(\frac{u-t}{a}\right) \psi\left(\frac{v-t}{a}\right) E[Y(u)Y(v)] dudv \\ &= C_1 a^{1+2H(t)} \quad (a \rightarrow 0) \end{aligned} \quad (3)$$

where

$$C_1 = -q(t) \iint |x - y|^{2H(t)} \psi(x) \psi(y) dx dy$$

$$\text{Let } y_t(a) = \log|WY(a, t)|^2, C_2 = E\left\{\log\left[\frac{|WY(a, t)|^2}{E(|WY(a, t)|^2)}\right]\right\}$$

$$\varepsilon_{t(a)} = \log\left\{\frac{|WY(a, t)|^2}{E(|WY(a, t)|^2)}\right\} - E\left\{\log\left[\frac{|WY(a, t)|^2}{E(|WY(a, t)|^2)}\right]\right\}$$

A regression model can be gained by (1) and (2) when a is very small:

$$y_t(a) \approx (\log C_1 + C_2) + [2H(t) + 1] \log a + \varepsilon_t(a) \quad (4)$$

A small-scaled series is constructed as follows.

$$a_1 > a_2 > \dots > a_L, a_j = 2^{-j}, j = 1, 2, \dots, n.$$

Let $x_j = \log a_j, y_j = y_t(a_j), j = 1, 2, \dots, n$. The least square method is used to get an evaluator of $H(t)$ in formula (4) for the couples $\{(x_j, y_j), j = 1, 2, \dots, n\}$:

$$\hat{H}(t) = \frac{1}{2} \left[\frac{\sum (x_j - \bar{x})(y_j - \bar{y})}{\sum (x_j - \bar{x})^2} - 1 \right] \tag{5}$$

where

$$\bar{x} = \sum x_j / n, \bar{y} = \sum y_j / n$$

It can be proved that $\hat{H}(t)$ is a consistent result^[13] of $H(t)$.

3 Algorithm Description

Now let us observe a stochastic time series process $Y(t)$ on the discrete and equally spaced points.

The time points may be limited in $[0,1)$. The sample size is $2^J, t_i = (i-1)/n, n = 1, 2, \dots, 2^J$. $y_{j,k} (k = 0, 1, \dots, 2^{j-1}, j = 0, 1, \dots, J-1)$ is an evaluated value of $WY(2^{-j}, K2^{-j})$. The latter is the discrete value by wavelet transformation $\hat{H}(t)$ in $a = 2^{-j}, t = K2^{-j}$. Wavelet transformation is carried on by *Daubechies'* compactly-supported wavelet bases with M moments^[13].

Step 1: $[0,1)$ is partitioned into 2^l equal-length sub-section I_m without interacting each other.

$$I_m = [(m-1)2^{-j}, m2^{-j}), 1 \leq l \leq (J-1), m = 1, 2, \dots, 2^l.$$

Step 2: $\hat{H}(t)$ is regarded as the average value of $H(t)$ in the correspond sub-section I_m . The time spot of $\hat{H}(t)$ is chosen at the point $2^{-l-1}(2m-1)$ in the middle of I_m . The double variables set is defined as follows:

$$\{(X_m, Y_m)\} = \left\{ \left[\log(2^{-j}), \log(|y_{j,k}|^2) \right] k2^{-j} \in I_m \right\}$$

$$0 \leq k \leq 2^j - 1, 0 < j \leq J - 1 \tag{6}$$

$\hat{H}(t)$ is evaluated by formula (5) on each I_m .

Step 3: The evaluated value of $\hat{H}(t)$ is smoothed by using local multinomial to form a curve that can be regard as the approach of the real figure of $H(t)$.

4 Determinant Standards of Similarity

Definition 1: Given two thresholds $\varepsilon_i (i = 1, 2)$ and two time series $\vec{X} = \{x_i\}_{i=0,1,\dots,n}$ and $\vec{Y} = \{y_i\}_{i=0,1,\dots,n}$ with same length n whose fractal dimension functions are $D_1(t)$ and $D_2(t)$ respectively where $D_i(t) \in C[a, b], i = 1, 2$. When the following two inequations are satisfied at the same time the two series \vec{X} and \vec{Y} are similar.

$$d_1(\vec{X}, \vec{Y}) = \left(\sum_{i=0}^{n-1} (y_i - x_i)^2 \right)^{\frac{1}{2}} \leq \varepsilon_1 \tag{7}$$

$$d_2(D_1(t), D_2(t)) = \text{Max}_{a \leq t \leq b} |D_1(t) - D_2(t)| \leq \varepsilon_2 \tag{8}$$

where

$d_1(\vec{X}, \vec{Y})$ is the Euclidean distance,

$d_2(D_1(t), D_2(t))$ is the measurement of function $D_1(t)$ and $D_2(t)$.

Lemma: Given two time series \vec{X} and \vec{Y} with same length n . The two new series \vec{S}_J and \vec{T}_J are obtained after \vec{X} and \vec{Y} are transformed respectively by wavelet in the layer J , then

$$d(\vec{S}_J, \vec{T}_J) \leq d(\vec{X}, \vec{Y}). \tag{9}$$

5 Simulation Results

In order to evaluate the effectiveness of the proposed method in searching the similarity of any two stochastic non-stationary time series, we chose two time series samples from HSI in stock market. The result reported in this section addresses in the following issues:

- The introduction of time-varying fractal dimension (or varying-time Hurst index) can depict the non-linear irregularities of stochastic non-stationary time series.
- The similarity of basic shape, in the meantime, the similarity of fractal feature curve of non-stationary time series data is considered. new standard of similarities proposed in the above chapter meets the needs of
- Local similarity between one data series and another one

Here are the two original non-stationary time series samples in Fig.1 and Fig. 2, they indicate the change of HSI in the two different periods.

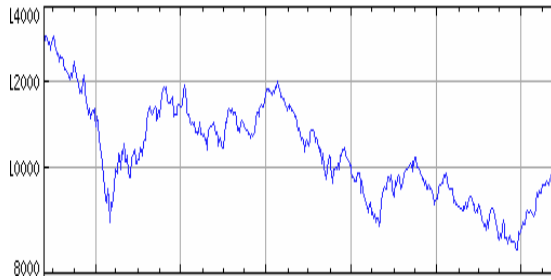


Fig. 1. change curve of HSI in the period I

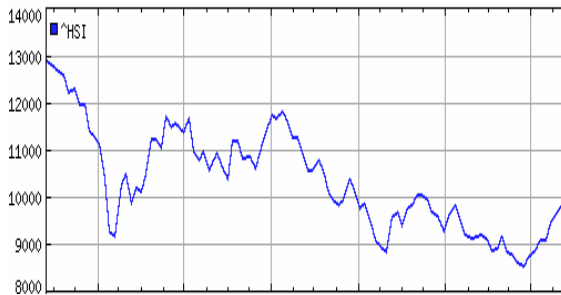


Fig. 2. change curve of HSI in the period II

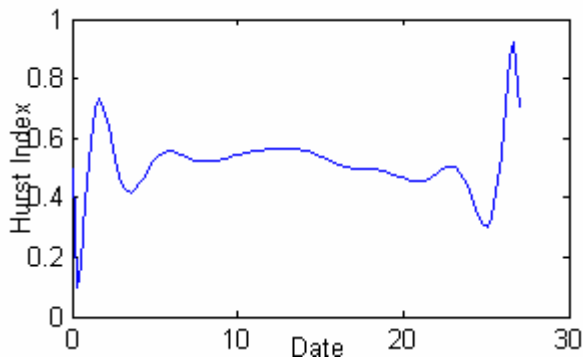


Fig. 3. Hurst index curve of time series from Fig.1

Here we adopt wavelet base **db4**. $\varepsilon_1 = 0.04$. Fig.3 and Fig.4 represent Hurst index curves from Fig.1 and Fig.2 respectively. Evolution of time-varying *Hurst* index is of great importance in stock investment strategies. Two hundred and fifty points are selected in Fig.1 and treated with wavelet base **db4**. $J = 8$. Hurst index discrete values are calculated by formula (5) and smoothed by a polynomial with eight order. So does

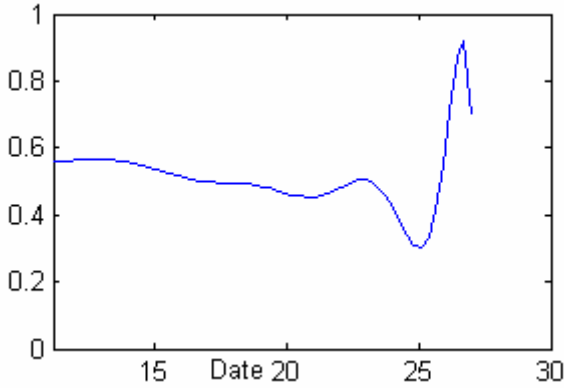


Fig. 4. Hurst index curve of time series from Fig.2

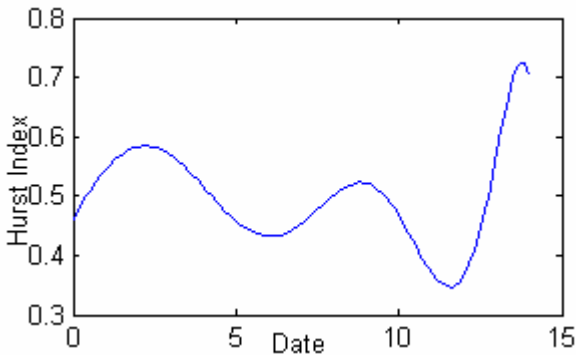


Fig. 5. Hurst index curve of time sub-series from Fig.5

Fig.2. Fig.5 is a segment cut from Fig.3. Hurst index curve in Fig.5 is similar to one in Fig.4 when $\epsilon_2 = 0.02$, that is, (nearly fractal character of data from Fig.2 approaches to that of data from local sub-series of Fig.1.)

The above two kinds of similarity matching can be combined to depict the similarity of dynamic characters of data from two time series.

6 Conclusion

We proposed a new standard of financial series similarity, by which the similarities of dynamic characters of data from two time series can be completely depicted. The similarity of curve basic figure is queried and measured at some resolution ratio level; in the meantime, the fractal dimension in local similarity is matched. We put emphasis on algorithm and matching of fractal time-varying Hurst index curves. The effectiveness of the method is validated by means of the simulation example in the end.

The work of this paper supplements the similarity mentioned in the literatures^{[6],[7]}.

References

1. Agrawal Rakesh, Faloutsos Christos, Swami Arun, Efficient similarity search in sequence databases[C]. Proc. of the 4th Conference on Foundations of Data Organization and Algorithms, Chicago, Oct. 1993, P69~84
2. CHEN M. S., HAN J., YU P.S. Data Mining: an overview from a database perspective[J]. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(6), P866~883
3. Daniel, Agrawal Divyakant, Abbadi Amr EI, Singh Ambuj k, Smith Terence R., Efficient retrieval for browsing large image database[C]. In Proceedings Conference on Information and Knowledge Management, 1996, P11~18
4. Keogh Eamonn, Padhraic Smyth., A probabilistic approach to fast pattern matching in time series databases[C]. Proceedings of the Third Conference on Knowledge Discovery in Database and Data Mining, 1997
5. Keogh Eamonn, Michael J. Pazzani, An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback[C]. Proceedings of the 4th Conference on Knowledge Discovery in Database and Data Mining, AAAI Press, 1998, 239~241
6. CHAN Franky, FU Wai-chee, Efficient time series matching by wavelets[C]. 15th IEEE International Conference on Data Engineering, Sydney, Australia, March 23~26, 1999, P126~133
7. ZHENG Cheng, OUYAN Wei-ming, CAI Qing-sheng, An Efficient dimensionality reduction technique for times series data sets[J]. Mini-Macro System, Vol.23, No.11, Nov. 2002, 1380~1383
8. P. J. Brockwell, Time series: Theory and Methods, New York: Springer-Verlag, 1991
9. Wang K. Discovering patterns from large and dynamic sequential data. Special Issues on Data Mining and Knowledge Discovery, Journal of Intelligent Information Systems, 1997, 9(1): 8~33
10. Kam P, Fu AWC. Discovering temporal patterns for interval-based events. In proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovering (DaWaK2000). UK, 2000
11. B. B. Mandelbrot and J. W. Van. Ness, Fractional Brownian motions, fractional noises and applications, SIAM Review 10, 4, 422-437, 1968
12. Hou Jianrong, Song Guoxiang. Application of Wavelet Analysis in the Estimation of Hurst Index. Journal of Xidian University (Science Edition), 2002, No. 1
13. Inrid. Daubechies, The wavelet transform: Time-Frequency Localization and Signal Analysis IEEE. Trans. On Information Theory, 36(5), 1990

A Hierarchical Model of Web Graph^{*}

Jie Han¹, Yong Yu¹, Chenxi Lin², Dingyi Han¹, and Gui-Rong Xue¹

¹ Shanghai Jiao Tong University, No. 800, Dongchuan Road, Shanghai, 200240, China

² Microsoft Research Asia, No. 49, Zhichun Road, Beijing, 100080, China

{micro_j, yyu, handy, grxue}@sjtu.edu.cn,
chenxil@microsoft.com

Abstract. The pages on the World Wide Web and their hyperlinks induce a huge directed graph – the Web Graph. Many models have been brought up to explain the static and dynamic properties of the graph. Most of them pay much attention to the pages without considering their essential relations. In fact, Web pages are well organized in Web sites as a tree hierarchy. In this paper, we propose a hierarchical model of Web graph which exploits both link structure and hierarchical relations of Web pages. The analysis of the model reveals many properties about the evolution of pages, sites and the relation among them.

1 Introduction

The Web pages and the hyperlinks among them has formed a huge directed graph where nodes represent Web pages and directed edges represent hyperlinks. As time goes by, the graph is evolving. Its scale increases explosively. It also contains a large set of link information which has led to innovations in information retrieval. For example, the link analysis methods like HITS algorithm[1] and PageRank algorithm[2]. Mathematical modelling is a powerful tool of studying the Web Graph.

The traditional random graph model (Erdős-Rényi model) $G_{n,p}$ [3] does not describe the Web graph well. For example, graphs generated by this model do not contain very popular pages; their in-degree distribution is binomial, which is far from the power-law distribution observed by former researches. A large number of models, including ACL model[4], LCD model[5] and Copying model[6] have been proposed to explain static and dynamic properties of the Web graph. These models pay much attention to four most important properties of the graph observed[7]: the evolving property, the power-law degree distribution (especially in-degree distribution), the small world property and the bipartite cores.

In fact, the Web graph contains not only the pages and hyperlinks among them, but also other elements. The existing models just model the link graph, and ignore other valuable relations among pages such as the hierarchical structure of Web pages on Web sites. In this paper we propose a class of hierarchical models, which include the hierarchical relations among pages as well as the link structure. With the help of hierarchical relations, we can define Web sites in the Web graph

^{*} This project is supported by National Foundation of Science of China (No.60473122).

models. The models assign a parent for each new page, and the graph induced by the relationship *parent* appears as a forest. Each tree of the forest is defined as a site. From the tree hierarchy, we can clarify how pages are organized. Further analysis of the model shows that the model meets most observations from the former research. With this model, we predict some properties about the pages, sites and their relationship.

The rest of the paper is organized as follows. In section 2, we review some related work. In section 3, we show the definition of hierarchical model. In section 4, we analyze the hierarchical model in detail. Finally we conclude in section 5.

2 Related Work

The most important properties of the Web graph are the power-law degree distributions. A series of integer S follows a power-law distribution, if the occurrence of k in S is in proportion to $k^{-\beta}$ for some $\beta > 1$. In the observation of Kumar et al.[8] and another experiment of Broder et al.[9], the β of in-degree is a remarkably constant, 2.1. In the former research of Bharat et al.[10], the in-degree and out-degree of the sites also follows a power-law distribution respectively, which is also observed by our research of Web graph in China[11]. Site page numbers also follow a power-law distribution. In our experiment, it is around 1.74 in Web graph in China[11]. The *small-world graphs* were first introduced in the study of social networks by Strogatz and Watts[12]. Its most important feature is the short average path length. Broder et al. reported that the average path length in their data set is 16[9], and the result is a bit smaller than the prediction of 19 from Albert et al.[13]. The density of bipartite cores is another interesting feature of the Web graph. The graph contains much more small bipartite cores than common graphs with the same scale. Kumar et al.[8] supposed that the Web communities in the Web are characterized by these bipartite cores .

A large number of models for the Web graph have been proposed, among which the most popular ones are ACL model[4], LCD model[5] and Copying model[6]. ACL model by Aiello et al. induced a kind of random graph with fixed power-law distribution of degree, which captured the power-law distribution property. LCD model by Bollobás introduced the method of preferential attachment, which is one of the most popular method to generate graphs with power-law degree distribution. Using the preferential attachment method, the terminal of a new coming edge is chosen according to the in-degree of each node. This mechanism brought the *rich gets richer* phenomenon and finally generate the power-law degree sequence. Copying model by Kumar et al. assumed that the creation of new node contains a copying process from an existing node. This mechanism generates the graph with a large number of bipartite cores as well as power-law degree distribution.

3 A Hierarchical Model

The existing models pay much attention to the evolution of the pages. At each time slot, only the new coming page and its out-links are taken into consideration.

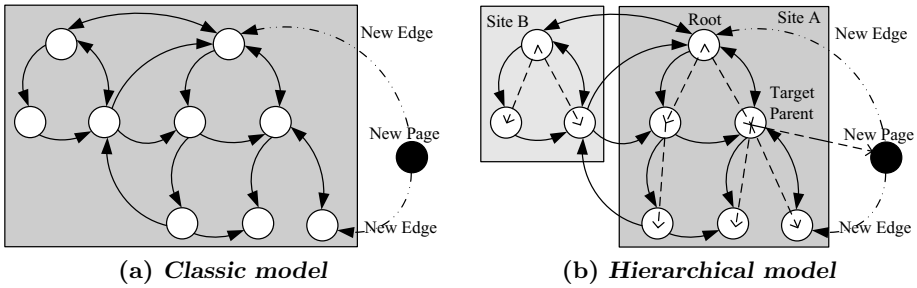


Fig. 1. Illustration of the Web models

As a result, the models only create the link graph of the Web, and neglect other relations among Web pages. Although the graphs generated meets many properties of Web graphs, they still cannot explain some very basic properties (for example, the structure of URLs) of the Web.

Actually, the organization of pages in each host appears to be a tree hierarchy rather than a graph structure. Unfortunately, most of existing models do not capture this property. For this reason, we propose a model to bring the hidden hierarchy into Web graph models, combined with the link structure.

Figure 1 illustrates the difference of existing Web graph models and the hierarchical models. All these models generate link graphs denoted by solid edges. The hierarchical models generate another graph denoted by dotted edges, which represents the parent-child relations among pages. From Figure 1(b), we can immediately obtain how the Web pages are organized (*locality*[14]) by the sub-graph generated by the dotted edges. According to the hierarchy, each tree can be defined as a *Web site*. The root node of each hierarchy tree denotes the root page of the site. Each node can be identified with its path from the root together with the id of its owner tree. With the help of the hierarchy structure, we can analysis and predict more properties about the locality of pages on the Web.

3.1 The Linear Hierarchical Model

In this paper, we analyze a simple version of the hierarchical models. In this model, the time goes in a linear way. i.e., we consider each creation of new page as a time slot, and pages are created in separate time slots. For convenience, we only model the creation of pages and do not care about the death and update of pages and hyperlinks. In this model, V represents the set of nodes, E represents the set of edges. Nodes are equivalent to pages and edges are equivalent to hyperlinks. In the rest of the paper, N_v means the number of pages, N_e represents the number of edges and $d^-(x)$ denotes the in-degree of node x , they are changing as time goes by.

This model is an evolving model based on preferential attachment. At time slot 1, the graph starts evolving with only one node and $d+1$ self loops, where d is a constant parameter which will be explained later. This node represents a page whose parent is a void page **nil**. It is also the *root page* of a site.

At each time slot, a new page is created and added to the graph. With probability ε , the page belongs to a new site and its parent is **nil**. Add a self loop for it to enable the simplest preferential attachment formula. With probability $1 - \varepsilon$, the page belongs to an existing site. Its parent is chosen from the existing pages by a *normalized preferential function* $f(x)$. It is assigned to the same site as its parent. Also, an edge is added to link the parent and the new page, from which Internet users can access the new pages by random surfer. In this simple version of the model, only the parent introduces the new page to the Web users.

Then d edges are added from the new page to the existing ones. To add an edge, we use the following method: with probability η , choose a target page in the **same site** by normalized *preferential function* $f(x)$; with probability $1 - \eta$, choose a target page from all the existing pages by normalized *preferential function* $f(x)$. We call the edges whose terminals are in the same site *local edges*, and call other edges *remote edges*.

The model produces two graphs. One is the Web graph constructed by the pages and the hyperlinks. The other is the hierarchical forest of pages on the relation *parent*, in which each tree represents a Web site. And, if we contract the pages of the same site to a node, the Web graph becomes a new weighted directed graph, the *host(site) graph*.

There are several parameters for the linear hierarchical model. ε is the *site factor*. It represents the creation rate of new sites which controls the average page number of sites. d is the out-link number of a new page. It is an important part of the out-degree distribution. η is the *locality factor*. It approximately determines the ratio of local edges and remote edges. $f(x)$ is a *preferential function*, which is the key of *preferential attachment mechanism*. The simplest $f(x)$ is $f(x) = d^-(x)$. When choosing a page from the set V , each page x_0 has the probability: $\frac{f(x_0)}{\sum_{x_i \in V} f(x_i)}$ to be linked. This formula describes that the probability of a new reference to a page is in proportion to its current popularity, i.e. its in-degree.

4 The Analysis of the Hierarchical Model Properties

The hierarchical model pays much attention to the hierarchical relations among pages. In this paper, we analyze the degree distribution, the site graph and the locality properties under the hierarchical structure.

4.1 Degree Distribution

In this section, we will analyze the degree distributions, especially the in-degree distribution, because it can be regarded as the popularity of a node.

Theorem 1. *The pages' in-degree series of the linear hierarchical model approximately follows a power-law distribution with the exponent $2 + \frac{1}{d}$.*

Proof. We use k_i to represent the expectation of $d^-(x_i)$. In the analysis, they are assumed as real numbers. We consider each addition of a new node as a time slot. What we care is the in-degree increase of the existing nodes.

If the new edge is a remote edge, the target page is chosen by $f(x)$. If it is a local one, the target page is chosen by the following process. First, each site S has the probability $\frac{\sum_{x_i \in S} f(x_i)}{\sum_{x_i \in V} f(x_i)}$ to be selected as the owner site of the new page. A page x_0 in S has the probability $\frac{f(x_0)}{\sum_{x_i \in S} f(x_i)}$, so the probability for a page to be chosen as a destination of local edge is $\frac{f(x_0)}{\sum_{x_i \in S} f(x_i)} \frac{\sum_{x_i \in S} f(x_i)}{\sum_{x_i \in V} f(x_i)} = \frac{f(x)}{\sum_{x_i \in V} f(x_i)}$. It is the same form as those in the case of remote edges. To sum up the two cases, we have the following equation:

$$\frac{\partial k_i}{\partial t} = d \frac{f(x)}{\sum_{x_i \in V} f(x_i)} = \frac{dk_i}{N_e} = \frac{dk_i}{(d+1)t} \tag{1}$$

Solve the equation (1) with the initial condition $k_i(t) = 1$, we obtain that $k_i = Ct^{\frac{d}{d+1}}$, where $C = t^{-\frac{d}{d+1}}$. Now consider $P(k_i(t) < k)$.

$$P(k_i(t) < k) = P(Ct^{\frac{d}{d+1}} < k) = P(t > k^{-\frac{d+1}{d}}t) \tag{2}$$

Note that the time goes in a linear way in this model, so $P(t > x) = 1 - \frac{x}{t}$.

We further obtain the degree distribution:

$$P(k_i(t) = k) = \frac{\partial P(k_i(t) < k)}{\partial k} = (-k^{-\frac{d+1}{d}})' = \frac{d+1}{d} k^{-2-\frac{1}{d}} \tag{3}$$

This implies that $P(k_i(t) = k) \propto k^{-2-\frac{1}{d}}$ □

In the real world, some research reported that d should be a bit larger than 7 [9]. From the proof we know that the exponent will be approximately 2.14 if $d = 7$. This matches the reality very well.

Theorem 2. *The sites' in-degree series of the linear hierarchical model approximately follows a power-law distribution with the exponent 2.*

Proof. We use s_i to represent the expectation of $d^-(s_i)$. They are assumed as real numbers. We still consider each coming of a new node as a time slot. Using an analysis similar to that in the proof of Theorem 1, with the minor probability of new site ignored, we get $\frac{\partial s_i}{\partial t} = (d+1) \frac{s_i}{\sum_{x_i \in V} f(x_i)} = \frac{(d+1)s_i}{N_e} = \frac{s_i}{t}$. Conduct an analogous calculus in Theorem 1, we obtain $P(s_i(t) = k) \propto k^{-2}$. □

In this model, the pages with a popular owner site may get more chance to be linked by new coming pages, thus they are prone to become popular. On the other hand, the popularity of pages in a site make their distribution to their own site. It is a process of mutual enhancement.

4.2 Site Size

As a fact, the page number in a site (*site size*) follows a power-law distribution as well. We can also proof it by analyzing the growth of the graph.

Theorem 3. *The site size of the linear hierarchical model approximately follows a power-law distribution with the exponent 2.*

Proof. Let ns_i represent the expectation of the number of pages in site i . Each time a node is added into the graph, the parent is chosen by the preferential function. So $\frac{\partial ns_i}{\partial t} = \frac{s_i}{(d+1)t} = C'$ where C' is some constant. So $ns_i = C't \propto s_i$. This equation means that ns_i is in proportion to s_i .

By Theorem 2, $P(s_i(t) = k) \propto k^{-2}$, so $P(ns_i(t) = k) \propto k^{-2}$. □

From the proof, we found that the size of a site may be in proportion to its popularity. A popular site is more possible to become large.

4.3 Locality

In the previous section we analyzed the degree sequence. In this section, we will study the *locality* of the model. More emphasis will be paid on the hierarchy tree and the local edges in the model.

Theorem 4. *The pages' children number of the linear hierarchical model approximately follows a power-law distribution with the exponent $2 + \frac{1}{d}$.*

Proof. Let nc_i represent the expectation of the children number of node i . Each time we add a node into the graph, the parent is chosen by the preferential function. So $\frac{\partial nc_i}{\partial t} = \frac{k_i}{(d+1)t}$, $nc_i = C't^{-\frac{1}{d+1}}$ where C' is some constant. Solving the equation, we obtain $nc_i = t^{\frac{d}{d+1}} \propto k_i$. This equation means that nc_i is in proportion to k_i . By Theorem 1, $P(k_i(t) = k) \propto k^{-2-\frac{1}{d}}$.

Thus, $P(nc_i(t) = k) \propto k^{-2-\frac{1}{d}}$. □

Theorem 4 reveals the fact that popular pages will have more children pages.

Also, the out-degree in the simple model is just an addition of d and the *introducing edges* which links the from parent to child. So out-degree also follows a power-law distribution, except the bias in small data. And, if we use a distribution $D(\mu, \sigma)$ instead of d , the graph generated by the model will be much closer to the real world, and the bias in small data of out-degree which is observed by many research can also be explained.

In each Web site, the root page is prone to be the most popular page. In the previous section, we have mentioned the expectation of the in-degree of page and site. The root page has the same initial condition as its owner site. So the proportion of the in-degree owned by the root page of a site is $t^{\frac{d}{d+1}}$.

To extend the concept, we introduce another property of page named *level*. If we use $level(i)$ to represent the level of node i , it will be formularized as :

$$level(i) = \begin{cases} 0 & \text{if } \exists s, i = root(s) \\ level(parent(i)) + 1 & \text{otherwise} \end{cases}$$

The in-degree and number of pages possessed by each level are also interesting. We use the notion $l(i)$ and $lc(i)$ to represent the expectation of page in-degree and page number possessed by level i respectively. For convenience, the following analysis will ignore the minor probability of creating new sites.

Theorem 5. *In a specified time slot t , $l(i) \propto \frac{\lambda^i}{i!}$ with $\lambda = \frac{\ln(t)}{d+1}$, so $l(i)$ is in proportion to a poisson distribution.*

Proof. The edges in the graph can be divided into two groups. One is the edges added from the new node, the other is the edges added from the parent to the new node.

Suppose $l'(i)$ is the sum of in-degree possessed by the nodes with level no greater than i , we can get:

$$\frac{\partial l'(i)}{\partial t} = \begin{cases} \frac{dl'(i)}{N_e} & \text{if } i = 0 \\ \frac{dl'(i)+l'(i-1)}{N_e} & \text{if } i > 0 \end{cases} \tag{4}$$

Solve the set of equations with the common initial condition $l'(i)|_{t=1} = d + 1$, we obtain:

$$l'(i) = \begin{cases} (d + 1)t^{\frac{d}{d+1}} & \text{if } i = 0 \\ (d + 1)\frac{1}{i!}\left(\frac{\ln(t)}{d+1}\right)^i t^{\frac{d}{d+1}} + l'(i - 1) & \text{if } i > 0 \end{cases} \tag{5}$$

This equation means:

$$l(i) = (d + 1)\frac{1}{i!}\left(\frac{\ln(t)}{d + 1}\right)^i t^{\frac{d}{d+1}} \propto \frac{\left(\frac{\ln(t)}{d+1}\right)^i}{i!} \tag{6}$$

□

Using an analogous method, we can also analyze the page number owned by each level. $lc(i)$ does not follow any general distributions. However, its approximate value in small data is in proportion to a poisson distribution. We describe it as the next theorem whose proof is similar to the previous one and omitted:

Theorem 6. *When t is large enough and i is small, $lc(i + 1) \propto \frac{\lambda^i}{i!}$, so $lc(i + 1)$ is approximately in proportion to a poisson distribution in small data, and λ is the same as that in Theorem 5.*

One application of Theorem 6 is the prediction of page number distribution of each directory level. If we regard depth of the pages' owner *directory* as the counterpart of the level of page in the model, by theorem 6, it is predicted that the series of pages number in a certain depth is approximately in proportion to a poisson distribution. Since no observation has been reported of this phenomenon, we measure the depth distribution of URLs in our data set of China Web Graph. The prediction was verified correct in our data set with a few bias.

Another application is the prediction of average path length in a site. In this simple model, root node reaches every node in the same site. The path length from root to a node i is less than $level(i)$. So the upper bound of average path length is the expectation of $level(i)$, $\frac{2\ln(t)}{d+1}$. In our observation in a simulation, the average path length is much less than $\frac{2\ln(t)}{d+1}$.

5 Conclusion

In this paper, we have proposed and analyzed a hierarchical Web Graph model. It concentrates on the pages, sites and their relations. Based on the model, we have proved that the in-degree series of both pages and sites, the sizes of sites and the children numbers of pages all follow a power-law distribution respectively. Additionally, we have proved that in a specified time slot, the expectation of page in-degree is in proportion to a poisson distribution. With the model, we also make a prediction that when the graph has evolved for a long period, the page number in a low-level Web directory also follows a poisson distribution. The prediction is also verified by our China Web Graph data. The model meets the situations on real Web well theoretically.

References

1. Kleinberg, J.: Authoritative sources in a hyperlinked environment. *Journal of the ACM* **46** (1999) 604–632
2. Sergey Brin, Lawrence Page, R., Winograd, T.: The pagerank citation ranking: Bring order to the web. Technical report, Computer Science Department, Stanford University (1998)
3. B.Bollobás: *Random Graphs*. Academic Press (1985)
4. Aiello, W., Chung, F., Lu, L.: A random graph model for massive graphs. *Proceedings of ACM Symposium on Theory of Computing* (2000) 171–180
5. B.Bollobás, O.Riordan, J.G.: The degree sequence of a scale-free random graph process. *Random Structures Algorithms* **18** (2001) 279–290
6. Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tomkins, A., Upfal, E.: Stochastic models for the web graph. In: *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, IEEE Computer Society (2000) 57
7. Bonato, A.: A survey of models of the web graph. *LNCS* **3405** (2005) 159–172
8. Kumar, R., Raghavan, P., Rajagopalan, S., Sivakumar, D., Tompkins, A., Upfal, E.: The web as a graph. In: *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. (2000) 1–10
9. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. *Comput. Networks* **33** (2000) 309–320
10. Bharat, K., Chang, B.W., Henzinger, M.R., Ruhl, M.: Who links to whom: Mining linkage between web sites. In: *Proceedings of the 2001 IEEE International Conference on Data Mining*. (2001) 51–58
11. Liu, G., Yu, Y., Han, J., Xue, G.: China web graph measurements and evolution. In: *Lecture Notes in Computer Science*. Volume 3399. (2005) 668 – 679
12. D.J.Watts, S.: Collective dynamics of ‘small-world’ networks. *Nature* **393** (1998) 440–442
13. Reka Albert, Hawoong Jeong, A.L.B.: Diameter of the world wide web. *Nature* **401** (1999) 130–131
14. Eiron, Nadav., M.K.S.: Locality, hierarchy, and birectionality in the web. In: *Second Workshop on Algorithms and Models for the Web-graph*. (2003)

Web Scale Competitor Discovery Using Mutual Information

Rui Li, Shenghua Bao, Jin Wang, Yuanjie Liu, and Yong Yu

Department of Computer Science and Engineering,
Shanghai JiaoTong University, Shanghai, 200240, P.R. China
{rli, lyj, shhbao, yyu}@apex.sjtu.edu.cn vicwang@sjtu.edu.cn

Abstract. The web with its rapid expansion has become an excellent resource for gathering information and people's opinion. A company owner wants to know who is the competitor, and a customer also wants to know which company provides similar product or service to what he/she is in want of. This paper proposes an approach based on mutual information, which focuses on mining competitors of the entity (such as company, product, person) from the web. The proposed techniques first extract a set of candidates of the input entity, and then rank them according to the comparability, and finally find and organize the reviews related to both original entity and its competitors. A novel system called "CoDis" based upon these techniques is implemented, which is able to automate the tedious process in a domain-independent and web-scale dynamical manner. In the experiment we use 32 different entities distributed in varied domains as inputs and the CoDis discovers 143 competitors. The experimental results show that the proposed techniques are highly effective.

1 Introduction

The data and information have become extraordinarily abundant nowadays due to the rapid development and relative maturity of the web! Prompt obtaining information, such as which company is or will be the enemy and what evaluation the customs in the same field make about, is definitely critical for one competitive company. Knowing about related or similar products which have the same features such as price, function and quality is also quite necessary to the customs who are about to purchase certain commodity. Given the fierce competition in the research field each researcher has to get the some idea about who is working in the similar fields. As has been noted, the process of locating and discover the entities (e.g. companies, products, persons) which are related or similar to original entity in some key aspects is defined as competitor discovery.

When one tries to discover the entity's competitor, he/she typically wants to get:

1. A list of the competitors' names ranked according to the comparability and related pages which describe each competitor.
2. The corresponding web pages which describe the comparable relation between original entity and its competitors, such as customs' review, news of their competition.

On the web, the most commonly used tools for getting information are search engines (e.g., Google¹, Yahoo!²). One first submits a search query representing the entity to a search engine system, which returns a set related web pages, and then he/she browses through the returned results to find information. But it's obviously a tedious process for finding competitors due to there are thousands of pages to be browsed through, and most of the pages only have information about the original entity. Especially in the case one knows little about the entity and people will never get the idea who will be the next competitor. To perform this novel task, we are confronted with several problems to deal with:

1. Ineffectiveness of the traditional method (i.e. sending the query containing only original entity name can not get pages covering both the entity and its competitors).
2. Lacking structure and noisy data in the web pages make it not easy to extract the competitor candidates.
3. Ambiguity of entity names in different context (i.e the same entity name may refer to distinguished entities (the two often belong to different fields)) and synonymous names in candidates list (i.e different names refer to the same entity).

In this paper, we propose a set of effective techniques to perform the task of extracting entity's competitors on the Web. The process starts with a search query representing the entity name and collects a set of ranking pages returned from search engines by the pre-defined query patterns (e.g., "Microsoft vs."). We call these pages informative pages, which may contain information of competitors. System use a set of linguistic patterns to extract competitors, which make use of the redundancy of the Web. It is unsupervised as it does not rely on any training data annotated by hand. Patterns used in our system do not rely on any domain-specific information, so it's domain independent. Since the information contained in the web pages is incremental, our system is dynamic. Then system rank the competitors according to the occurrence of competitor with the original entity using mutual information. Finally, system output a list of competitor items, each item representing a competitor, consists of the competitor name, competitor rank, home page of the competitor and related pages which describe the competition between the competitor and the original entity. We also proposed an effective method to deal with ambiguity problem by adding domain information.

The structure of this paper is organized as follows: Section 2 discusses some related work. Section 3 describes the techniques we proposed and ranking algorithms. Section 4 shows the empirical results of our experiments. Finally, we make concluding remarks in Section 5.

¹ <http://www.google.com/>

² <http://www.yahoo.com/>

2 Related Work

Zanasi[1] proposed the idea of competitive intelligence through data mining public resource such as web pages and public news. Some work [2,3] has been done and helps companies and individuals gain marketing information by mining on-line resource. All of these work focus on, however, mining opinions and extracting sentiment. None of them detect the comparable products or discover company's competitors. Our work aims to establish a domain independent system for discovering targeted entity's competitors. To the best of our knowledge, CoDiS is the first system for discovering competitors by mining Web resource. In following parts, we will discuss some techniques related to our work.

2.1 Entity Extraction

Many methods for the information extraction have been proposed since inception of the Web. Most of them have focused on the use of supervised learning techniques such as Hidden Markov Models, rule learning, or random fields [4,5]. These methods above nevertheless require a large set of training data, and the results are affected by the similarity between the test data and training data. The Wrapper-based approach including [6,7] is used for extracting information from highly structured documents. But it not suitable for the whole Web pages with the lack of structure.

Hearst [8] as well as Charniak and Berland [9] makes use of pattern-based approach to discover taxonomic and part-of relation from text respectively. Hahn and Schnattinger [10] also make use of such patterns and incrementally established background knowledge to predict the correct ontological class for unknown named entities appearing in a text. Recently a number of researchers solved different problems using such pattern based approach, including Ciniانو's PANKOW [11] system which aims to categorize instances with regards to a given ontology, and Etzioni's work [12] which established a domain independent's Information Extraction system. Liu [13] implemented a system extracting concepts and definitions. The core idea of such pattern-based approaches is that one may justify the ontological relationship with reasonable accuracy when he/she recognizes some specific idiomatic/syntactic/semantic relationship. Our task is to extract competitor names from web pages.

2.2 Mutual Information Based Ranking

Turney [14] used point wise mutual information to evaluate 80 questions from the Test of English as a foreign Language(TOEFL) and got a score of 74%, while Latent Semantic Analysis, another method of evaluating the similarity of words, only got a score of 64%. Etzioni [12] also uses PMI-IR methods to assess the probability of extraction in KonwItAll system. Zhu [15,16] calculates the relation strength between two entities based on co-occurrence and distances between two entities in a data set. Our work differs from Zhu's because we focus on competitive relationship. For instance "Microsoft" and "Windows" are not

competitors although they have high score given by Zhu's algorithm. The core idea here is to make use of information redundancy for validating correlation.

3 Competitor Discovery Using Mutual Information

In this section, we first propose a novel method for gathering informative pages from web. Then we introduce a set of linguistic patterns for extracting competitors, and discuss how to rank correlation strength between the targeted entity and its competitors based on mutual information. Finally the ambiguity problem is studied.

3.1 Gathering Informative Pages

As we defined in Section 1, we call those pages informative pages, which may contain information of competitors. These pages are our only resources from which competitors are extracted, so we need to find these pages accurately and effectively. Etzioni [17] introduced the metaphor of an Information Food Chain where search engines are herbivores "grazing" on the web and intelligent agents are information carnivores that consume output from various herbivores. In term of this metaphor, our system is an information carnivore that consumes the output of existing search engine. We send the queries by the combination of our pre-defined linguist patterns and input entity's name. The patterns which will be shown in the next section often used in pages which contain accurate competitive meanings. For example, searching for "Nokia", we send a set of queries such as "Nokia Vs" and "especially Nokia and", to search engine. With Google API, we collect top 100 pages for each query as our informative pages for extracting competitors. The results of our experiment discussed in Section 4 proves that these patterns are of high effectiveness for gathering informative pages.

3.2 Competitor Extraction

This step is to extract the competitor from those informative pages obtaining from the previous steps. Precise entity extraction requires sound linguistic rules. Due to the diversity of the Web and the lack of structure, this is unfortunately not a trivial task. From our experiment and previous research([12][11][13]), we identify a set of linguistic patterns to extract the competitors' names of original entity. Following parts, we will describe the patterns that we exploit and give a corresponded example from our informative pages(all examples use "Nokia" as input entity name).

Hearst Patterns. The first patterns have been used by Hearst to identify isa relationships between the concepts referred by two terms in text. However, the two terms are usually the competitors. Here we use CN as competitor names, EN as entity name.

The patterns reused from Hearst are:

H1: such as EN (,CN)* or || and CN

e.g., “phones such as Nokia, Motorola, Samsung, and Sony Ericsson”

H2: especially EN (,CN)* and CN

e.g., “Especially Nokia, Siemens, ...”

H3: including EN (,CN)* and (CN)

e.g., “cheap sim free mobile phones including nokia, motorola, samsung,”

Comparison Patterns. The next patterns are often used with two compared entities. We often compare two entities using “A or B” and “A vs.B” in document. So we apply the following two rules to extract competitors.

C1: CN vs EN

e.g. “IT-Director.com: Microsoft vs. Nokia”

C2: EN vs CN

e.g. “Nokia vs SonyEricsson at Forever Geek”

C3: EN or CN

e.g. “lets you to manage Nokia or Samsung mobile phone”

C4: CN or EN

e.g. “Who will win the mobile war - Microsoft or Nokia?”

3.3 Ranking Algorithms

After extracting a list of candidates using above patterns, we need to evaluate the candidates. The candidates’list may contain noisy data of original pages. We have explored two versions for filtering the noisy competitors in the candidates’list. An algorithms based on mutual information is employed for assessing competitors and ranking them according to their comparability.

Noisy Competitor Filtering. The simplest version for filtering noisy data just adds all the hits of extraction patterns resulting from one competitor.

$$counts(c, e) := \sum_{p \in P} count(c, e, p) \quad (1)$$

Where $counts(c,e)$ means the hits of all extraction patterns, c means the competitor and e means the original entity. Due to the redundance of the web, we set a threshold for assessing whether it’s a noisy competitor. In our experiment the threshold is set to 1. That’s to say we ignore all the names matched only once by pre-defined patterns. The second method linearly weight the contribution of each pattern for calculating $counts(c,e)$.

$$counts(c, e) := \sum_{p \in P} \omega_p count(c, e, p) \quad (2)$$

Where ω_p is the weight of pattern p . We give the pattern C1 and C2 higher weight of the contribution in our experiment since more competence meanings are expressed i them which showed in section 4. However, by our experiment, the method proves little benefit when compared with formula(1)

Mutual Information Based Ranking. After filtering the noisy competitor in the candidates'list, a rank list of relevant competitors is presented. We use Point wise Mutual Information(PMI) to measure the comparability between original entity and its competitors. In following formula, c_i represents competitor, e represents original entity.

$$Score(c_i|e) = hits(e, c_i)/hits(e) \quad (3)$$

Different competitors can be compared now via the score given by the formulation(3).

3.4 Entity Disambiguation

Ambiguity of a named entity is a traditional problem which prevents the performance of competitor discovery systems in two aspects, polysemy and synonymy.

Polysemy Problem. Polysemy means one entity name may have different meanings under different contexts. For example, the term Liverpool may refer to not only a football club, but also a city of England. So the output of our system may be a list containing other terms(i.e Birmingham, London, Manchester Unite). But when one is interested in the football field, he/she may not be interested in competitors as Birmingham and London. Generally speaking, search engines are not able to handle this problem directly, and neither are our system as long as no further information is offered about the entity. Traditionally cluster method is always applied in the IR for each document, but it does not work in our system since it does not discard the entity 'London', but just cluster "London" into the class. Our system is capable of solving the problem above partially by means of putting domain information additionally through the interface we provide for users. For example, We search for "Liverpool" and put domain information 'football' additionally, and then our system will generate queries through predefined patterns with the added "football". So the informative pages may all be related to "football" domain, and competitor discovered in these pages may also be related to "football". Adding a domain information has little influence on the extracting strategy.

To add domain information to our ranking algorithms, we modify equation (3) as follows , D represents for domain information.

$$Score(c_i|e, D) = hits(e, c_i|D)/hits(e|D) \quad (4)$$

Synonymy Problem. Synonymy means different entity names may refer to the same entity. Our observation reveals that the problem may happen in the following situations. As an illustration, we use "Nokia 7220" as input in situation 1 and 2, and "Stanford" in situation 3.

- The brand name and brand product name are all in the candidates list.
E.g., "Nokia", "Nokia 6610".

Here “Nokia” appears as a brand name, it have little meaning for discovery a competitor of “Nokia 7720”. In the filtering process, we check every name in the list whether is a brand name of another entity in the list, if there are more than one entity have this brand name, we discard pure brand name from our list, if just on candidate has this brand name, we add the counts of this brand name to the product name, and discard the brand name from the list.

- The brand product name and the product name are all in the candidate list. E.g.: Candidates: “Nokia 6610”, “6610”.

Here the “6610” and “Nokia 6610” refer to the same entity. For this case, we check every name in the list whether there is another entity’s name contains this name as product name. If there is one we add the counts of the entity with product name to the entity which name is presented as brand product name.

- The full name and its abbreviation are all in the list e.g.: “Carnegie Mellon University”, “CMU”, “Harvard University”

For this situation, we use Xu et al.’s method [18] to discovery the abbreviation and its full name in the list during the filtering process, and add the counts of abbreviation term to the full name entity.

4 Experiment Results

This section evaluates the proposed technique and our system. The Figure 1 shows the overall system architecture and experiment process. We first use the Google search engine to obtain the initial set of informative pages. The size of this set of documents is limited to the first hundred results per query returned by Google. For every entity we have 7 different pre-defined patterns, so the number of documents is about 700 per input entity. Then extract the given entities’ competitors from these pages, and rank them based on the mutual information.

In order to evaluate how effectively each pattern among what we defined in Section 3.2 discovers the informative pages with comparison to traditional web search (using entity name as a query), we manually label the top 10 returned pages for each of the 10 entities to check out whether the returned pages contain the competitor names. The results are showed in the following Table 1. By the experiment we can find that our approach increases the percentage of discovering the informative pages by 49%, especially with CP1 and CP2 patterns. Here CP represent the Comparison Patterns, and HP represent the Hearst Patterns defined in Section 3.2.

Table 2. demonstrates the output of the system, 32 sets of the competitors, each of which are made due to one of the varied 32 input entities. These 32 input entities are distributed in 5 different fields including football clubs, universities, companies, products and football stars. Here we take football clubs and universities as examples by which we illustrate our system because rankings of them pronounced by different social organizations and braches have already exist, and we can simply make use of these rankings to access precision of our system.

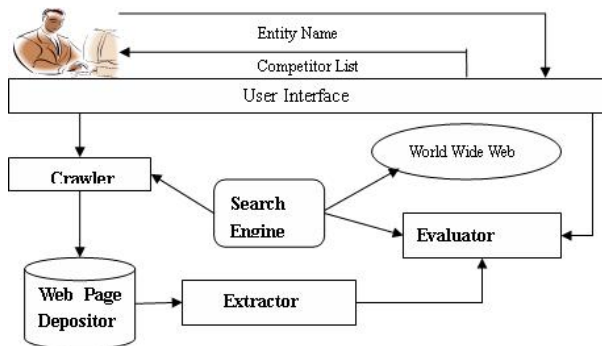


Fig. 1. Architecture of CoDis

Table 1. Statistics of Informative Pages

	Original	Average	HP1	HP2	HP3	CP1	CP2	CP3	CP4
Total Result	13	433	65	55	60	79	75	47	52
Percentage	13%	62%	65%	55%	60%	79%	75%	47%	52%

From Table 2, we can see that our system is capable of discovering each input entity’s competitors effectively and accurately. In each box ,the first line gives the entity name which is the input of our system. For each entity, we listed only top 5 (if there more than 5 entity name in the candidates’ list)ranking competitors.

From Table 2 We can find that, however, “Rice”, and “Liverpool” render our examples themselves ambiguous. The ambiguity here results from the most three commonly used meanings—the harvest of certain crop, person’s name, and the name of a American university—and the usually understandings of the word “Liverpool”—the city in England and a well-known football club. In order to solve this problem we can provide the domain information for each entity with the method proposed in Section 3.4 (i.e. equipping the input entity ‘Rice’ with the domain information “university” and providing the domain information “foot-ball” for the input entity “Liverpool”), and the results with this method is represented in Table 3. In Table 3, first line gives the entity name and second gives result without domain information , third line gives result with domain limitation. As our experiment showed, we are able to solve the problem of ambiguity by means of providing the input entity with additional restricts. Besides, experiments show that, not only the ambiguity is eliminated but also the competitors of the input entities can be restricted in the particular fields through providing domain information for the input entities. The example is also showed in Table 3.The method providing ‘notebook’ as the domain limitation for searching “IBM”’s competitors in notebook area generates the distinguishing results compared to the previous one.

Table 2. Result Of Competitor Discovery

AC Milan Barcelona Manchester Unite Juventus Madrid Inter Milan	Juventus Lazio Arsenal Porto Roma AC Milan	Arsenal Chelsea Liverpool Manchester United Aston Villa Real Madrid	Liverpool Birmingham London Edinburgh Real Bristol Manchester United
Newcastle Manchester Liverpool Leeds Glasgow Bolton	West Ham Aston Newcastle Tottenham Blackburn Fulham	Aston villa West Ham Arsenal Newcastle Bolton Leeds	Charlton Chelsea Arsenal Fulham Liverpool Aston villa
Dartmouth Stanford Georgetown Boston college Williams MIT	MIT Brown Yale Stanford Som Humboldt	Princeton Texas Cambridge Yale Stanford Harvard	Berkeley Brown Davis Yale Wisconsin UCLA
Stanford Harvard Chicago Michigan MIT Princeton	Rice West Brown Bush Houston Michigan	Yale Chicago Princeton Stanford Columbia Duke	Washington Uni. Harvard Stanford MIT Colorado College Greenville college
IBM Intel Sun Oracle HP Apple	Microsoft Linux IBM Oracle Sony Google	Google Yahoo Microsoft eBay MSN Skype	Nokia Motorola Sony Ericsson Siemens Samsung Microsoft
Nike Disney Adidas Toyota Guinness Starbucks	Adidas Reebok Salomon Sony Prada Nike	IBM T43 Dell i6000 Fujitsu Hpl2000 Sony vaio IBM t60	IBM x32 Dell Fujitsu Sony TR IBM x40 Asus s5200
Xbox Play Station Pc Gaming GC Gamecube	Canon A70 Canon a-80 Olympus mju 300 Sony dsc-p43 Nikon 5700 Olympus c720	Nokia 7270 Nokia 6260 Sonyericcson s700	Motorola v360 sagem myx6 Nokia 6020
Ronaldo Ronaldinho Beckham Rooney Francisco Raul	Thierry Henry Ronaldo Chelsea Zxinedine David Beckham Robert Pires	Zidane Beckham Henry Raul Owen Trezeguet	Lampard Drogba Henry John Terry Steven Gerrard Beckham

Table 3. Domain Information for Competitor Discovery

	Liverpool	Rice	IBM
Codis	Birmingham	West	Intel
	London	Brown	Sun
without	Edinburgh	Bush	Oracle
domain	Bristol	Houston	HP
D	Manchester Unite	Michigan	Apple
Coids	Manchester Unite	Texas	Toshiba
	Arsenal	East carolina	HP
with	everton	Stanford	Dell
domain	Newcastle	Rutgers	Apple
D	Chelsea	Cornell	Mac

5 Conclusion and Future Work

This paper has introduced a highly effective method for competitor finding and ranking. First, we discover the certain linguistic patterns from common sense or by our observation which are used for extracting targets entity's competitors. And our algorithm is enable us to evaluate the comparability of the competitors in the meanwhile reduce the web information noise and then propose a ranking of the competitors by means of mutual information based on "web-scale statistics". This paper also introduced some heuristic rules to solve the ambiguity and synonymy which are both the traditionally critical flaws in the web search.

As for future work, we would like to automatic find more competitor patterns through bootstrapping and carefully solve the entity name disambiguates.

References

1. Zanasi, A.: Information gathering and analysis competitive intelligence through data mining public sources. *Competitive Intelligence Review* (9(1)) 44 – 54
2. Morinaga, S., Yamanishi, K., Tateishi, K., Fukushima, T.: Mining product reputations on the web. *kdd-02,2002*. In: *KDD-02*. (2002)
3. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: *KDD-04*. (2004)
4. Califf, M., Mooney, R.: Graph drawing by force-directed placement. *Journal of Machine Learning Research* 4 (2003) 117–210
5. Freitag, D., McCallum, A.: Information extraction with hmms and shrinkage. In: *Processdings of the AAAI-99 Workshop on Machine Learning for Information Extraction*. (1999)
6. Ashish, N., Knoblock, C.: Wrapper generation for semi-structured internet sources. *SIGMOD Record* 26(4) (1996)
7. Cohen, W., Hurst, M., Jensen, L.: A flexible learning system for wrapping tables and lists in html documents. In: *Proceedings of World Wide Web (www02)*. (2002)
8. Hearst, M.: Automatic acquisition of hyponyms from large text corpora. In: *Proceedings of the 14th International Conference on Computational Linguistics*. (1992)

9. Charniak, E., Berland., M.: Finding parts in very large corpora. In: In proceeding of the 37th Annual Meeting of the ACL. (1999)
10. Hahn, U., Schnattinger, K.: Towards text knowledge engineering. In: Proceedings of the 15th Naional Conference on Artificial Intelligence and the 10th Conference on Innovative Application of Artificial Intelligence (AAAI'98/IAAI'98). (1998) 524–531
11. Cimino, P., Handschuh, S., Staab, S.: Towrds the self-annotating web. In: Processdings of World Wide Web (WWW-04). (2004)
12. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, S.: Web-scale information extraction in knowitall(preliminary results). In: Processdings of World Wide Web (WWW-04). (2004)
13. Liu, B., Chin, C.: Mining topic-specific concepts and definitions on the web. In: Processdings of World Wide Web (WWW-03). (2003)
14. Turney: Mining the web for synonyms:pmi-ir versus lsa on toefl. In: Proceedings of the Twelfth European Conference of MachineLearning. (2001)
15. Zhu, J., Goncalves, A.L., Uren, V.S., Motta, E., Parcheco, R.: Mining web data for competency management. In: Proc. of 2005 International Conference on Web Intelligence. (2005)
16. J.Zhu, G.: Corder:community relation discovery by named entity recognition. In: K-CAP. (2005)
17. Etzioni, O.: Moving up the information food chain: Softbots as information carnivores. In: Proceedings of the Thirteenth National Conference on Artificial Intelligence. (1996)
18. Xu, J., H., Y.: Using svm to extract acronyms in text. In: Proceedings of International Conference on Machine Learning and Cybernetics(ICMLC 2005). (2005)

Co-expression Gene Discovery from Microarray for Integrative Systems Biology

Yutao Ma and Yonghong Peng

Department of Computing, University of Bradford, West Yorkshire, UK, BD7 1DP
qwe1979@hotmail.com, y.h.peng@bradford.ac.uk

Abstract. Advance of high-throughput technologies, such as the microarray and mass spectrometry, has provided an effective approach for the development of systems biology, which aims at understanding the complex functions and properties of biological systems and processes. Revealing the functional correlated genes with co-expression pattern from microarray data allows us to infer the transcriptional regulatory networks and perform functional annotation of genes, and has become one vital step towards the implementation of integrative systems biology. Clustering is particularly useful and preliminary methodology for the discovery of co-expressed genes, for which many conventional clustering algorithms developed in the literature can be potentially useful. However, due to existing large amount of noise and a variety of uncertainties in the microarray data, it is vital important to develop techniques which are robust to noise and effective to incorporate user-specified objectives and preference. For this particular purpose, this paper presented a Genetic Algorithm (GA) based hybrid method for the co-expression gene discovery, which intends to extract the gene groups that have maximal dissimilarity between groups and maximal similarity within a group. The experimental results show that the proposed algorithm is able to extract more meaningful, sensible and significant co-expression gene groups than the traditional clustering methods such as the K -means algorithm. Besides presenting the proposed hybrid GA-based clustering algorithm for co-expression gene discovery, this paper introduces a new framework of integrative systems biology employed in our current research.

1 Introduction

Understanding the diverse biological functions of large number of genes on a genome-wide scale is a key challenge to the development of modern systems biology, and is important for many applications in the biomedical science, for example the understanding of complex diseases such as cancers. The most advanced technology for gene function identification is based on the interaction between genes under diverse biological conditions, and the role of genes in the particular biological systems and process [1,2,3]. Fig.1 shows the integrative systems biology framework employed in our functional genomics research, which aims at revealing the complex properties and functions of biological systems and processes, based on the integration of diverse biological data resources such as the microarray data, protein interaction

data, and protein-DNA interaction data and so on. This framework highlights that one important step towards the integrative systems biology is the discovery of functional correlated genes, which are co-expressed and share similar expression patterns under varied conditions. Two genes having sufficient similarity in terms of expression patterns are considered as functional related. The discovery of significant co-expression genes provides effective evidence for the construction of system-level genetic interaction map. The conventional functional genomics approach attempts to identify the functional related genes by means of calculating exhaustively all the possible gene pairs and searching for those gene pairs having sufficient similarity (larger than a certain threshold). Given a large number of genes involved in annotation of genome-wide function, the computational load is huge. In order to tackle such problem, in this study, we intend to develop a generic clustering approach that recursively searches for the functional related genes interactively.

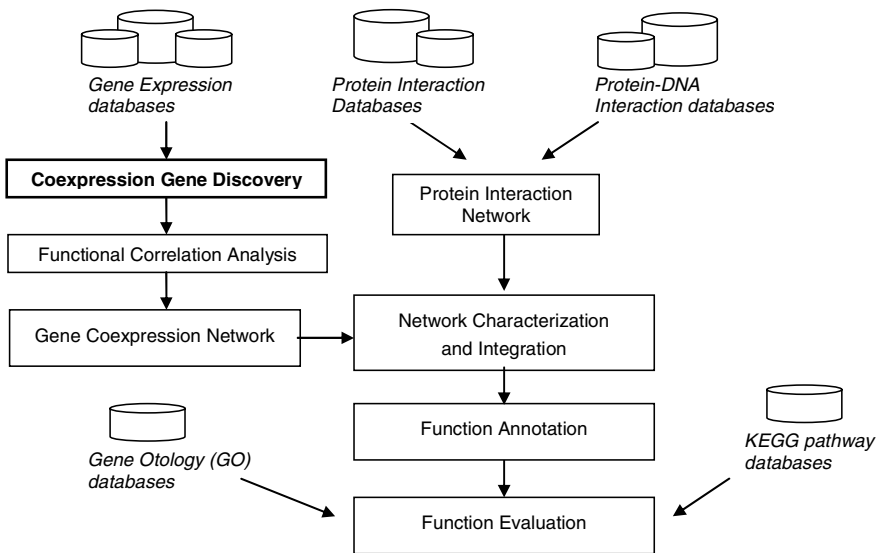


Fig. 1. Co-expression Gene Discovery in Integrative Systems Biology

Clustering is a procedure to partition given data items into groups in terms of a set of specified features. Give a data set of $X = \{x_i, i = 1, \dots, n\}$ each data item x_i is assigned to one cluster (denoted as C_k), so that if $d(x_i, x_j) < d(x_i, x_l)$ x_i and x_j are assigned in the same cluster otherwise x_i and x_l are associated to different clusters. The $d(x_i, x_j)$ is the dissimilarity between x_i and x_j . As a result of clustering, the genes within the same group are intended to share certain kind of functional relationship. There are two kinds of advantages of using clustering based functional correlation discovery over the conventional approaches. One is that while not missing the meaningful co-expression between genes, appropriate partitioning gene set would greatly reduce the computational load as needed in calculating

exhaustively the correlation of gene-pairs. Secondly, the clustered genes would enable the gene co-expression network to reflect well the scale-free characteristics of many nature systems [4,5].

During the past years, many clustering algorithms, such as K -means, hierarchical, self-organizing maps (SOM), expectation maximization (EM), have been developed and applied to the gene expression data analysis [6, 7]. In this paper, a hybrid Genetic Algorithm (GA) based clustering method is introduced for revealing the gene co-expression patterns. GA is one of the evolutionary computing techniques, which have been used in many real-world problems such as image clustering analysis [8] and stock data analysis [9]. The main motivation of employing GA in our study is due to the need of incorporating user-specified preference in defining the co-expressed genes. For example, the user may define the objective function as the similarity in terms of the changing patterns or the distance between expression patterns. In the conventional clustering methods such as K -means and hierarchical clustering, it is inconvenient to incorporate high-level searching preference. In fact, the k -means algorithm was designed to optimize a single objective function, $\sum_{j=1}^K \sum_{x_i \in C_j} (x_i - u_j)^2$, where the C_j represents the j -th cluster, x_i is the data point to be clustered, and u_j is the centroid of associated cluster.

2 Co-expression Gene Profiles in Microarray

A microarray dataset is represented by a $m \times n$ matrix $M=[e_{ij}]_{m \times n}$, where e_{ij} denotes the expression level of the i -th gene in the j -th sample, the m and n are respectively the numbers of genes and the number of samples (biological conditions) [10]. One row of matrix M , $g_i=(e_{i1}, e_{i2}, \dots, e_{in})$, represents the expression profile for the associated gene (g_i) over various biological conditions (such as healthy and cancerous) or over the development of biological process (time-course microarray).

The expression profile, $g_i=(e_{i1}, e_{i2}, \dots, e_{in})$, is called the expression pattern of gene (g_i), which provides important biological information regarding the behavior of a gene under various biological conditions. For co-expression gene discovery, the conventional method is to estimate the similarity between the expression profiles of each gene-pair [1,2,3]. Due to the large number of genes involved, the load of calculating the similarity for all the gene-pairs is a huge and even infeasible when it is necessary to integrate a large number of microarray datasets obtained in diverse experimental studies. The clustering analysis provides an alternative approach with high efficiency to address this problem. By means of clustering the genes in terms of the expression profiles, we can thus identify functional relationship between genes.

Given no standard methodology to measure the functional similarity between two genes, biological user may have diverse preference in defending the functional similarity. Incorporating the user-specified objective and preference is thus essential in practice. For this purpose, we designed a hybrid GA-based gene clustering approach, which attempts to distribute the gene having sufficient user-specified functional similarity into the same group and the data points having sufficient difference into different groups. More particularly, given a microarray dataset represented by a $m \times n$ matrix $M=[e_{ij}]_{m \times n}$ it attempts to distribute the m genes into a set

of clusters denoted by C_1, C_2, \dots, C_K such that (i) $C_i \neq \Phi$ for $i=1, \dots, K$; (ii) $C_i \cap C_j = \Phi$ for $i=1, \dots, K, j=1, \dots, K$ and $i \neq j$; (iii) $\|\cup C_i\| = m$; (iv) within each C_i the gene expression profiles have sufficient similarity while the gene expression profiles from different C_i and C_j have maximal dissimilarity.

3 Hybrid GA-Based Algorithm for Co-expression Gene Discovery

A GA is an evolutionary search for an optimal solution based on a fitness function. Fig.2 outlines the hybrid GA-based clustering algorithm. Different from conventional GA, the proposed method does not intend to search for an optimal solution, but aims to produce a set of cluster with good quality, i.e. to distribute genes, adaptively, into a set of appropriate and good quality clusters. To do this, the proposed approach employs an external evaluation mechanism, which assesses the quality of clusters, and guides the construction of clusters, as shown in the right-hand side of Fig.2.

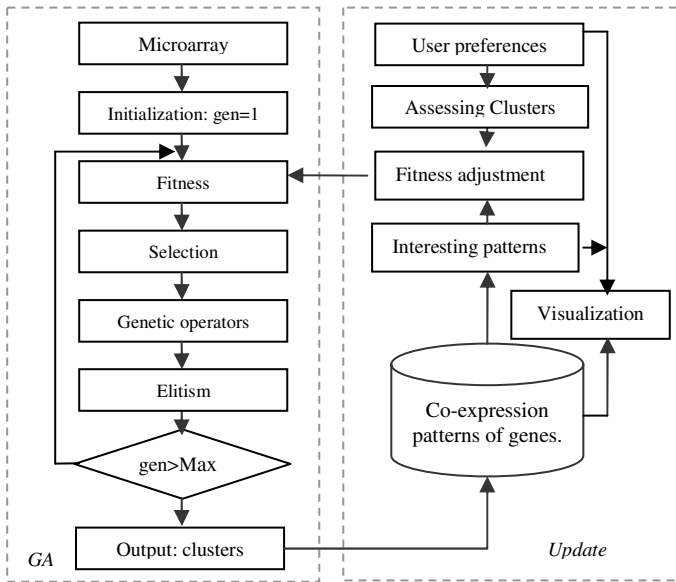


Fig. 2. Hybrid GA-based co-expression gene discovery

3.1 GA-Based Clustering

The left-hand side of Fig.2 is the conventional genetic algorithm, which evolutionarily search for the genes group by means of genetic operations (selection, crossover, mutation and so on). During the evolutionary searching, the GA generates a set of chromosomes in each searching step. A chromosome contains a totally of N elements represented by $Ch_l(i) = (G_{l1}(i), G_{l2}(i) \dots G_{lN}(i))$, where N is the pre-set number of clusters (the maximal number of clusters), in which $G_{lk}(i)$ denotes the centre of the i -th cluster

of the k -th chromosome at the l -th generation. The cluster centre $G_{lk}(i)$ is represented by a vector $G_{lk}(i) = (e_{lk1}(i), e_{lk2}(i), \dots, e_{lkn}(i))$ corresponding to a microarray data matrix $M=[e_{ij}]_{m \times n}$. The number of clusters (K) could be different from the pre-set number N , and the system can adaptively adjust the number of clusters, which provides more flexibility for the users. As detailed in section 3.2, when matching with a certain criterion, the gene cluster is set to be $G_{lk}(i) = (\infty, \infty, \dots, \infty)$ as an invalidated cluster.

When the cluster centres have been generated, the memberships of a gene $g_i=(e_{i1}, e_{i2}, \dots, e_{in})$ from microarray $M=[e_{ij}]_{m \times n}$ to them are determined by $\mu_l(i, j) = \begin{cases} 1 & \text{if } s_l(i, j) \geq s_{k \neq l}(i, j) \\ 0 & \text{otherwise} \end{cases}$, where $\mu_l(i, j)$ denotes membership of gene $g_i=(e_{i1}, e_{i2}, \dots, e_{in})$ to the cluster- l , and $s_l(i, j)$ is the similarity between gene g_i to the centre of cluster- l .

In the implementation of GA, tournament selection and two points crossover is used to generate offspring, and when the $G_{lk}(i) = (e_{lk1}(i), e_{lk2}(i), \dots, e_{lkn}(i))$ is selected for mutation, the following operation will be performed for its each element: $e_{(l+1)ki} = e_{lki} \times (1 \pm \delta)$, where δ is a random number between 0 and 1. The probabilities for crossover and mutation are set to be 0.8 and 0.01 respectively in our implementation. At each step of evolutionary operation, the best chromosome (or few best chromosomes) adopted from the previous population are copied to the new population, and the genetic algorithm evaluate the performance of each elements of chromosome of the new generation.

3.2 External Evaluation of Clusters

The right-hand side of Fig.2 is a component for the evaluation of cluster quality, which can be seen as the interface between user and computer that incorporates the user-specified objectives and preference into the clustering process. The main function of this component is to (1) assess the quality of clusters, (2) provide interactive visualization, and (3) adaptive guide the search process and adjust the number of cluster if necessary. The quality of clusters produced is assessed in each generation, so as to guide GA to generate improved clusters in successive generations.

The quality of clustering is evaluated in terms of two aspects in this study, which are the quality of each clusters, and the discrimination between the clusters, as detailed below.

1) The quality of each cluster is assessed by $cq = \frac{\sum \bar{s}_m}{n}$, where

$$\bar{s}_m = \sqrt{\frac{\sum_{i=1}^M (\bar{g}_i - \bar{g}_m)^2}{M}} \quad \bar{s}_m = \sqrt{\frac{\sum_{i=1}^M (\bar{g}_i - \bar{g}_m)^2}{M}}$$

is the standard deviation vector, and n is the number of elements of the vector (i.e. n is the number of

microarray data samples as in $M=[e_{ij}]_{m \times n}$, and $\bar{g}_m = \frac{\sum_{i=1}^M \bar{g}_i}{M}$ is called the mean-pattern of a cluster, where M is the number of genes in the associated cluster.

- 2) The discrimination between different clusters is measured based on the Pearson's correlation coefficients (cc) between the mean-patterns of clusters

$\bar{g}_{m_i}, i=1 \sim K$. We used the $ds = \frac{2}{1+|cc|} - 1$ to assess the dissimilarity between

clusters.

For adjusting the number of clusters, a simple strategy is employed in this study, which adaptively drops a cluster by setting $G_{jk}(i) = (\infty, \infty, \dots, \infty)$ when the associated cluster has been confirmed to be a poor cluster, in terms of cq .

4 Experimental Results

4.1 Experiment Dataset and Clustering Results

Leukemia dataset [10] is used in this paper for the purpose of illustrating the performance of the presented approach and system. In Leukemia data there are 7129 genes, and for 38 samples, i.e. 27 ALL (acute myeloid leukemia) and 11 AML (acute lymphoblastic leukemia). The objective of the original research was to identify the most informative genes for the purpose of disease modeling and more accurate classification of ALL/AML patients.

Fig.3 (b) and (c) show respectively the results of the hybrid GA clustering method and the K-means method. Visual analysis and manual inspection of the clustered genes found that our method produced more meaningful clusters than the K-means, as summarized in the following:

- 1) The K-means distributed many genes having sufficient similarity into separate groups. For example, in Fig.3(c), the genes of (a) groups 7 and 9 and (b) groups 1 and 10 have actually been found to be very similar in terms of their expression patterns. In contrast, the hybrid GA clustering method is able to gather these genes into appropriate clusters, i.e. it combines the groups (9 and 7) and groups (1 and 10) produced by K-means respectively into group 9 and group 7, as shown in Fig.3 (b).
- 2) The hybrid GA clustering method extracts more compact clusters than the k-means method. This can be illustrated by the extracted differential expression gene groups, i.e. groups 2, 3, 8, 10 and groups 2, 3,4, 5 respectively in Fig.3(b) and (c). Distributing the interesting genes into appropriate compact groups is significant and useful for biological applications, which helps to interpret the functions and relationship among them. For example, the identification of small number of genes that share differential expression pattern is able to help us to focus on investigating their effect to a particular cancer.



Fig. 3. Clustering Analysis of Leukemia dataset

4.2 Statistical Evaluation of Clusters

The statistical evaluation of clustering is still an open issue although there are several algorithms have been proposed in the literature [12,13]. In this paper, the Davies-Bouldin (DB) index [12, 13] is used to characterize the overall quality of clusters, and the gene correlation to measure the quality of gene groups.

DB index: DB index is the ratio of the sum of within-cluster scatter to between-cluster separation, $DB = \frac{1}{K} \sum_{i=1}^K \max_{j, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{|z_i - z_j|} \right\}$, where $S_{i,q} = \left(\frac{1}{|C_i|} \sum_{x \in C_i} \|x - z_i\|_2^q \right)^{1/q}$

is the scatter within cluster C_i , where z_i is the centroid of cluster C_i , and the $|C_i|$ denotes the number of members (genes) in cluster C_i . In this study, $q=2$ is used.

The *DB* values of the associated clustering results of the hybrid GA approach and the *K*-means are shown in Table 1. A small *DB* value indicates that the average distance between the elements within one cluster is small and distance between clusters is big, and thus implies the quality of clusters is good. It has been seen that the *DB* value of clusters of hybrid GA method is constantly smaller than that of *K*-means, which indicates the hybrid GA method produced better clustering results than *K*-means.

Table 1. DB values of clusters produced by Hybrid GA and K-means

No	Clusters No.	K-means	Hybrid GA
1	5	3.4846	3.2036
2	5	3.4817	3.3456
3	10	4.9019	4.6093
4	10	4.9292	4.2949
5	15	5.1782	5.0422
6	15	5.1182	5.0667

Gene Correlation: To investigate the characteristics of each cluster in further detail, we calculated the average Pearson’s correlation coefficient for all the gene-pairs within each cluster. The results are shown in Table 2. It has been seen that two clusters (groups 3 and 7) produced by the *K*-means algorithms are associated with very small average correlation coefficients (i.e. 0.046 and 0.021 respectively), which indicated that these two clusters have very poor quality. It has been seen that all the clusters produced by the proposed GA based clustering algorithm have much better property than these two clusters.

Table 2. Correlation of genes within clusters

Clusters	Hybrid GA		K-means	
1	0.114		0.243	
2	0.419	Up-regulated	0.380	Up-regulated
3	0.243	Down-regulated	0.046	Down-regulated
4	0.280		0.233	Up-regulated
5	0.119		0.414	Down-regulated
6	0.164		0.169	
7	0.163		0.021	
8	0.304	Down-regulated	0.314	
9	0.139		0.225	
10	0.220	Up-regulated	0.234	

Let us pay more attention to the gene groups of differential expression, i.e. the groups 2,3,8,10 of hybrid GA and the groups 2,3,4,5 produced by the K-means clustering. As shown in Fig.3, the groups (2 and10) and groups (3 and 8) obtained by the hybrid GA method and the groups (2 and 4) and groups (3 and 5) of k-means are respectively up-regulated and down-regulated differential expression gene clusters, as indicated in Table 2. The average correlation coefficients of up-regulated and down-regulated expression genes of hybrid GA are respectively the 0.3195 (i.e. $(0.419+0.22)/2=0.3195$) and 0.2735 (i.e. $(0.243+0.304)/2=0.2735$), while the coefficients of gene groups produced by the k-means clustering are respectively 0.3064 and 0.230. These results indicated clearly the hybrid GA generated interesting clusters with high correlation coefficients, thus outperforms the K-means method.

Computational efficiency: One important issue of GA-based clustering method is the efficiency of converging on a desired solution. In this study, considerable experiments have been performed to test the efficiency of algorithm. It has been seen that the proposed hybrid genetic algorithm is able to efficiently produce the desired clusters. One example is shown in Fig. 4, which indicates that the fitness value reached a stable and desired level within about 20 generations.

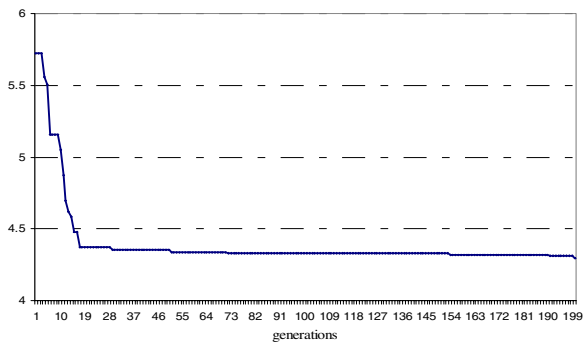


Fig. 4. Efficiency of the hybrid GA approach

5 Conclusion and Future Work

The co-expression gene discovery from diverse microarray data is one important step towards integrative systems biology. This paper presents a hybrid genetic algorithm based generic clustering approach for the co-expression gene discovery. The developed approach is able to incorporate the user-specified objectives and preference in evolutionary construction of appropriate clusters by means of an external cluster evaluation component.

Experimental results of Leukemia dataset demonstrate the performance of the proposed approach. It has clearly seen that the clusters generated by the proposed approach have better biological interpretation in visual observation and manual inspection. The statistical numerical analysis, the *DB* index and the Pearson's correlation coefficient, has further confirmed that the proposed clustering method outperforms the *K*-means method. It has been seen that the clusters produced by the

proposed method not only have better overall quality but also have better quality for each single cluster. As a component of the integrative systems biology platform, a particular advance in the future is to extract multiple-structured clusters or network-linked clusters in order to capture better functional relationship of genes.

References

1. A. Tanay, I. Steinfeld, M. Kupiec and R. Shamir, Integrative analysis of genome-wide experiments in the context of a large high-throughput data compendium, *Molecular Systems Biology*, Published online: 29 March 2005.
2. S. Imbeaud and C. Auffray, Functional Annotation: Extracting functional and regulatory order from microarrays, *Molecular Systems Biology*, Published online: 25 May 2005.
3. X.J. Zhou, M.C. Kao, et al., Functional annotation and network reconstruction through cross-platform integration of microarray data. *Nat Biotechnol* Vol. 23, 238–243, 2005.
4. I. K. Jordan *et al.*, Conservation and co-evolution in the scale-free human gene coexpression network, *Mol. Biol. Evol.* vol. 21, 2058–2070, 2004.
5. M. G. Grigorov, Global properties of biological networks. *Drug Discov Today* vol.10, 365–372, 2005.
6. A. F. Famili, G. Liu, and Z. Liu, Evaluation and Optimization of Clustering in Gene Expression Data Analysis, *Bioinformatics*, vol. 20, 1535-1545, 2004.
7. D. Jiang, C. Tang, and A. Zhang, Cluster Analysis for Gene Expression Data: A Survey, *IEEE Trans. Knowledge and Data Engineering*, vol. 16, 1370-1386, 2004.
8. S. Bandyopadhyay and U. Maulik, Genetic Clustering for Automatic Evolution of Clusters and Application to Image Classification, *Pattern Recognition*, vol.35, 1197-1208, 2002.
9. R. J. Povinelli and X. Feng, A New Temporal Pattern Identification Method for Characterization and Prediction of Complex Time Series Events, *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, 339-352, 2003.
10. T.R. Golub, and D.K. Slonim, Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, *Science*, vol.286, 531-537, 1999.
11. U. Maulik and S. Bandyopadhyay, Genetic Algorithm Based Clustering Technique, *Pattern Recognition*, vol.33, 1455-1465, 2000.
12. U. Maulik, Performance Evaluation of Some Clustering Algorithms and Validity Indices, *IEEE trans. Pattern Analysis and Machine Intelligence*, vol. 24, 1650-1654, 2002.
13. A. Fazel. Famili, Ganming. Liu, and Ziying. Liu, Evaluation and Optimization of Clustering in Gene Expression Data Analysis, *Bioinformatics*, vol. 20, 1535-1545, 2004.
14. A. V. Lukashin, and R. Fuchs, Analysis of Temporal Gene Expression Profiles: Clustering by Simulated Annealing and Determining The Optimal Number of Clusters. *Bioinformatics*, vol. 17, 405-414, 2001.
15. C. Arima, and T. Hanai, Gene Expression Analysis Using Fuzzy K-Means Clustering, *Genome Informatics*, vol. 14, 334-335, 2003.
16. M.B. Eisen, P.T. Spellman, P.O. Brown, and David Boststein, Cluster Analysis and Display of Genome-wide Expression Patterns, *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, 14863-14868, 1998.

Cardiovascular Disease Diagnosis Method by Emerging Patterns*

Heon Gyu Lee¹, Kiyong Noh², Bum Ju Lee¹, Ho-Sun Shon¹, and Keun Ho Ryu^{1,**}

¹Database/Bioinformatics Laboratory, Chungbuk National University, Cheongju, Korea
{hglee, bjlee, shon0621, khryu}@dblab.chungbuk.ac.kr

²Korea Research Institutes of Standards and Science, Korea
kyno@kriss.re.kr

Abstract. Currently, many researches have been pursued for cardiovascular disease diagnosis using ECG so far. In this paper we extract multi-parametric features by HRV analysis from ECG, data preprocessing and heart disease pattern classification method. This study analyzes the clinical information as well as the time and the frequency domains of HRV, and then discovers cardiovascular disease patterns of patient groups. In each group, its patterns are a large frequency in one class, patients with coronary artery disease but are never found in the control or normal group. These patterns are called emerging patterns. We also use efficient algorithms to derive the patterns using the cohesion measure. Our studies show that the discovered patterns from 670 participants are used to classify new instances with higher accuracy than other reported methods

1 Introduction

The most widely used signal in clinical practice is the electrocardiogram (ECG). It is frequently recorded, and widely used for the assessment of cardiac function [1],[2]. ECG processing techniques have been proposed to affect pattern recognition, parameter extraction, spectro-temporal techniques for the assessment of the heart's status, denoising, baseline correction and arrhythmia detection [3],[4],[5]. It has been reported that Heart Rate Variability (HRV) is related to autonomic nerve activity and is used as a clinical tool to diagnose cardiac autonomic function in both health and disease [6]. This paper provides classifying technique that could automatically diagnose Coronary Artery Disease (CAD) in the framework of an ECG pattern and clinical investigations. Hence, through ECG, we are able to present features that could well reflect the existence and non-existence of a CAD. Above features can be perceived through HRV analysis and they are based on following knowledge [7],[8].

- In patients with CAD, reduction of the cardiac vagal activity evaluated by spectral HRV analysis was found to correlate with the angiographic severity.
- The reduction of variance (standard deviation of all normal RR intervals) and low-frequency of HRV seem related to an increase in chronic heart failure.

* This works was supported by the Regional Research Centers Program of Ministry of Education & Human Resources Development, and Korea Science and Engineering Foundation (#1999-2-303-006-3).

** Corresponding author.

- The low-frequency of HRV is decreased or absent in chronic failure patients with advanced disease and is related to the progression of the heart failure.

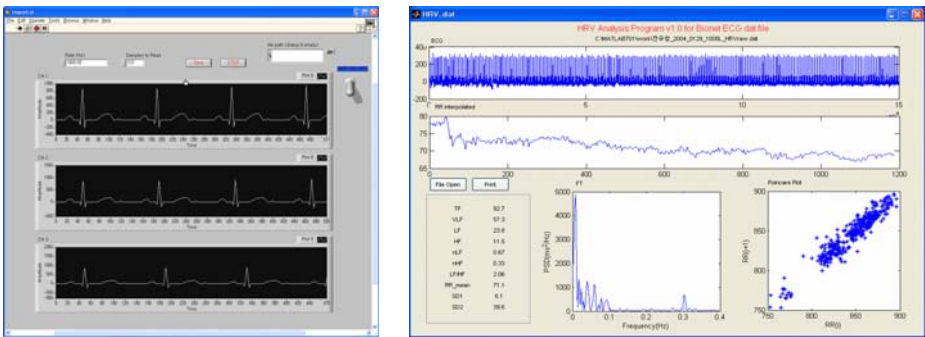
Our aim of this paper is to not only build the classifier for cardiovascular disease but also verify the effectiveness and accuracy of classifier. To achieve the purpose, we use emerging patterns (EPs) [9]. EPs are defined as itemsets whose supports change significantly from one data class to another. EPs are also easy to understand because they are just collections of attributes in dataset and this property is especially important for biomedical applications. However, the volume of EPs generated is very large for high dimensional medical data. In order to maintain the efficiency in using EPs, our classification method uses top EPs instead of full set of EPs and can be extended by using EP cohesion measure for uninformative patterns pruning in our application. The proposed emerging pattern method can also relax independence assumption of classifiers like NB (Naive Bayesian) [10] and DT (Decision Tree) [11]. For example, the NB makes the assumption of conditional independence, that is, given the class label of a sample, the values of the attributes are conditionally independent of one another. When the assumption holds true, then the NB is the most accurate in comparison with all other classifiers. In practice, however, dependences can exist between variables of the real data. Our classifier can consider dependences of linear characteristics of HRV and clinical information. In this work, we have compared several different classification methods and have validated their accuracy for diagnosing of heart disease. Experiments also show that with proper classification methods, the results of diagnosis can be improved.

2 Feature Extraction and Data Preprocessing

The ECG signals are recorded by electrocardiography, and are transmitted immediately to a personal computer for recording for 5 minutes. The sampling frequency for electrocardiographic signals is 500Hz. The recorded electrographic signals are retrieved afterward to measure the consecutive RR intervals by using software for the detection of the R wave. The last 512 stationary RR intervals are obtained in each recumbent position for HRV analysis. The power spectra of 512 RR intervals are obtained by means of fast Fourier transformation. The direct current component is excluded in the calculation of power spectrum to remove the non-harmonic components in the very low-frequency region ($<0.04\text{Hz}$). The area of spectral peaks within the whole range of 0 to 0.4Hz was defined as *Total Power (TP)*, the area of spectral peaks within the range of 0 to 0.15Hz as *Low Frequency power (LF)*, and the area of spectral peaks within the range of 0.15 to 0.4Hz as *High Frequency power (HF)*, respectively. The *Normalized Low Frequency power* ($nLF = 100 \cdot LF/TP$) is used as an index of sympathetic modulation; the *Normalized High Frequency power* ($nHF = 100 \cdot HF/TP$) as an index of vagal modulation; and the *Low/High Frequency power ratio (LF/HF)* as the index of sympathovagal balance. Table 1 shows the results of extraction of HRV features from ECG signal, and the example of ECG monitoring program and feature extraction process from raw ECG signal is shown in Fig. 1.

Table 1. Features of HRV analysis

Feature	Description
TP	The variance of normal RR intervals in HRV over the 5 min ($\leq 0.4\text{Hz}$).
VLF	Power in very low frequency range ($\leq 0.4\text{Hz}$).
LF	Power in low frequency range (0.04~0.15Hz)
HF	Power in high frequency range (0.15~0.4Hz)
LF/HF	Ratio LF/HF
nLF	Normalized low frequency power ($=100 \cdot \text{LF}/(\text{TP}-\text{VLF})$)
nHF	Normalized high frequency power ($=100 \cdot \text{HF}/(\text{TP}-\text{VLF})$)
SDNN	Standard deviation of all RR intervals

**Fig. 1.** Development of HRV analysis program

Since the extracted features and clinical information datasets contain continuous variables, those variables also must be made discrete. Therefore, decision tree has been used because the intervals are selected according to the information they contribute target variable. Due to the decision tree's discretization [12], all continuous contributed variables are cut up into a number of intervals. Table 2 shows the selected features in raw training data and the example of discretization for the continuous variables presents in Fig. 2. Normalization is the process of converting values associated with nominal data items so that they correspond to unique integer labels.

Table 2. The result of feature extraction

Data	Extracted/Selected Features
HRV	TP, VLF, LF, HF, nLF, nHF, LF/HF, SDNN
Clinical Info.	Age, Hyper Blood Pressure, Diabetes Mellitus, Smoking, Old Myocardial Infarction, Ejection Fraction, Blood Glucose, Total Cholesterol, Triglyceride, Hyperlipidemia, Systolic Blood Pressure, Diastolic Blood Pressure



Fig. 2. Data preprocessing of the training dataset

3 Emerging Patterns Discovery

This section describes the building classifier phase which consists of discovering the emerging patterns. A kind of knowledge pattern, called Emerging Patterns (EPs), is introduced in [8]. Emerging patterns are defined as those itemsets whose support increases significantly from one dataset to another. Suppose a data object $obj = \{a_1, \dots, a_n\}$ follows the schema (A_1, \dots, A_n) , where A_1, \dots, A_n are called attributes. Let $C = \{c_1, \dots, c_m\}$ be a finite set of class labels. A training dataset is a set of data objects such that, for each object obj , there exists a class label $c_{obj} \in C$ associated with it. A classifier is a function from $\{A_1, \dots, A_n\}$ to C , which assigns a class label to an unseen example.

In general, given a training data, the task of classification is to build a classifier from the training dataset such that it can be used to predict class labels of unknown objects. Emerging patterns can serve as a classification model because they represent knowledge which discriminates between different classes of datasets.

Let I denote the set of all items in the encoding dataset D . A set X of items is also called an itemset, which is defined as a subset of I . We say any instance S contains an itemset, A in a dataset D , $sup_D(X)$, is $count_D(X)/|D|$, where $count_D(X)/|D|$ is the number of instances in D containing X .

Definition 1. Given two different classes of datasets D_1 and D_2 , the growth rate of an itemset X from D_1 to D_2 is defined as

$$Growth\ Rate(X) = Gr(X) = \begin{cases} 0 & \text{if } sup_1(X) = 0 \text{ and } sup_2(X) = 0 \\ \infty & \text{if } sup_1(X) = 0 \text{ and } sup_2(X) > 0 \\ \frac{sup_2}{sup_1} & \text{otherwise} \end{cases} \quad (1)$$

Emerging patterns are the itemsets with large growth rate from D_1 to D_2 .

Definition 2. Given $\rho > 1$ as a growth rate threshold, an itemset X is called a ρ -emerging pattern (ρEP or simply EP) from D_1 to D_2 if $Gr(X) \geq \rho$.

When D_1 is clear from context, an EP X from D_1 to D_2 is simply called an EP of D_2 . The support of X in D_2 , $sup_{D_2}(X)$, denoted as $sup(X)$, is called the support of the EP .

Emerging patterns can be described by borders [9]. A collection of sets represented by the border $\langle L, R \rangle$ is $[L, R] = \{Y \mid \exists X \in L, \exists Z \in R, X \subset Y \subset Z\}$. For instance the border $\langle \{\{1\}, \{2\}\}, \{1, 2, 3, 4\} \rangle$ represents those sets which are either supersets of $\{1\}$ and subsets of $\{1, 2, 3, 4\}$ or supersets of $\{2\}$ and subsets of $\{1, 2, 3, 4\}$. Clearly, borders are usually much smaller than the collections they represent. The collection of emerging patterns discovered from different classes of data, can be concisely represented by their border $\langle L, R \rangle$, where L is the sets of the minimal itemsets and R is the sets of the maximal itemsets. After describing emerging patterns, we select EP s satisfying the following conditions are the most expressive patterns for classification.

- Their growth rates are large. Very large or even infinite growth rates ensure EP 's significant level of discrimination
- They have enough supports in the target class. The support makes an EP cover at least a certain number of examples in the target class of training dataset. Itemsets with too low supports are regarded as noise.
- They are contained in the left bound of the border representing the EP collection. EP s in the left bound of the border have no subsets within the collection of EP s. That is, any proper subset of such an EP is not an EP any more. A shorter EP means less attributes. If we can use less attributes to distinguish two data classes, adding more attributes will not contribute to classification.

After all expressive EP s are discovered, the resulting set of EP s can still be huge and contain many uninformative patterns. To make the classification effective and efficient, we need to prune unnecessary EP s to delete noisy information. EP cohesion measure (EC) [13] is used for EP s ranking. Emerging pattern ranking is needed to select the best EP s in case of overlapping EP s. The proposed cohesion measure is adapted from the cohesion measure as defined below.

Definition 3. For a EPs $(item_1, \dots, item_n)$ of length n , EC is a ranking defined as

$$EC(item_1, \dots, item_n) = \frac{Count(item_1, \dots, item_n)}{\sqrt[n]{Count(item_1) \cdot \dots \cdot Count(item_n)}} \tag{2}$$

where $Count(item_1, \dots, item_n)$ is a number of instances where the itemsets occur together, $Count(item_i)$, $i=1, \dots, n$, is a number of instances containing $item_i$. Intuitively, EC is high when individual components of an item occur together and infrequently separately. All emerging patterns are ranked according to the following criteria.

Definition 4. Given two EPs, e_i and e_j , e_i is said to have higher rank than e_j (also called e_i precedes e_j), denoted as $e_i \succ e_j$ if

- ① $EC(e_i) > EC(e_j)$, or
- ② $EC(e_i) = EC(e_j)$, but $sup(e_i) > sup(e_j)$, or
- ③ $EC(e_i) = EC(e_j)$, $sup(e_i) = sup(e_j)$, but $length(e_i) > length(e_j)$.

Note the above order does not consider the growth rates of EPs. It is because EPs are defined as EPs with very large growth rates.

4 Classification Using Emerging Patterns

After a set of EPs is selected for classification, as discussed in previous section, EP-based classifier is ready to classify new objects. In this section, we test the usefulness of EPs in classification by conducting a 10-fold cross-validation application. We use 670 datasets from our ECG database which consists of patients (CAD) group and normal people. EP-based classification algorithm is presented in Fig. 3.

Algorithm 1: Classification method using EPs

Input: a set of EPs E_i for class C_i , the testing dataset $testD$. (i denotes *normal* or *CAD* class label)

Output: the classification a test object o of $testD$.

1. Sort E_i by rank in descending order;
2. While both $testD$ and E_i are not empty.
 - For each EP e_i in the rank descending order.
 - Find all data object $o \in testD$ containing e_i .
 - Object o is classified as class C_i .

Fig. 3. EP-based classification Algorithm

EP-based classification approach shows potential to discover differences from the ECG and clinical information of two groups (CAD, Normal). A total of 155 and 99 EPs in the CAD and normal people group were discovered by our algorithm. Table 3 shows the example of the top 10 EPs which occur in the 380 patients with CAD dataset, and the top 10 EPs which occur in the 280 normal dataset.

Table 3. Discovered EPs in two class groups

EPs	Support in CAD group	EPs	Support in normal group
{5,39,43,47,51}	0.893	{6,11,56,61,82}	0.75
{8,39,43,47,51}	0.893	{6,51,56,61,82}	0.75
{39,43,47,51}	0.891	{11,51,56,61,82}	0.632
{39,43,47,64}	0.862	{11,56,61,82}	0.632
{11,39,43,47}	0.862	{9,46,51,56,86}	0.605
{10,39,43,47}	0.835	{9,46,76,82}	0.605
{10,39,43,51}	0.822	{12,51,56,61,82}	0.578
{5,10,11,47}	0.663	{12,76,82}	0.578
{5,11,43,51}	0.663	{51,56,61,76,87}	0.556
{8,10,47}	0.663	{51,56,61,76,89}	0.556

5 Experiments and Results

Coronary arteriography is performed in patients with angina pectoris, unstable angina, previous myocardial infarction, or other evidence of myocardial ischemia. Patients with stenosis of the luminal narrowing greater than 0.5 were recruited as the CAD group, the others were classified as the normal. The accuracy was obtained by using the methodology of stratified 10-fold cross-validation. We compare our classifier with Naïve Bayesian and state-of-art classifiers; the widely known decision tree induction C4.5; an association-based classifier CBA [14] and CMAR [15], a recently proposed classifier extending NB using long itemsets. The result is shown on Table 4.

Table 4. Description of summary results

Classifier	Precision	Recall	F-Measure	Class	RMSE
Naïve	0.814	0.576	0.675	CAD	0.4825
Bayesian	0.659	0.862	0.747	Control	
C4.5	0.88	0.889	0.884	CAD	0.334
	0.882	0.872	0.877	Control	
CBA	0.921	0.939	0.93	CAD	0.2532
	0.935	0.915	0.925	Control	
CMAR	0.945	0.896	0.92	CAD	0.2788
	0.889	0.941	0.914	Control	
EP-based	0.960	0.938	0.951	CAD	0.2246
Classifier	0.945	0.957	0.938	Control	

In the experiments, the parameters of the five methods are set as follows. All NB and C4.5 parameters are default values. We test both C4.5 tree method and rule method. For CBA, we set support threshold to 0.05 and confidence threshold to 0.8 and disable the limit on number of rules. Other parameters remain default. For CMAR and our model, the support and confidence thresholds are set as same CBA. We used *precision*, *recall*, *f-measure* and *root mean square error* rate to evaluate the performance. The result is shown Table 6. As can be seen from the table, our classifier outperforms NB, C4.5, CBA and CMAR.

6 Conclusion

Most of parameters employ in diagnosing disease have both strong and weak points together. Therefore, it is important to develop multi-parametric indices diagnosing cardiovascular disease in order to enhance the reliability of the diagnosis.

According to the purpose of this paper, we built the classifier of cardiovascular disease and verified the effectiveness and accuracy of classification method. We extracted new multi-parametric features and used clinical information for diagnosing cardiovascular disease. We used extended emerging pattern by using EP cohesion measure for uninformative pattern pruning. Experimental results showed high accuracy and efficiency achieved by our classifier. When proposed classifier compared with other classifiers, it outperformed Bayesian classifiers, decision tree and associative classifiers in accuracy.

References

1. Cohen, A.: Biomedical Signal Processing. CRC press, Boca Raton, FL (1988)
2. Conumel, P.: ECG: Past and future, Annals NY Academy of Sciences. Vol.601 (1990)
3. Pan, J.: A real-time QRS detection algorithm, IEEE Trans. Eng. 32 230-236 (1985)
4. Taddei, A., Constantino, G., Silipo, R.: A system for the detection of ischemic episodes in ambulatory ECG, Computers in Cardiology. IEEE Comput. Soc. Press (1995) 705-708
5. Meste, O., Rix, H.: Ventricular late potentials characterization in time-frequency domain by means of a wavelet transform. IEEE Trans. Biomed. Eng. 41 (1994) 625-634
6. Thakor, N.V., Yi-Sheng, Z.: Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection. IEEE Trans. Biomed. Eng. 38 (1991) 785-794
7. Kuo, C.D., Chen, G.Y.: Comparison of Three recumbent position on vagal and sympathetic modulation using spectral heart rate variability in patients with coronary artery disease. American Journal of Cardiology 81 (1998) 392-396
8. Noh, K., Lee, H.G., Lee, B.J., Shon, H., Ryu, K.H.: Associative Classification Approach for Diagnosing Cardiovascular Disease. To be appeared in ICIC 2006 (2006)
9. Li, J, Mining emerging patterns to construct accurate and efficient classifier. Ph.D thesis, Melbourne University 2000
10. Duda, R., Hart, P.: Pattern classification and scene analysis. John Wiley New York, (1973)
11. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann San Mateo, (1993)
12. Fayyad, U.M., Irani, K.B.: Multi-Interval discretization of continuous-valued attributes for classification learning. In Proc. of the Interna'l Joint Conf. on AI (1993) 1022-1027
13. Forsyth, R., Rada, R.: Machine Learning applications in Expert Systems and Information Retrieval. Ellis Horwood Limited (1986)
14. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In Proc. of the 4th Interna'l Conf. Knowledge Discovery and Data Mining (1998)
15. Li, W., Han, J., Pei, J.: CMAR: Accurate and Efficient Classification Based on Multiple Association Rules, In Proc. of 2001 Interna'l Conf. on Data Mining (2001)

DNA Microarray Data Clustering by Hidden Markov Models and Bayesian Information Criterion

Phasit Charoenkwan¹, Aompilai Manorat¹, Jeerayut Chaijaruwanich¹,
Sukon Prasitwattanaseree², and Sakarindr Bhumiratana³

¹Department of Computer Science, Faculty of Science,
Chiang Mai University, Chiang Mai 50200, Thailand
jeerayut@science.cmu.ac.th

²Department of Statistics, Faculty of Science,
Chiang Mai University, Chiang Mai 50200, Thailand

³National Center for Genetic Engineering and Biotechnology (BIOTEC)
113 Thailand Science Park, Phahonyothin Road, Klong 1, Klong Luang,
Pathumthani, 12120, Thailand

Abstract. In this study, the microarray data under diauxic shift condition of *Saccharomyces Cerevisiae* was considered. The objective of this study is to propose another strategy of cluster analysis for gene expression levels under time-series conditions. The continuous hidden markov model was newly proposed to select genes which significantly expressed. Then, new approach of hidden markov model clustering was proposed to include Bayesian information criterion technique which helped to determine the size of model. The result of this technique provided a good quality of clustering from gene expression patterns.

1 Introduction

It is known that genes regulate the characteristics, behaviors, appearances and functions of living cells, and the attempt to establish the functional relationships between genes is important for many applications such as finding drug targets and increasing agricultural yields. Traditionally, the study of gene relationships is done within wet laboratories and could only be investigated for a single or small group of genes at a time. To understand gene relationships at the genomic scale, it becomes very expensive and fastidious works, because of the tremendous number of genes. The recent development of microarray technology has produced an explosion of transcriptional expression studies at the genomic scale for many species under different experimental conditions.

In general, it is supposed that a group of genes with similar expression patterns may be suggestive of an associated biological function. Therefore, clustering becomes a useful technique for the analysis of gene expression data. However, the gene expression from microarray data is only a snapshot of the complex and dynamic system of living cells. This temporal information is time dependent by nature, and should not be represented in vector space as done by many conventional clustering methods such as k-means and hierarchical clustering. This temporal and sequential signal, defines the

dynamic characteristics of the phenomenon under study. The generation of this signal could be viewed as a probabilistic walk through a fixed set of states. When the states are not directly observable, or it is not feasible to define states by exhaustive enumeration of feature values, the hidden markov model (HMM) is one of the most appropriate models [4] for this case.

Recent studies [2, 3] have used HMMs as model-based clustering to partition a set of temporal expression data into clusters. Each cluster had its associated HMM represent the common characteristics of genes in the cluster. The membership of a gene in a certain cluster was proposed to be determined by the likelihood of its expression pattern generated by the HMM of the cluster.

However, microarray data consists of large numbers of genes, and cluster analysis of such gene sets using HMMs take a very long time. To avoid this problem, this paper proposes the null model of left to right HMM to select only genes that significantly express during the time-course experiments for cluster analysis. Here, the microarray data under a diauxic shift condition for *Saccharomyces Cerevisiae* [5] was considered as a case study. For the cluster analysis, we employed a Bayesian information criterion to determine the appropriate number of states of our continuous HMM clustering model.

2 Methods

2.1 Hidden Markov Model

A continuous hidden markov model is a finite set of states, transition, and observation probabilities. Each state is associated with the observation probability density function represented by a finite mixture Gaussian model [1], and transitions among the states are defined by a set of transition probabilities. HMMs can be defined by complete parameter set of the model in the following form:

$$\lambda = (\pi, a, b) \tag{1}$$

where π is the prior probability of the initial state of N states, a is the state transition probability distribution denoted by $a = \{a_{ij}\}$, and a_{ij} is the transition probability from state i to state j while can be written as:

$$a_{ij} = P[q_t = state_i \mid q_{t+1} = state_j], 1 \leq i, j \leq N \tag{2}$$

where q_t is the state at time t . For the special case where any state can reach any other state in a single step, we have $a_{ij} > 0$ for all i , and j .

The continuous observation probability distribution b in state j is formulated using the mixture Gaussian model:

$$b_j(O) = \sum_{m=1}^M c_{jm} \eta[O, \mu_{jm}, U_{jm}], 1 \leq j \leq N \tag{3}$$

where η is the probability density function, M is the number of mixtures, c is the mixture coefficient, μ is the mean, and U is the covariance matrix of the observations.

2.2 Left-to-Right Hidden Markov Model

LRHMM (Left-to-Right Hidden Markov Model), see illustration in Fig.1., is the hidden markov model which only allows the transitions of states from left to right, and could allow transition jumps. The prior probability of LRHMM is set to 1 at the first state and 0 for any other. Continuous observation probability distributions are the same as in a general HMM.

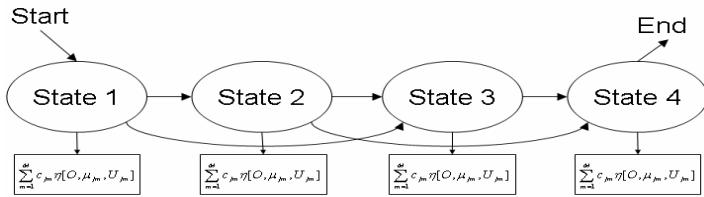


Fig. 1. Two jumps Left to Right Hidden Markov Model

2.3 Bayesian Information Criteria (BIC)

Generally, the appropriate number of states for an HMM is not known. Bayesian information criterion (BIC) could be employed to find the appropriate structure of the HMM [4]. BIC is a model scoring derived from the Laplace approximation:

$$\log P(M | X) \approx \log P(X | M, \theta) - \frac{d}{2} \log N \quad (4)$$

where d is the number of significant parameters (larger than 10^{-6}) of the initial, transition, and emission probabilities, in the HMM model M , N is the number of genes in expression dataset X used for training model M , and θ is the configuration parameter of model M . $\log P(X|M, \theta)$ is the total log-likelihood, which tends to promote larger and more detailed models of data, whereas the second term $-d/2 \log N$, is the penalty term which favors smaller models with less parameters. BIC selects the best model M for the data X by balancing these two terms.

BIC is used to express the quality of HMM. The appropriate number of states for the HMM is determined by the maximum value of BICs. The maximum number of states must not exceed the number of time points observed.

2.4 Hidden Markov Model Clustering Algorithm

The idea to construct our HMM clustering model was adapted from the incremental k-means algorithm. The algorithm for construction of HMM clustering is as follow.

```

Algorithm Construction of HMM Clustering Model
Input: Gene expression data
Output: HMM clustering model
k = 1
While (gene pool is not empty)
{
    Randomly select a gene from the gene pool to
    construct HMM of cluster k
    While (clustering is changed)
    {
        For each gene i in gene pool
        {
            Evaluated the likelihood of emerging ex-
            pression of gene i from HMM of cluster k
            If (likelihood >= Threshold) Then
                Assign gene i to HMM cluster k
            }
            Reconstruct HMM again from genes in the
            cluster k
        }
        Remove genes in cluster k from the gene pool
        Increment k
    }
}

```

Fig. 2. Algorithm for construction of HMM clustering model

3 Experiment

We used microarray data for *Saccharomyces Cerevisiae*, in diauxic shift experiment condition. The expression data, obtained from [5], recorded the fluctuation of expression levels of approximately 6,400 genes at 7 time-courses; 9h, 11h, 13h, 15h, 17, 19h and 21h. Genes which contained any missing expression values were deleted.

Due to the large number of genes to be clustered, the computational time for construction of an HMM clustering model could be very long, and the initial model for each cluster might be far from optimal. To avoid this problem, genes which significantly expressed were detected first by HMM filtering model then used as the seed to construct the overall HMM clustering.

The HMM filtering model was constructed using a null model of LRHMM of seven states. Each state allowed two jumps. The observation probability distribution at each state was initially set to be normal and their mean and variance were fixed to be zero and one respectively. All gene expression data were used as training set to construct transition and observation probabilities. The obtaining HMM null model would represent the machinery of stationary or non expressed genes. The likelihoods of emerging expression patterns of all genes from the HMM null model were evaluated. Genes with low likelihood values were considered to be significantly expressed.

After the significant genes were selected, the HMM clustering was constructed. The Bayesian information criterion was used in the HMM training process to find the appropriate number of states for the HMM. HMM with the highest score of BIC was

chosen. To reduce the running time of HMM construction, HMM clustering model of each cluster was firstly constructed from a small set of significant genes, then improved from the whole set of significant genes.

4 Results and Discussion

After, the HMM null model was constructed and the likelihoods of emerging expression patterns of all genes from the model were evaluated. From the likelihood values, the genes were ranked into three groups. Fig.3 (a),(b) and (c) show the expression patterns of genes and the range of likelihoods of each group. The numbers of genes in groups (a),(b), and (c) are 5289, 516, and 27 respectively.

In Fig.3(a), the expression of genes were stationary and consisted of a large number of genes. These genes were considered to be irrelevant for the expression analysis, and were separated from the clustering analysis. In Fig.3(b), the expression pattern

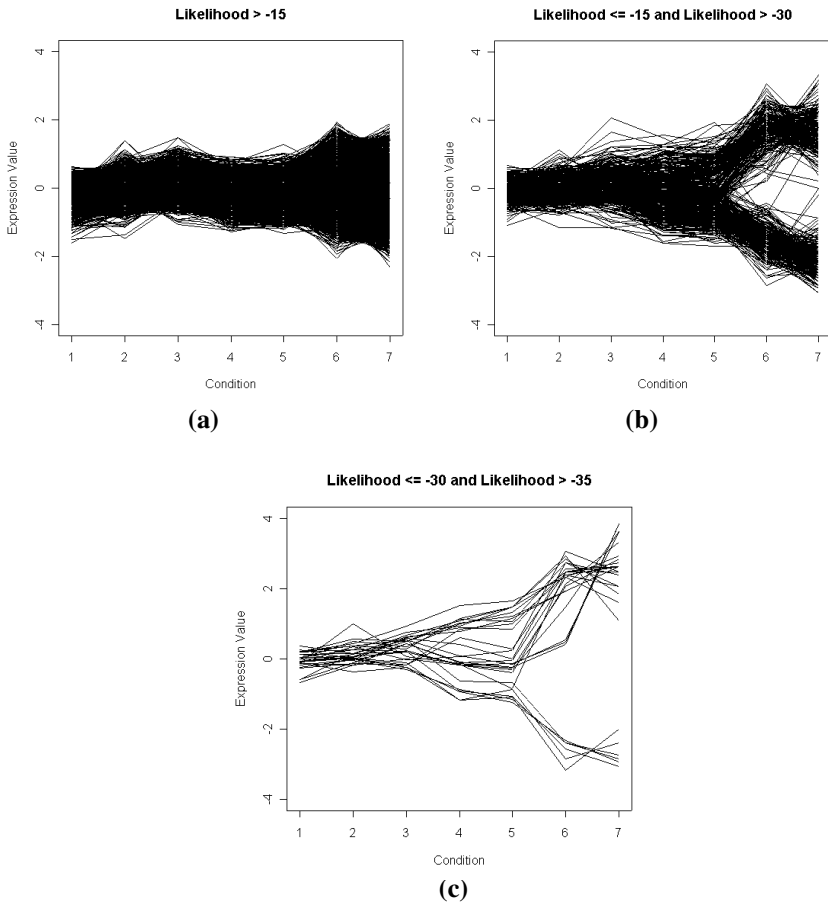


Fig. 3. Expression patterns of three groups of genes

looked to have two groups but the number of genes was still large. The genes of Fig.3 (c) significantly expressed, and were chosen to construct the HMM clustering model.

Fig.4(a), and (b) show clustering results of significant genes from Fig.3(c). In Fig.4(a), the expression pattern slowly increased at first stages and rapidly increased in the last two periods. In Fig.4(b), the expression pattern gradually decreased. After the HMM clustering model had been constructed, it was applied to a whole set of significant genes. Fig.4(c), and Fig.4(d) show clustering results of all significant genes from Fig.3(b).

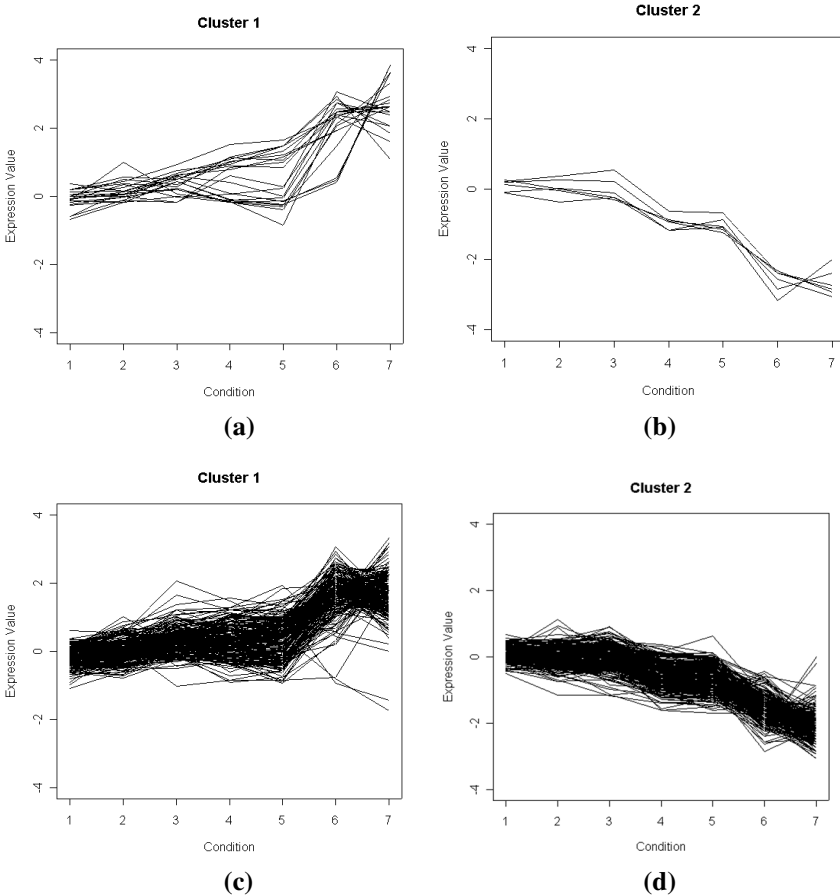


Fig. 4. Clustering results of significant genes

Table 1 (a) and (b) show the molecular functions and biological process of some most significant genes from Fig.4(a) and (b). Most of the genes with known functions were found to be in carbohydrate metabolism, and involved diauxic-shift condition [6]. It is proposed that the genes with unknown function in Table 1 might involve also

Table 1. Molecular function and biological process of most significant genes

(a) Cluster 1

Gene Name	Molecular Function	Biological Process
CYB2	L-lactate dehydrogenase activity	Pyruvate metabolism
YKL187C	Unknown	Unknown
YDL085W	Unknown	Unknown
FBP1	fructose-bisphosphatase activity	Glycolysis
PCK1	phosphoenolpyruvate carboxykinase activity	TCA cycle
ICL1	isocitrate lyase activity	Glyoxylate and Dicarboxylate metabolism
ACS1	acetate-CoA ligase activity	Glycolysis
YBL049W	Unknown	Unknown
COX5B	cytochrome-c oxidase activity	Oxidative phosphorylation
HSP31	Unknown	Unknown
YBR147W	Unknown	Unknown
SDH4	succinate dehydrogenase activity	TCA cycle
STF2	Unknown	Unknown
HSP104	heat shock protein activity	Unknown
YLR149C	Unknown	Unknown
SDH3	succinate dehydrogenase activity	TCA cycle
ACO1	Unknown	Unknown
LSC2	succinate-CoA ligase activity	TCA cycle
ALD4	aldehyde dehydrogenase activity	Glycolysis
PNC1	Unknown	Unknown
HXT7	glucose transporter activity	Unknown

(b) Cluster 2

Gene Name	Molecular Function	Biological Process
YLR413W	Unknown	Unknown
RLP7	Unknown	Unknown
YJL109C	snoRNA binding	Unknown
MKC7	aspartic-type signal peptidase activity	Unknown
GPP1	glycerol-1-phosphatase activity	Unknown
SAM1	methionine adenosyltransferase activity	Methionine metabolism

in carbohydrate metabolism and diauxic-shift condition. This discovery was useful for molecular cellular study.

5 Conclusion

This paper proposed the use of HMM models to identify and cluster significantly expressed genes from the time-course expression data. The experimental results show that these methods could be well applied to time-course microarray data and provide interesting biological results.

Acknowledgements

The authors thank Asso.Prof.Dr. Robert W. Cutler from Bard College, US, for his useful comments. This project was supported by the National Center for Genetic

Engineering and Biotechnology, Thailand and the Faculty of Science, Chiang Mai University, Thailand.

References

1. Rabiner, L.B.: A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, vol. 77, NO. 2 (1986)
2. Schliep, A.: Using hidden Markov models to analyze gene expression time course data. Bioinformatics, vol. 19 Suppl. 1 (2003) i255-i263.
3. Schliep, A.: Robust inference of groups in gene expression time-courses using mixtures of HMMs. Bioinformatics, vol. 20 Suppl. 1 (2004) i283-i289.
4. Li, C.: A Bayesian Approach to Temporal Data Clustering using Hidden Markov Models. International Conference on Machine Learning (ICML 2000), Stanford, California (2000) 543-550
5. Eisen Lab.: Public microarray expression data for yeast *Saccharomyces cerevisiae*. [Online]. Available: <http://rana.lbl.gov/EisenData.htm>.
6. DeRisi, J. L.: Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale. Science, vol 278, Issue 5338, (1997) 680-686.

Application of Factor Analysis on *Mycobacterium Tuberculosis* Transcriptional Responses for Drug Clustering, Drug Target, and Pathway Detections

Jeerayut Chaijaruwanich¹, Jamlong Khamphachua¹,
Sukon Prasitwattanaseree², Saradee Warit³,
and Prasit Palittapongarnpim³

¹Department of Computer Science, Faculty of Science,
Chiang Mai University, Chiang Mai, 50200, Thailand
jeerayut@science.cmu.ac.th

²Department of Statistics, Faculty of Science,
Chiang Mai University,
Chiang Mai, 50200, Thailand

³National Center for Genetic Engineering and Biotechnology (BIOTEC)
113 Thailand Science Park, Phahonyothin Road,
Klong 1, Klong Luang,
Pathumthani, 12120, Thailand

Abstract. Recently, the differential transcriptional responses of *Mycobacterium tuberculosis* to drug and growth-inhibitory conditions were monitored to generate a data set of 436 microarray profiles. These profiles were valuably used for grouping drugs, identifying drug targets and detecting related pathways, based on various conventional methods; such as Pearson correlation, hierarchical clustering, and statistical tests. These conventional clustering methods used the high dimensionality of gene space to reveal drug groups basing on the similarity of expression levels of all genes. In this study, we applied the factor analysis with these conventional methods for drug clustering, drug target detection and pathway detection. The latent variables or factors of gene expression levels in loading space from factor analysis allowed the hierarchical clustering to discover true drug groups. The t-test method was applied to identify drug targets which most significantly associated with each drug cluster. Then, gene ontology was used to detect pathway associations for each group of drug targets.

1 Introduction

The study of differential transcriptional responses of *Mycobacterium tuberculosis* to drug and growth-inhibitory conditions has been done by Boshoff et al. in 2004 [1]. In that study, 436 microarray profiles of different drugs treatments were generated. The treatments were clustered according to their similar pattern of transcriptional expression and the relevant genes of each group of treatments were associated. The techniques used were unsupervised gene selection, interrelated clustering and biclustering.

However, it seemed that the confidence level of these methods was limited by the small number of experiment samples compared to the large number of genes. Such high dimensionality of data led to an over fitting problem in the estimation of model parameters. Moreover, most of the genes were irrelevant or redundant in their behaviors. This noise contamination made the cluster analysis such as hierarchical clustering less reliable. With conventional clustering methods, the genes hardly clustered into non-overlapping groups, and might not correspond to the real complex behavior of the living cells. To overcome these inconveniences, this study employed new paradigms of cluster analysis based on principal factor analysis to cluster drug treatments, and used statistical t-tests to select target genes which most significantly associated with each drug cluster. Furthermore, since the ontology for these genes is already known [5], we were able to detect related active pathways for each drug group.

Factor analysis [3] is a multivariate statistical technique, especially concerned with latent variable modeling. The central hypothesis used with factor analysis on DNA microarray data proposes that the observed expression levels of genes linearly combine by a set of latent variables, called factors, of the observable expression levels, as illustrated in Fig. 1.

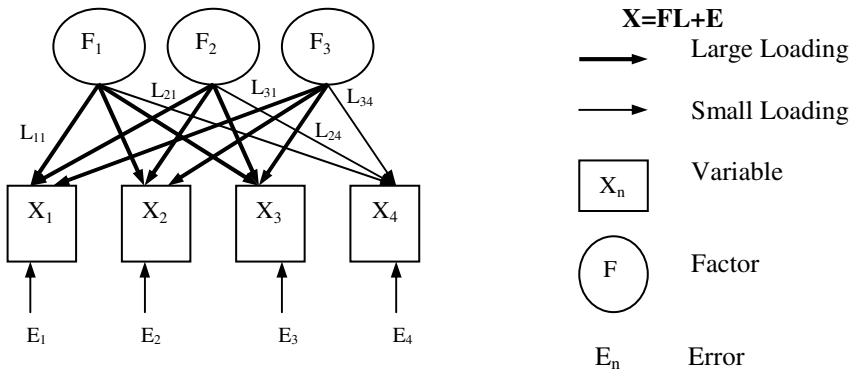


Fig. 1. Illustration of Factor Analysis Model

In principal factor analysis, each factor extracted by the principal component analysis algorithm attempts to summarize the total variance of data. The contribution of each factor to the observed expression levels of the genes in each drug is represented by the elements of the loading matrix L, i.e. arrows in Fig. 1. To avoid over fitting and obtain the independence of small number of factors compared to the number of genes, loading factors would be used in cluster analysis instead of the original expression levels. The factors could be rotated into orthogonal space. Next, the t-test method was applied to identify relevant genes associated with each group of treatments, where some genes

can be found in several clusters. Finally, the pathway associated with each group of drugs was identified by using the ontology of corresponding genes.

2 Materials and Methods

2.1 Datasets

The experimental dataset consisted of transcriptional responses of *Mycobacterium Tuberculosis* to drug and growth-inhibitory conditions [1]. It originally consisted of spotted array measurements of 4,320 genes in 436 drug treatment sets such as [1ug/mL No.121940: DMSO, 12h (mAdb expid=44709)], [1ug/mL No.111891: DMSO, 12h (mAdb expid=44710)], [24ug/mL clotrimazole: DMSO, 6h (mAdb expid=44713)], etc. Microarray data have been deposited in the Gene Expression Omnibus at NCBI (www.ncbi.nlm.nih.gov/geo) with GEO accession numbers GSE1642 and GSE 1694.

2.2 Missing Values Elimination

Missing values, denoted by “NA”, in the microarray data were detected and filtered out. The treatments which had percents of missing values less than or equal to 5% were chosen.

2.3 Data Normalization

After the missing values had been filtered out, the standard normalization method was applied to the data. By this method, the expression levels were centered by subtracting from their mean and dividing by their standard deviation. The centered distributions for each treatment would have mean equal to 0 and standard deviation equal to 1.

2.4 Factor Analysis

The next process of analysis was the determination of drugs groups by using factor analysis and clustering algorithm. Factor analysis [3] constructed an underlying orthogonal factor model of an original gene expression X-matrix $n \times m$ (where n was the number of genes and m was the number of drugs) of the form:

$$X=FL+E \tag{1}$$

L was the loadings matrix of size $k \times m$, where k was the number of factors, and F was the scores matrix of size $n \times k$, and E was the residual matrix, which contained both specific variance of individual gene and errors in the model (see Fig. 1.). This study used the so-called principal factor solution to construct this factor model. Specifically, in the first step, basing the correlation matrix R (using Pearson correlation method) derived from X , communalities (i.e. the proportion of the variance explained by common factors) were computed from the multiple squared correlation coefficient

between the i th variable and the rest. These communalities replaced the diagonal (Λ) entries of the correlation matrix, which was subjected to diagonalization. New communalities were computed from the loadings at the chosen dimensionality. The first few factors which had variance ≥ 1 were chosen, and obtained by scaling the eigenvector matrix (H), as follow.

$$H\Lambda^{1/2} \quad (2)$$

Finally, the factor loadings were rotated by *varimax* rotation method [2]. The model of rotation was as follow.

$$L_{\text{new}} = L_{\text{old}}T \quad (3)$$

T was the orthogonal transformation matrix estimated by varimax algorithm. The purpose of this rotation was to maximally project each drug onto one of the factors in order to simplify the factor model and make it more readily interpretable.

2.5 Hierarchical Clustering

Next, the average linkage hierarchical clustering method [2, 4] was used to cluster drugs in loading space at the optimal dimensionality. The groups of drugs were manually selected from the tree of clustering.

2.6 Drug Targets Detection

This process was to find the set of genes associated with each drug cluster. Once drug clusters were found, the groups of genes contributing heavily to the specific characteristic of each group were identified by using a Student's *t*-test [4]. It measured the differential expression of the genes in the cluster as comparing with the rest. *T*-test scores were transformed to p-values. Genes with p-value lower than 0.05 were taken as differentially expressed for that particular cluster.

2.7 Pathway Detection

The final objective was to find pathways which were effected by each of drug groups. Since, functions of genes were assigned by their ontology, fundamentally it is gene ontology that was used to detect pathways associated with drug clusters. Gene ontology (GO) can be described in three categories; molecular function (MF), biological process (BP) and cellular component (CC), where the ontology of some genes was not annotated yet, the pathways of those genes were report as unknown.

3 Results and Discussion

Representation of dataset is shown in Table 1. The missing expression values, denoted by "NA", were detected and filtered out. The number of drug treatments and genes left after filtering the missing values reported in Table 2. and Fig. 2.

The treatments which had percents of missing value less than or equal to 5% were chosen. Any gene which had missing value in chosen treatments was eliminated. Consequently, 159 drug treatments and 1,143 genes were chosen for this study.

Table 1. Example of gene expression data

Genes	Drugs ID						
	GSM28217	GSM28218	GSM28219	GSM28220	GSM28221	GSM28222	GSM28223
Rv0001	0.644006	0.434344	0.196201	0.168632	-0.258616	-0.142778	-1.488008
Rv0002	-0.922116	-0.831289	-0.873130	-1.381421	-1.006963	-1.589476	-0.204861
Rv0003	NA	NA	NA	2.262851	NA	1.945534	NA
Rv0004	1.371151	NA	1.673556	NA	NA	1.553278	1.859460
Rv0005	NA	NA	NA	NA	NA	NA	NA
Rv0006	0.231554	0.187419	NA	0.581418	0.657052	NA	0.173467

Table 2. Number of drugs and genes left after filtering the missing values

Percent of missing value(0-this value) in each drugs	drugs	Genes
1%	4	3827
2%	49	2940
3%	87	2183
4%	125	1524
5%	159	1143
6%	180	864
7%	205	681
8%	228	548
9%	257	413
10%	269	338

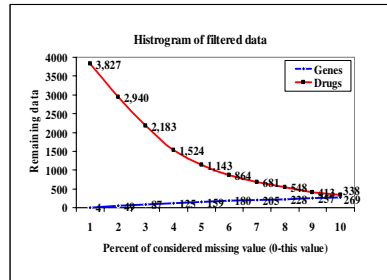


Fig. 2. Number of drugs and genes left after filtering the missing values

Fig.3. showed the example of distributions of data before and after normalization. From the missing value preprocessing, the set of 159 drug treatments and 1,143 genes were selected. The correlation matrix of expression ratios were calculated and shown as example in Table 3.

After performing factor analysis from the correlation matrix, the standard deviation, variance, cumulative variances, proportion of variances and cumulative proportion of variances of each factor were calculated and shown in Table 4. Twenty four factors explaining variance larger than 1 were selected. The cumulative proportion of variances of chosen factors was equal to 0.8361.

Table 5. showed the factor loading of first three factors. These loading values indicated the contribution impact of drug in each factor. The large positive loading values

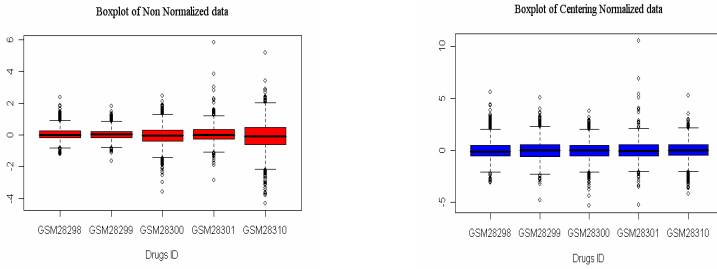


Fig. 3. Box plot of original and normalized gene expression of five treatments. The line at the center of each box represented the mean value of distribution; the size of the box represented the standard deviations of the distribution; the two horizontal lines bracketing the box represented the extreme values of the distribution.

Table 3. Example of correlation matrix(R)

Drug ID	GSM28298	GSM28299	GSM28310	GSM27862	GSM27994	GSM28013
GSM28298	1	0.689691	0.353218	0.098556	0.255518	0.279179
GSM28299	0.689691	1	0.243459	0.214843	0.196886	0.205264
GSM28310	0.353218	0.243459	1	-0.255594	0.380235	0.344468
GSM27862	0.098556	0.214843	-0.255594	1	0.031196	0.015443
GSM27994	0.255518	0.196886	0.380235	0.031196	1	0.930862
GSM28013	0.279179	0.205264	0.344468	0.015443	0.930862	1

Table 4. Summary of factor analysis

Component	S.D.	Var	Cum.Var	Prop.of Var.	Cum. Prop.of Var.
Factor1	6.3328	40.1048	40.1048	0.2522	0.2522
Factor2	3.7435	14.0143	54.1192	0.0881	0.3404
Factor3	3.5644	12.7051	66.8242	0.0799	0.4203
...
Factor.24	1.0023	1.0045	132.9351	0.0063	0.8361
Factor.25	0.9652	0.9317	133.8668	0.0058	0.8419

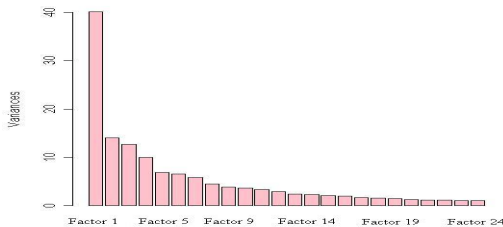


Fig. 4. Scree plot of explaining variance

indicated highly positive influence of that drug in the factor, and vice-versa. It could be observed that the drug id GSM28298 and drug id GSM28299, whose correlation coefficient was high in Table 3., had similar loading values in the first three factors in Table 5.

The matrix of 159 drugs and 24 factors was applied by hierarchical clustering. The tree of hierarchical clustering was shown in Fig.5. (a). From this tree of clustering, 20 clusters were identified manually.

Table 5. Samples of factor loading matrix (L)

	Factor1	Factor 2	Factor3
GSM28298	-0.327567	0.030084	-0.059268
GSM28299	-0.229162	0.061318	-0.088495
GSM28300	-0.229566	-0.335290	-0.120677
GSM28013	-0.189935	-0.163244	-0.050001
GSM27994	-0.112073	-0.225375	-0.084350
GSM28104	-0.042683	-0.781241	-0.022499
GSM28105	-0.047431	-0.765431	-0.027159

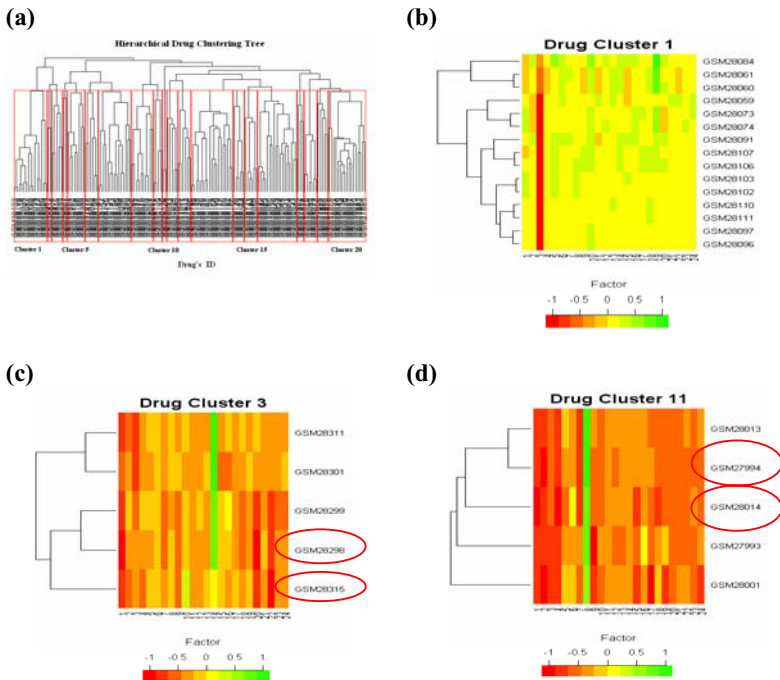


Fig. 5. Hierarchical clustering of drug treatments

Fig.5. (b), Fig.5. (c) and Fig.5. (d) showed the heat map of hierarchical clustering of the 1st, 3rd, and 11th clusters respectively. The color in the figure indicates the level of factor loading with red when low (-1), and green when high (+1). Drugs in the same cluster would have similar pattern of loading factors. From Fig.5. (c), the drugs id GSM28298 and GSM28299 were found in the same cluster and similarly for the drugs id GSM27994 and GSM28013 in Fig.5. (d). Furthermore, it found that the correlation coefficients of both cases were, see Table 5.

When applying t-tests to each of 20 clusters, a large number of (i.e. 799 among 1,143) genes were found for the 1st cluster, and many genes were found for several drug clusters, see Table 6. Table 6. reported resulting details of the 3rd cluster of drugs such as group of drug ids, drug treatments, number of related genes, list of related genes and list of pathways or three principal ontology found.

Table 7. summarized the experimental results, i.e. group of drug, number of drugs in each cluster, number of related genes in each group, number of genes which annotated by gene ontology, and number of corresponding pathways.

Table 6. Numbers of drug clusters, drug targets and pathways found

Group	Number of drugs	Number of related genes	Number of genes which annotated by gene ontology	Number of corresponding pathways		
				Molecular function	Biological process	Cellular component
1	15	799	189	189	151	23
2	2	238	65	98	81	17
3	5	354	84	135	96	16
4	2	156	42	75	55	12
5	8	476	115	142	121	18
6	6	466	121	141	102	22
7	15	665	160	175	143	23
8	10	611	136	184	122	21
9	4	382	73	115	87	15
10	2	437	110	147	117	18
11	5	393	87	116	96	16
12	6	370	87	136	97	13
13	19	685	145	165	121	23
14	5	506	101	134	95	20
15	6	542	134	150	128	20
16	18	639	134	181	133	18
17	3	245	80	100	87	16
18	6	486	117	131	109	19
19	5	338	52	93	76	16
20	17	690	156	174	141	21

Table 7. Examples of results of the 3rd drug group

Drug ID	Treatments	Related genes	
		Number	Examples
GSM28298	2mM b-mercaptoethanol: DMSO, 6h (mAdb expid=49262)	354 genes	Rv0015c, Rv0016c, Rv0019c, Rv0031, Rv0038, Rv0039c, Rv0044c, Rv0046c, Rv0054, Rv0074, Rv0088, Rv0093c, Rv0096, Rv0103c, Rv0113, Rv0145, Rv0153c, Rv0158, Rv0183, Rv0198c, Rv0200, etc.
GSM28299	2mM DTNB: DMSO (mAdb expid=49263)		
GSM28301	50uM Nigericin: DMSO, 6h (mAdb expid=49265)		
GSM28311	50uM Nigericin: DMSO, 6h (mAdb expid=49333)		
GSM28315	0.1mM GSNO/10ug/mL menadione: DMSO, 6h (mAdb expid=49337)		
Examples of Pathways			
Molecular Function	Biological Process	Cellular Component	
transferase activity,ATP binding,metal ion binding,structural constituent of ribosome,catalytic activity,oxidoreductase activity,nucleotide binding,RNA binding,iron ion binding, etc.	protein biosynthesis, Molybdopterin cofactor biosynthesis, metabolism, electron transport, fatty acid biosynthesis, carbohydrate metabolism, tricarboxylic acid cycle, porphyrin biosynthesis, lipid metabolism, fatty acid metabolism, etc.	integral to membrane, membrane, cytoplasm, signal recognition particle (sensu Eukaryota), cell wall, intracellular, ribosome,small ribosomal subunit, ribonucleoprotein complex, large ribosomal subunit, etc.	

4 Conclusions

From the original dataset of DNA Microarray of *M. tuberculosis* containing 4,320 genes and 436 drug treatments, this study showed that after considering the missing values elimination, data normalization and high dimensionality reduction of data using factor analysis technique, only 159 drugs and 1,143 genes could be reliably analyzed. 20 clusters of drugs were found, and the relevant genes of each drug cluster were identified. The set of pathways of these drug targets which were annotated by gene ontology were also identified. However, the lists of genes and these pathways found were not yet validated. The biological function analysis or profound studies must be further performed.

Acknowledgements

The authors thank Asso.Prof.Dr. Robert W. Cutler from Bard College, US, for his useful comments. This project was supported by the National Center for Genetic Engineering and Biotechnology, Thailand and the Faculty of Science, Chiang Mai University, Thailand.

References

1. Boshoff, H.I.M, et al.:The Transcriptional Responses of *Mycobacterium tuberculosis* to Inhibitors of Metabolism. JBC. Vol.279, No.38 (2004) 40,174-40,184.
2. Johnson, R.A., Wichern, DW.:Applied Multivariate Statistical Analysis. Upper Saddle River, NJ, Prentice Hall (1992) 477-510,679-689.
3. Lozano, J.J., et al.:Dual activation of pathways regulated by steroid receptors and peptide growth factors in primary prostate cancer revealed by Factor Analysis of microarray data. BioMed Central (2005)
4. Stekel, D.:Microarray Bioinformatics.Cambridge University Press, Cambridge (2003) 90-97,112-123,158-168.
5. The Gene Ontology Consortium.:The Gene Ontology. <http://www.geneontology.org/index.shtml> (2006)

First Steps to an Audio Ontology-Based Classifier for Telemedicine

Cong Phuong Nguyen, Ngoc Yen Pham, and Eric Castelli

International Research Center MICA
HUT – CNRS/UMI2954 – INPGrenoble
1, Dai Co Viet, Hanoi, Vietnam
{Cong-Phuong.Nguyen, Ngoc-Yen.Pham,
Eric.Castelli}@mica.edu.vn

Abstract. Our work is within the framework of studying and implementing a sound analysis system in a telemedicine project. The task of this system is to detect situations of distress in a patient's room based on sound analysis. If such a situation is detected, an alarm will be automatically sent to the medical centre. In this paper we present our works on building domain ontology of such situations. They gather abstract concepts of sounds and these concepts, along with their properties and instances, are represented by a neural network. The ontology-based classifier uses outputs of networks to identify classes of audio scenes. The system is tested with a database extracted from films.

1 Introduction

In recent years telemedicine is widely studied and applied. It can be broadly defined as the transfer (e.g. telephone lines, the Internet, satellites, etc) of electronic medical data (e.g. images, sounds, live video, patient records, etc) from one location to another. The system that we present is developed for the surveillance of elderly, convalescent persons or pregnant women. Its main goal is to detect serious accidents such as falls or faintness at any place in the apartment. If a serious accident is detected, an alarm will be automatically sent to the medical centre. Firstly most people do not like to be supervised by cameras all day long while the presence of microphone can be acceptable. Secondly the supervision field of a microphone is larger than that of a camera. Thirdly, sound processing is much less time consuming than image processing, hence a real time processing solution can be easier to develop. Thus, the originality of our approach consists in replacing the video camera by a system of multichannel sound acquisition. The system analyzes in real time the sound environment of the apartment and detects abnormal sounds (falls of objects or patient, scream, groan) that could indicate a distress situation in the habitat.

This system is divided into different small modules. In the process of developing it, a sound acquisition module, a sound of daily life classifier, a speech/nonspeech discriminator and a speech/scream-groan discriminator are constructed [10], [11], [12], [18]. The habitat we use for our experiments is a 30m² apartment (depicted in Fig. 1) equipped with various sensors, especially microphones. There is one microphone in each room (toilet, kitchen, shower-room, hall and living-room) of the apartment. This

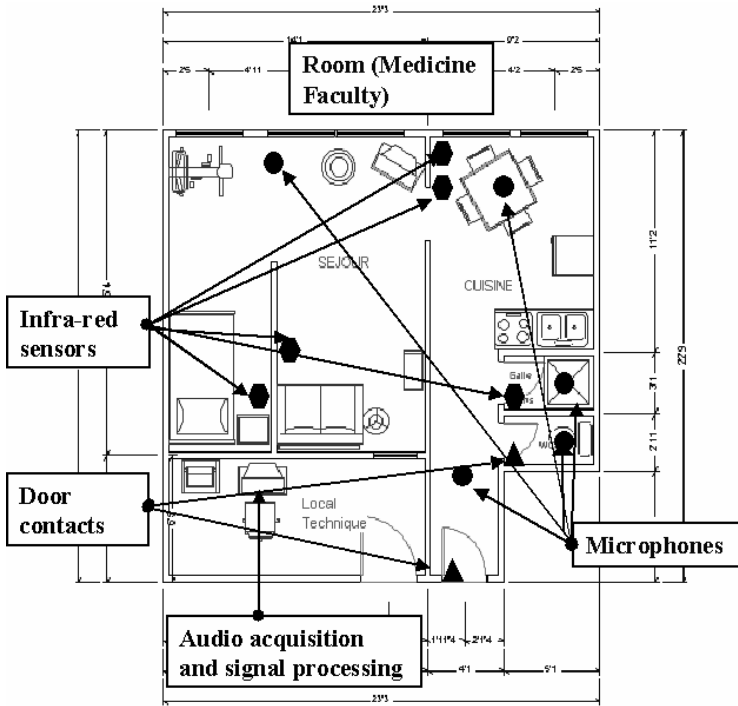


Fig. 1. The apartment used for telemedicine. Microphones are installed in each room in order to assure the sound surveillance in the whole apartment.

installation allows a sound surveillance in the whole apartment. In this apartment, audio signals are acquired by five microphones and feed to a multi-channel data acquisition card (National Instruments DAQ) installed on a slave computer. Sound source can be localized through comparison of the sound levels of the microphones. If two simultaneous detections are recorded, only the channel with the maximum signal level is considered. The microphones used are omni-directional, condenser type, small size and low cost. A signal conditional card consisting of an amplifier and an anti-aliasing filter is attached to each microphone. The four modules mentioned above, developed in LabWindows/CVI, process acquired signals to detect situations of distress.

The combination of these four modules seems to be an audio classification system. It can be seen that audio classification has been studied for many years. They are applied to speech recognition, audio content-based analysis, audio segmentation, audio retrieval, broadcast news transcription, etc. Works described in [22], [4], [16], [28] are four examples among many systems. But there is a difference between the problem of audio classification and our problem. Audio classification in the literature is applied to classify different homogeneous audio segments, i.e. each segment consists of a unique type of audio signal (e.g. speech). Meanwhile, our system is intended to classify an audio scene containing different types (e.g. a segment of scream and a segment of fallen chair). In other words, an audio scene's category is determined by

its types of segment. In order to complete the system, we propose sound ontologies which can be used in an ontology-based classifier. Ontology is an abstract concept of an audio scene representing a situation of distress in the house. It can be used to detect situations of distress, to classify audio scenes, to share information of audio scenes among people and software, to analyze domain knowledge, or to save (as metadata) audio scenes in a database for further usages.

This article is structured as follows. Sect. 2 discusses works related to ontology-based audio applications. Sect. 3 describes the proposed ontology of audio scenes and the neural network used to represent ontologies and the ontology-based classifier. Sect. 4 presents the database of audio scene and the evaluation of ontologies. Sect. 5 outlines our conclusion and next steps in future to complete this ontology-based system.

2 Related Work

Ontology has been researched and developed for years. In audio applications, it is applied probably for the first time by Nakatani and Okuno [19]. They propose a sound ontology and its three usages: ontology-based integration (for sound stream segregation), interfacing between speech processing systems, and integration of bottom-up and top-down processing. Their sound stream segregation means generating an instance of a sound class and extracting its attributes from an input sound mixture. Khan and McLeod [13] utilize a domain-specific ontology for the generation of metadata for audio and the selection of audio information in a query system. In this work, an audio ontology is defined by its identifier, start time, end time, description (a set of tags or labels) and the audio data. MPEG-7 Description Definition Language and a taxonomy of sound categories are employed by Casey [5] for sound recognitions. The audio content is described by qualitative descriptors (taxonomy of sound categories) and quantitative descriptors (set of features). Amatriain and Herrera [1] use semantic descriptors for sound ontology to transmit audio contents. Their description includes both low-level descriptors (e.g. fundamental frequency) and high-level descriptors (e.g. 'loud'). WordNet, an existing lexical network, is used by Cano et al. in [3] as a ontology-backbone of a sound effects management system. Ontology is applied to the disambiguation of the terms used to label a database in order to define concepts of sound effects, for example, the sound of a jaguar and the sound of a Jaguar car. A system of ontology-based sound retrieval is proposed by Hatala *et al* in [9] to serve museum visitors. Ontology is used to describe concepts and characteristics of sound objects as an interface between users and audio database. In [14], Kim *et al* develop an MPEG-7-based audio classification and retrieval system targeted for analysis of film material. They test three structures for sound classification: a one-level, a hierarchical, and a hierarchical with hints. The classification is based on the Euclidean distance. The second structure gives the lowest recognition rate, while the third one gives the highest recognition rate. In most cases, ontology is used to describe an audio file (e.g. a sound effect) and to manage the database.

Our work is to build a sound ontology applied to classifying an unknown audio sample detected in habitat. We will present in the next section ontology for abstract concepts of audio scenes and for detecting situations of distress.

3 Ontology-Based Sound Classification

An ontology-based sound classification includes three problems: defining ontology, representing it, and applying it to classification. These problems will be presented in this section.

3.1 Sound Ontology

From the classified sounds, we intend to extract the true meaning of the scene. For example, when a sound of a fallen chair and a sound of scream are detected, it should be interpreted as “the patient has tumbled down” (making the chair fall). Or when we detect a sound of groan, we can say that the patient is ill or hurt. This is a mapping between concrete sounds and abstract concepts. In other words, an abstract concept is defined by determined sounds. Sound ontology seems to be the most appropriate to our work.

Ontology is defined as a set of representational terms as defined by Gruber in [8]. In this paper, some situations are defined as ontology. Ontology consists of concept, concept properties, relations and instances. The hierarchical relations between concepts in text or image applications can be established. But in our applications, such relations between situations are not obvious. So we simply define concepts of situations based on concept properties and their facets. The ontology of the situation of patient falling in house is depicted in Fig. 2 as an example. When the patient falls, he

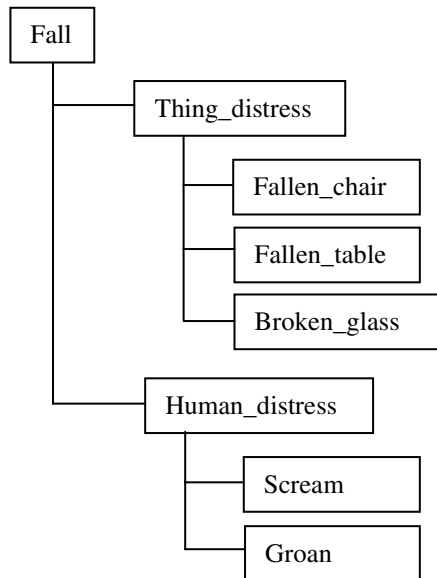


Fig. 2. An example of the ontology representing the “fall” concept. It has two properties and five facets.

Table 1. Properties and facets of three ontologies. “Hurt” concept has no thing_distress property. “Water_in_toilet” facet taken into account is due to the fact that when the patient is sick/ill in the toilet, sound of water is often detected.

	Thing_distress	Human_distress
Fall	Fallen_chair, Fallen_table, Broken_glass	Scream, Groan
Hurt		Scream, Groan
Sick/ill	Water_in_toilet	Cough, Pant, Vomit

can make a chair or a table fall over, or can break something such as a glass. After the fall, the patient probably screams or groans due to pain or shock. Sounds are divided into two categories. Sounds of fallen chair, fallen table and broken glass are categorized in the “thing_distress” class, sounds of scream and groan are of “human_distress”. So if a sound of “thing_distress” and/or a sound of “human_distress” are successively detected, we can probably say that the patient has fallen. Of course it can also be said that those sounds come from a chair tumbled down by a man and from another man being hurt. But if we suppose that normally the patient lives alone in the house then the “fall” interpretation is the most appropriate. Two other concepts of hurt and sick/ill are listed in Table 1. It is noted that water_in_toilet includes sound of water discharge, of water flowing from shower, water rushing or dripping from faucet, etc. In short, the concept of sound can be identified based on its attached concept properties and their respective facets. The usage of these concepts will be presented in next paragraph.

3.2 Ontology-Based Sound Classification

Ontology-based classifications are mostly applied in text, image and biological applications. The image classification system presented by Breen et al. in [2] uses a neural network to identify objects in a sports image. Category of the image is determined by a combination of detected image objects, each objects being assigned an experimental weight. Image ontology in this case is used to prune the selection of concepts: if the parent and the children are selected, the later will be discarded. In order to automatically classify web page, Prabowo *et al* in [23] apply a feed-forward neural network represent relationships in ontology. The output of this network is used to estimate similarity between web pages. Mezaris *et al* in [18] propose object ontology for an object-based image retrieval system. In this ontology, each immediate-level descriptors is mapped to an appropriate range of values of the corresponding low-level arithmetic feature. Based on low-level features and query keywords, a support vector machine will result the final query output. In [21], Noh et al. classify web pages using ontology. In this ontology, each class is predefined by a certain keyword and their relations. Classes of web pages are classified by extracting term frequency, document frequency and information gain, and by using several machine learning algorithms.

Taghva *et al* in [25] construct an email classification system in which ontology is applied to extract useful feature, these feature are inputs of a Bayesian classifier. The text categorization system of Wu *et al* in [27] also employ an ontology predefined by keywords and semantic relationship of word pair. These keywords are chosen by a term frequency / inverse document frequency classifier. The domain of a text is categorized by a “score”. Maillot *et al.* in [17] introduce their ontology-based application for image retrieval. Each image object in an image is detected by a trained detector. The relations of detected image object, established in ontology, will determine the category. In [26], Wallace *et al.* present their multimedia content indexing and retrieval system. The ontology (basing on MPEG-7 description schemes) of the system includes semantic entities, semantic relations and a thesaurus. A text-mining-based ontology enhancement and query-processing system is presented by Dey and Abulaish [6]. Their key ideas are to learn and to include imprecise concept descriptions into ontology structures. Instead of using “very sweet” and “intensely sweet” in the wine ontology, they assign “sweet” for both wines, “very” for the first and “intensely” for the second. The degree of similarity between “very” and “intensely” is computed by a fuzzy reasoner. In [15], Laegreid *et al.* present a system for biological process classifications using Gene Ontology (GO, stored at Gene Ontology Home page, [7]). Their goal is to model the relationships between gene expression and a biological process, to learn and classify multiple biological process roles, and to use this model to predict the biological participation of unknown genes. Their method uses biological knowledge expressed by GO and then generates the model. In work of Robinson *et al.* [24], results of cluster analysis of gene expression microarray data is based on GO terms and associations.

In ontology-based image applications, an image object is often defined by lower properties, such as form, color, viewing angle, background, etc. The method of defining related lower properties in image applications is hard to apply to audio domain, because an audio object, such as a laugh, is hard to be defined.

Text applications use relationships between text objects to classify. They are often known or predefined. For example, in a text application “net” can be interpreted as “a fishing tool” if words such as “fish”, “boat”, “river” are found in the same sentence of paragraph because they have obvious relation; or it can be a “group of connected computers” if there are “port”, “cable”, “TCP/IP”, “wi-fi” nearby.

GO is usually applied as a basis of biological applications. It provides a structured and controlled vocabulary describing genes and gene products in organisms. This ontology consists of biological objectives, molecular functions and cellular components, organized into hierarchies. To our knowledge, audio applications can take no advantage of this type of ontology.

In our work methods used in text applications are considered because we hope to find the meaning of an audio scene by a group of sounds. The predefined classifying rules of Noh *et al.* is hard to apply to audio domain because so far we do not know a predefined rule for sounds. The keyword-based ontology of Wu *et al* is also difficult to be used in our work because it needs relationship between sounds. The imprecise concept descriptions of Dey and Abulaish demand adjectives and adverbs. It cannot be applied to audio domain since audio signal has no equivalence. The method of using neural network to represent ontology of Prabowo *et al* seems to be the most appropriate for us since it demands only two levels of concept and properties.

Therefore in our work the relationship among concept, concept properties and facets is represented by a feed forward neural network described in [23]. The task of this network is to produce an output value that is used to estimate the similarity between the ontology and a new sound situation.

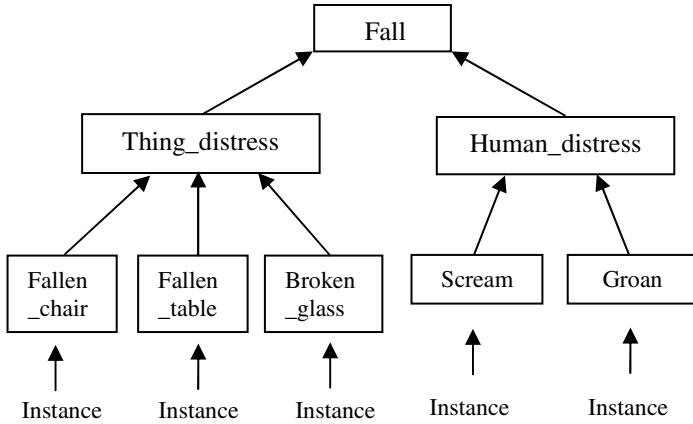


Fig. 3. The feed-forward neural network representing the “fall” ontology. Weights of links between layers depend on number of properties, number of facets and number of detected instances.

This model has three layers: input, hidden and output. The input layer of the network represents a set of instances. The hidden layer represents a set of concept properties. The number of neurons in the hidden layer equals the number of concept properties. The output layer (consists of one neuron) is the concept representatives. The two reasons to choose sigmoid transfer function,

$$f(x) = 1/(1 + e^{-x}) \quad (1)$$

for neurons, are as follows. Different concepts have different numbers of properties, so their outputs are normalized form 0 to 1. And they vary continuously. The neural network representing the “fall” concept is depicted in Fig. 3. In this example there are two hidden neurons and five input neurons. Two properties “thing_distress” and “human_distress” are represented by two hidden neurons. Input neurons are used to model the five instances of the concept. The two other neural network representing “hurt” and “sick/ill” have the same number of layers. “Hurt” has one hidden neuron and two input neurons, “sick/ill” has two hidden neurons and four input neurons.

The weights of links between the output and hidden layer depend on the number of hidden neurons. The weights of links between a hidden neuron also depend on the number of its attached input neurons. Details of calculation of those weights can be found in [23]. If n instances are matched, the weight of each instance is the square root of n .

During the classification phase, similarity between an input sample and a concept is calculated in order to assign a concept to the sample. Classes of sounds are

extracted. If an instance is found, its respective concept properties input is set to 1, otherwise it is zero. According to detected instances and their classes, weights of links of layers are determined. The output of each neural network therefore is a function of weighted properties of the respective concept and is the similarity between the sample and the concept. The input sample will be assigned to the concept with which it has the highest similarity.

4 Experimental Evaluation

An audio scene database of situations of distress is difficult to build. Firstly, such a database can be collected in hospitals, but recording this type of audio scenes in hospitals is nearly impossible due to the concerns of privacy. Secondly, recording them in a studio is feasible, but situations of distress are hard to be simulated by speakers, making a not true corpus. And finally in fact audio scenes are so numerous that it is hard to build a database (from a sound effect database) that can cover many situations. Therefore we collect manually audio scenes from films. The scenes we target are the ones in which the character is in house and in a situation of distress: falling down, being hurt, sick or ill. From 150 films, 44 scenes of situations of distress with total duration of 680 seconds are collected.

This classifier (illustrated in Fig. 4) should function automatically: detects sounds, estimates similarities between the sample and ontology, and outputs the concept of the scene. But in this first stage of building an ontology-based classification system, the first steps are manually carried out. 44 audio scenes of situations of distress and 50 of normal (non distress) situations are tested. Results of the ontology-based classification are presented in Table 2.

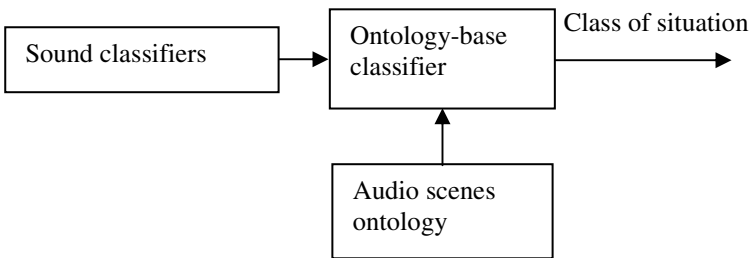


Fig. 4. The ontology-based classifier. Detected sounds are classified by the sound classifiers. Its output is fed to the audio scene classifier.

The misidentified scene of “fall” is the one in which the character falls down unconsciously making the door shut, and the unique sound we get is a shutting door. It seems to need a redefinition for this concept. But if we add this type of sound into the ontology, the system will identify actions of shutting doors as “fall”, and that will lead to wrong classification results. Therefore the fall ontology does not need to be redefined. There is one normal situation that is wrongly identified as a “hurt” one. In this

situation the character cries when she is too happy. So arises the need of an algorithm (should be developed in the future) that is capable of discriminating between the cries coming from happiness and those coming from hurt. Based on results of classification, it can be seen that the ontologies are appropriate for audio scenes of situations of distress.

Table 2. Results of ontology-based classification. Number of correctly identified is the number of audio scenes of situations of distress which are correctly identified as its assigned concept in the database. Number of wrongly identified is the number of audio scenes of normal situations which are identified as a situation of distress.

	Fall	Hurt	Sick/ill
Number of scene	20	19	5
Number of correctly identified	19	19	5
Number of wrongly identified	0	1	0

5 Conclusion

We present in this article an ontology-based audio scene classifier which can be used to detect audio scenes of situations of distress. The construction of the ontology is within the framework of a telemedicine project. Three ontologies of sounds are defined. Concept, properties and instances of an ontology are modeled by a feed forward neural network. The output of a neural network presenting a concept is the similarity between it and the input sample. At first stages of developing the classifier, we defined three domain ontologies and tested them manually. These ontologies work well with our first audio database of situations of distress which is extracted from scenes of films.

In the future a fully automatic ontology-based system needs to be built. In order to achieve this, the following tasks need to be undertaken. First, more sound classes should be classified to cover a larger range of different types of sound. Second, sounds need to be separated from music because audio scenes collected from film are often mixed with music, making sound classifiers work inexactly. Third a combination of ontologies and sound classifier should be built. Fourth more situations of distress need to be defined. And finally, a bigger database should be acquired to obtain more complete domain ontologies. Besides the extension of this audio database, we also think of acquiring a text database from the Internet. This text database will consist of paragraphs that use types of sound in order to describe audio scenes in house. In short it is a text database of audio scenes. Audio object classes, context of the scene, or distribution of audio object can probably be extracted from this database.

Acknowledgement

The authors gratefully acknowledge the receipt of a grant from the Flemish Interuniversity Council for University Development cooperation (VLIR UOS) which enabled them to carry out this work.

The authors also gratefully acknowledge the receipt of grants from French RESIDE-HIS project and French MAE/CORUS program.

References

1. Amatriain, X., Herrera, P.: Transmitting Audio Contents as Sound Objects. Proceedings of AES 22th International Conference on Virtual, Synthetic and Entertainment Audio, Espoo Finland (2002)
2. Breen, C., Khan, L., Kumar, A., Wang, L.: Ontology-Based Image Classification Using Neural Networks. Proc. SPIE (the International Society for Optical Engineering) Vol. 4862 (2002) 198-208
3. Cano, P., Koppenberger, M., Celma, O., Herrera, P., Tarasov, V.: Sound Effects Taxonomy Management in Production Environments. Proceedings of AES 25th International Conference, London UK (2004)
4. Carey, M.J., Parris, E.S., Lloyd-Thomas, H.: A Comparison of Features for Speech, Music Discrimination. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Munich Germany (1997)
5. Casey, M.: MPEG-7 Sound-Recognition Tools. IEEE Transaction on Circuits and Systems for Video Technology, Vol. 11, No. 6 (2001)
6. Dey, L., Abulaish, M.: Ontology Enhancement for Including Newly Acquired Knowledge about Concept Descriptions and Answering Imprecise Queries. Web Semantics Ontology, Idea Group (2006) 189 - 225
7. Gene Ontology Home: <http://www.geneontology.org/>
8. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2) (1993) 199-220
9. Hatala, M., Kalantari, L., Wakkary, R., Newby, K. : Ontology and Rule Based Retrieval of Sound Objects in Augmented Audio Reality System for Museum Visitors. Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia Cyprus (2004) 1045 – 1050
10. Istrate, D., Vacher, M., Castelli, E., Sérignat, J.F.: Distress Situation Identification through Sound Processing. An Application to Medical Telemonitoring. European Conference on Computational Biology, Paris (2003)
11. Istrate, D., Vacher, M., Sérignat, J.F., Besacier, J.F., Castelli, E.: Système de Télésurveillance Sonore pour la Détection de Situation de Détresse (Sound Telesurveillance System for Distress Situations Detection). ITBM-RBM Elsevier (2006)
12. Istrate, D., Castelli, E., Vacher, M., Besacier, L., Sérignat, J.F.: Sound Detection and Recognition in Medical Telemonitoring. IEEE Transactions on Information Technology in Biomedicine (to be published)
13. Khan, L., McLeod, D.: Audio Structuring and Personalized Retrieval Using Ontologies. Proceedings of IEEE Advances in Digital Libraries, Library of Congress, Washington DC (2000) 116-126
14. Kim, H.G., Moreau, N., Sikora, T.: Audio Classification Based on MPEG-7 Spectral Basis Representations. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 5 (2004)
15. Laegreid, A., Hvidsten, T.R., Midelfart, H., Komorowski, J., Sandvik, A.K.: Predicting Gene Ontology Biological Process from Temporal Gene Expression Patterns. Genome Res 13 (2003) 965-979
16. Lu, L., Jiang, H., Zhang, H.J.: A Robust Audio Classification and Segmentation Method. ACM Multimedia (2001) 203-211

17. Maillot, N., Thonnat, M., Hudelot, C.: Ontology Based Object Learning and Recognition: Application to Image Retrieval. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton FL USA (2004)
18. Mezaris, V., Kompatsiaris, I., Srintzis, M.G.: An Ontology Approach to Object-Based Image Retrieval. In proceedings of the IEEE International Conference on Image Processing (2003)
19. Nakatani, T., Okuno, H.G.: Sound Ontology for Computational Auditory Scene Analysis. Proceedings of the fifteenth National Conference on Artificial Intelligence (AAAI-98), Vol.-1 (1998) 30-35
20. Nguyen, C.P., Pham, N.Y., Castelli, E.: Toward a Sound Analysis System for Telemedicine. 2nd International Conference on Fuzzy Systems and Knowledge Discovery, Chansha China (2005)
21. Noh, S., Seo, H., Choi, J., Choi, K., Jung, G.: Classifying Web Pages Using Adaptive Ontology. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Washington D.C. (2003) 2144-2149
22. Pfeiffer, S., Fischer, S., Effelsberg, W.: Automatic Audio Content Analysis. IEEE Multimedia, 3(3) (1996) 27-36
23. Prabowo, R., Jackson, M., Burden, P., Knoell, H.D.: Ontology-Based Automatic Classification for the Web Pages: Design, Implimentation and Evaluation. The 3rd International Conference on Web Information Systems Engineering, Singapore (2002)
24. Robinson, P.N., Wollstein, A., Bohme, U., Beattie, B.: Ontologizing Gene-Expression Microarray Data: Characterizing Clusters with Gene Ontology. Bioinformatics 20(6) (2004) 979-981
25. Taghva, K., Borsack, J., Coombs, J., Condit, A., Lumos, S., Nartker, T: Ontology-Based Classification of Email. International Conference on Information Technology: Computers and Communications, Las Vegas Nevada (2003)
26. Wallace, M., Avrithis, Y., Stamou, G., Kollias, S.: Knowledge-Based Multimedia Content Indexing and Retrieval. Multimedia Content and the Semantic Web – Methods, Standards and Tools, Wiley (2005) 299-338
27. Wu, S.H., Tsai, T.H., Hsu, W.L.: Text Categorization Using Automatically Acquired Domain Ontology. The Sixth International Workshop on Information Retrieval with Asian Languages, Sapporo Japan (2003) 138-145
28. Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I.: Semi-supervised Adapted HMMs for Unusual Event Detection. IEEE Conference on Computer Vision and Pattern Recognition, San Diego USA (2005)

Obstacles and Misunderstandings Facing Medical Data Mining

Ashkan Sami

Department of Computer Science and Engineering
Shiraz University, Shiraz, Iran
sami@cse.shirazu.ac.ir

Abstract. Medical Data Mining is a very active and challenging research area in Data Mining community. However researchers entering Medical Data Mining should be aware that in core clinical, dentistry and nursing, data mining is not welcomed as much as we believe and publication of results in these journals based on Data Mining algorithms is not easily possible. In this paper, in addition to presenting one of our “successful” KDD projects in Urology that did not get to anywhere, we back up our belief based on designed searches on PubMed and review literature based on these searches. Our findings suggest that few Data Mining algorithms made their ways into core clinical journals. The paper concludes by reasons we have collected through our experiences.

1 Introduction

Data Mining algorithms have found their place as one of the prime methodologies to extract knowledge from data in variety of fields. New methodologies and algorithms are greatly welcomed in majority of scientific and engineering domains. Just as an illustration, when the first complete solution to Graph-Data mining, AGM [1] was introduced in 2000, the result of this research was published in a chemical journal in 2001 [2]. Anyone in Data mining community is aware that Knowledge Discovery from Database (KDD) especially Data Mining (DM) has affected almost all scientific disciplines in one way or another.

Due to uniqueness and challenging nature of Medical data mining [3], a lot of researchers in KDD community have been attracted to Medical data mining [4-6]. Moreover, knowledge discovery in medical data is considered very rewarding. What kind of knowledge can be more important than adding new insights to medical knowledge that will affect the wellbeing of all humans?

Medical Data Mining is a very active and challenging research area in Data Mining community. The driving force of Genomics, microarray analysis and protein research is DM. In some other areas like cancer research, pharmaceutical preparations and Pharmacovigilance it has had major contributions. Even some medical doctors in core clinical fields by literature review of KDD in medicine recommend the need to deploy the methodologies. As an example, Lucas writes, “With the increasing availability of biomedical and health-care data with a wide range of characteristics there is an increasing need to use methods which allow modeling the uncertainties that come with the problem, are capable of dealing with missing data, allow integrating data from various sources, explicitly indicate statistical dependence and independence, and

allow integrating biomedical and clinical background knowledge. These requirements have given rise to an influx of new methods into the field of data analysis in health care, in particular from the fields of machine learning and probabilistic graphical models.”

In addition, search of PubMed¹ which is a well-known medical literature indexing with a fairly restrictive query shows an increasing trend. This fact might lead us to believe that medical community is very rapidly accepting KDD and DM. The search is performed through Entrez [8], which is the integrated, text-based search and retrieval system used at NCBI. The query, presented in Table 1, searches for English journal publications indexed by PubMed that are not literature review or congresses and have ‘data mining’ in their title or abstract or are related to ‘knowledge discovery in database’. The graphical presentation of number of journal publications in English based on the same query for years from 1997 to 2005 is illustrated in Figure 1.

Table 1. The query used to find DM & KDD journal publications indexed in PubMed in 2003

("data mining"[TIAB] OR "data-mining"[TIAB] OR "datamining"[TIAB]) OR ("Knowledge discovery"[All Fields] AND ("databases"[TIAB] OR "databases"[MeSH Terms] OR database[Text Word])) AND (Journal Article[pt] NOT review[pt] NOT Congresses[pt] AND English [la]) AND ("2003/01/01"[PDAT]:"2003/12/31"[PDAT])

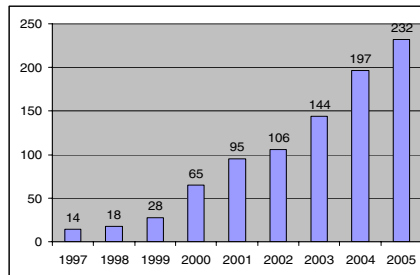


Fig. 1. Number of journal publications in PubMed related to KDD or DM per year

New technologies in health care also provide vast amount of data about patients that are well beyond the capabilities of humans to manually extract knowledge. Use of measuring devices that are capable of storing and transferring the measurement results, inexpensive hard drives, and common use of computers to store the medical data and increase reliance on digital imaging technology are sample examples that medicine is indeed in an era that must adopt the new knowledge extracting methodologies for diagnosis, prognosis and critical care.

In contrast to all said above, data miners entering any area of nursing, dentistry, or core clinical care, like Urology, Gynecology, Anesthesiology, Pediatrics, Epidemiology, Psychology, Surgery and etc and would like to have publications in those fields based on KDD or DM methods in purely medical journals should be fully

¹ PubMed is a service of the U.S. National Library of Medicine that includes over 16 million citations from MEDLINE and other life science journals for biomedical articles back to the 1950s.

aware that in these fields the story is completely different. There are “successful” data mining projects that literally get to nowhere. In this paper, we try to create a clearer understanding of “real” status of DM and KDD in core medical fields’ journals.

The organization of this paper is as follows. First, we present our “successful” KDD project in Urology that did not get to publication. In section 3, some searches for DM and KDD publication in Nursing, Dentistry and Core Clinical Journals and a review of the algorithms used is presented. In section 4, we take a closer look at PubMed by observations from MeSH Database. Section 5 presents the reasons. Finally section 6 concludes the paper by a summary.

2 A “Successful” KDD Project in Urology That Got Nowhere

There are two type of urological problem: storage (or obstructive) and voiding symptoms. Urologists are very interested to find out which category of the symptoms is more prominent.

A complete list of Storage symptoms are: Day-time frequency of urination (times/day), Night-time frequency of urination, Rush to the toilet to urinate, Urine loss before reaching the toilet, Bladder pain, Urine leaking when coughing/sneezing, Urine leakage, Urine loss while asleep, Urinate again (within 15 minutes), Day-time frequency of urination (time/h).

Voiding symptoms are: Delay before starting to urinate, Strain to start urinating, Strain to continue urinating, Reduced urinary stream, Stop and start urinating more than once, Burning when urinating, Incomplete bladder emptying, Terminal dribble, Slight wetting of pants after urination.

Data concerning these symptoms came from Tsurugaya Project distribution CD, Tohoku University Medical School, Sendai, Japan, were data was collected through standard questionnaires especially designed for other purposes with multiple choice answers as the main methodology to collect status of symptoms.

Since the main concern of Urologists was to find the more prominent category with respect to Quality of Life (QOL), we used decision trees for the evaluation. The symptoms were considered as antecedents (inputs) to the decision tree and QOL as the consequent (output). 10-fold cross validation was used for all the decision trees and weights of all the antecedents were considered equal.

C 5.0, which is proprietary software, was our main choice for the analysis. Decision trees were constructed to find the most important contributors. In decision trees for all and male volunteers Incomplete Bladder Emptying, which is a voiding factor, was the first node. We performed these tests to gain some understanding and see which symptom has the most discriminating power.

In the second attempt, we used a more straight forward method by separating obstructive and voiding symptoms and made decision trees for each category for all the volunteers, male volunteers and female volunteers. We treated symptoms as numerical and in another trail as ordered sets. We found out that the accuracy of decision trees for voiding symptoms were slightly higher than storage symptoms. However since the difference was not very significant, we concluded that none of the categories could be considered more important with respect to the other.

After presenting the results to Urologist through various meetings and convincing them the results were accountable. However when the work was presented to an editor of Japanese Journal of Urology, he strongly objected the method by not being cited before in another Urological literature. Our literature review of Urological and other core medical journals did not reveal a common usage, so the paper was not published. The method used does not have much novelty, if any, so it is not appropriate for a computer science conference or journal either.

Since this story might sound like a failure story and cannot by itself reveal a general fact concerning usages of KDD or DM methods in Medicine, we present a closer look at medical journals publishing results based KDD or DM methodology. It is important to note that other researchers in completely different cultural and geographic setting had same concerns [9].

3 Searching DM and KDD in Core Clinical Journals

Even though the results of the search presented in the introduction might lead us to believe that there is a growing trend of accepting data mining algorithms in medical community, three reasons that will be presented shortly disqualify extending the results to purely clinical, nursing and dentistry journals.

First, large volumes of publication are related to “data-mining-welcomed” area of medicine like bioinformatics, especially microarray analysis, protein research and cancer. Also vast number of papers is related to drug discovery in which medical community embraced data mining algorithms very openly. It is no surprise to see a very novel data mining algorithm published in respected medical journals like *Nucleic Acids Research*.

Secondly, engineering and computer science journals that have publications related to life sciences are also indexed in PubMed. For example, journals like *Neural Networks*, some of the IEEE publications like *IEEE Transaction in Pattern Analysis and Machine Intelligence*, even chemical journals like *Analytical Chemistry* are all indexed in PubMed. Not to mention Journals like *Artificial Intelligence in Medicine*, that are mainly targeted at researchers in our community.

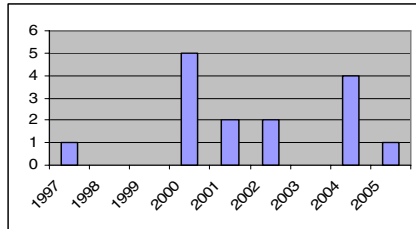
In addition, PubMed indexes all life science articles. In other words, journals and conferences in animal studies to environment are all indexed by PubMed. However searching core clinical journals like, Urology, Gynecology, Anesthesiology, Pediatrics and etc, reveals another story.

Searching for any journal article in any language that has “data mining” or “knowledge discovery” anywhere in the paper in nursing, dentistry and core clinical journals related to humans will revealed disappointing results, only 15 publications from 1997 to the end of 2005 [10-24]. Considering large quantity of journals and fields involved in clinical studies, this query is a good indication that DM is by far much less popular in core clinical practices than it is believed by our community. The distribution is illustrated in Figure 2 and the query used is depicted in Table 2.

Even though 15 journal articles among all journals of Dentistry, Nursing, Urology, Epidemiology, Surgery, Pediatrics and etc were found, review of these journals unfortunately may result in more disappointments.

Table 2. The query used to find journal publications in core clinical, nursery and dentistry

```
("data mining"[All Fields] OR "data-mining"[All Fields] OR "datamining"[All Fields]) OR
"Knowledge discovery"[All Fields] AND ("Journal Article"[Publication Type] NOT
"Review Literature"[MeSH]) AND jsubsetaim[text] AND "humans"[MeSH Terms] AND
("1997/01/01"[PDAT] : "2005/12/31"[PDAT])
```

**Fig. 2.** DM and KDD Journal papers in core clinical, nursery and dentistry

Majority of articles were mainly dealing with Gene research applied to clinical studies for example [15][18][19][23]. Some [12][14][20][22] were using decision tree mainly CART and calling it DM. [20] was the only that did not use CART, but the results were validated by statistical methods. In [22] neural networks and Genetic Algorithm were also deployed. Rough sets were used in [10]. We saw use of Neural Network mainly Bayesian Neural Network in [16] and [21]. However they cited previous publications in medicine using the same method, [16] cited [25]. [24] was the only journal in nursing. However, we could not find a single paper that is using a novel data mining algorithm especially designed for the medical data used. Moreover, none of the journals were related to dentistry.

Regardless of the fact that no common consensus on clear definition of data mining exists, the well known Apriori and its variations was not observed in any of these literatures.

As a side note, an interesting observation worth mentioning, Harris in Archives of Internal Medicine [26] is the first one to use the term “data mining” in a clinical journal back in 1984. However the term “data mining” in the abstract is used as an alternative term for data exploration by sequential multiple logistic regression.

4 Observations in MeSH Database

MeSH stands for Medical Subject Heading. MeSH terminology is used to assign topic headings to every article that is entered into Medline. One can search MeSH terms in PubMed through the MeSH Database. It is possible to retrieve articles by topic (according to what the article is actually about) instead of searching with keywords and hoping that the correct term has been chosen.

A keyword search is likely to retrieve a high number of irrelevant citations, and miss a great many very useful ones. However, it is appropriate to search PubMed using keywords if the term does not appear in the MeSH Database, or if the term is very new or highly specific.

Search of MeSH Database reveals that for 'decision tree' and 'cluster analysis' MeSH term has been introduced. This can be an implication that decision tree and cluster analysis have gain acceptance by Medical community. Searches on PubMed, Table 3, for core clinical, dentistry and nursing journals based on 'decision tree' and 'cluster analysis' MeSH terms reveal these two classes of algorithms have fairly gained acceptance even in these journals. Figure 3 shows the distribution of publications from 1990 up to 2005.

Table 3. The query used to find journal publications that used Decision Tree or Cluster Analysis in core clinical, nursery and dentistry in 1992

```
("Cluster Analysis"[MeSH] OR "Decision Trees"[MeSH]) AND ("Journal Article"[Publication Type] NOT "Review Literature"[MeSH]) AND jsubsetaim[text] AND "humans"[MeSH Terms] AND ("1992/01/01"[PDAT] : "1992/12/31"[PDAT])
```

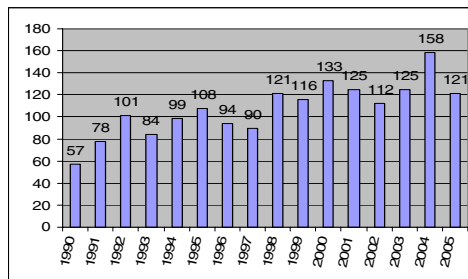


Fig. 3. Articles indexed in PubMed that used DT or Cluster Analysis in core clinical journals

Even though searches for journal publications in core clinical, nursery and dentistry that used Decision Tree or Cluster Analysis revealed acceptance of these classes of algorithms, the trend has not changed drastically after flourish of Data Mining in 1990's. Then again, majority of cluster analysis papers were mainly dealing with Genomic or protein research applied to clinical care.

Due to nature of clinical care, practitioners and researchers are fairly conservative and do not easily adopt new algorithms. For example, decision tree in 1988 entered MeSH Database, which is a couple of years after major breakthroughs in decision tree analysis. 'Data Mining' is not adopted as a MeSH term (as the data of this paper), even though its emergence dates back more than a decade.

5 Discussion

Physicians do not consider small patterns that are highly regarded in data mining. In addition, they believe that knowledge discovery is based on hypothesis which medical researchers set up through his/her experience and the prevailing or urged problem with a high demand for a solution. Even the selection of most data sets or parameters are based on such hypothesis. They believe it is the study design and material and not the analysis method that leads to knowledge. Statistical tools are just used to verify the assumption and method.

Thus the mindset of physicians is very different than researchers in DM or KDD who believe in data-driven knowledge extraction methodologies. One reason in fields like Genomics, data mining has boomed is that these kind of assumptions cannot be made by human experts.

Data driven methods that will be adopted for medicine should provide statistical significance or will not grab the attention of those in medical community. As an illustration, Mundt et. al [20] used a “new” decision tree in Psychology literature, but the journal accepts his publication since the outputs are validated statistically. Regardless of the fact he had a completely wrong understanding of QUEST decision tree by introducing it as “an optimum binary decision tree” in his paper [20].

Another aspect of resistance to DM and KDD by medical community might be traced to discouraging words from Statisticians who perform Bio-data analysis [9].

6 Summary

We presented a clearer status of DM and KDD in a specific health science namely, clinical care, nursing and dentistry journals. We briefly showed that Data Mining is gaining popularity in health sciences but has not gained its proper status in the journals where the main audiences are clinical medical practitioners and researchers. Mainly, we would like to warn researchers in DM and KDD that publications in purely medical journals faces much more challenge than envisioned. We presented some of the reasons for this dilemma and hope to see the day that barriers between the two societies vanish.

References

1. Inokuchi A., Washio T., Motoda H., “An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data”, Principles of Data Mining and Knowledge Discovery (PKDD 2000), 13-23, LNAI 1910, Springer-Verlag, (2000).
2. Inokuchi, A., Washio, T., Okada, T. and Motoda, H. “Applying the a priori-based graph mining method to mutagenesis data analysis.” J. Comput. Aided Chem., 2:87-92, (2001).
3. Krzysztof J. Cios, and G. William Moore, “Uniqueness of medical data mining”, Artificial Intelligence in Medicine 26(1-2), 1-24. (2002)
4. Damien McAullay, Graham Williams, Jie Chen, Huidong Jin, Hongxing He, Ross Sparks and Chris Kelman, “A Delivery Framework for Health Data Mining and Analytics”, the 28th Australasian Computer Science Conference, The University of Newcastle, Australia. Conferences in Research and Practice in Information Technology, Vol. 38. (2005)
5. Lavrac, N., “Selected techniques for data mining in medicine.” Artificial Intelligence in Medicine, 16:3-23. (1999)
6. Sakamoto N. “Object-oriented development of a concept learning system for time-centered clinical data.” J Med Syst. Aug;20(4):183-96, (1996).
7. Lucas, P.: Bayesian analysis, pattern analysis, and data mining in health care. Curr. Opin. Crit. Care. 10(5) 399-403, (2004)
8. <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed> (Last visited April 15, 2006).
9. Zaiane, O, Personal communications of first author through email during February and March, (2006)

10. Ohrn A, Rowland T. "Rough sets: a knowledge discovery technique for multifactorial medical outcomes." *Am J Phys Med Rehabil.* Jan-Feb;79(1):100-8, (2000).
11. Aoki N, Wall MJ, Demсар J, et al "Predictive model for survival at the conclusion of a damage control laparotomy." *Am J Surg.* Dec;180(6):540-4; discussion 544-5, (2000).
12. Strum DP, Sampson AR, May JH, Vargas LG. "Surgeon and type of anesthesia predict variability in surgical procedure times." *Anesthesiology.* May;92(5):1454-66, (2000).
13. Elevitch FR, Silvers A, Sahl JD. "Projecting corporate health plan utilization and charges from annual ICD-9-CM diagnostic rates: a value-added opportunity for pathologists." *Arch Pathol Lab Med.* Nov;121(11):1187-91, (1997).
14. Rider LG, Giannini EH, et al "International consensus on preliminary definitions of improvement in adult and juvenile myositis." *Arthritis Rheum.* Jul;50(7):2281-90, (2004).
15. Zeggini E, Thomson W, Kwiatkowski D, Richardson A, Ollier W, Donn R; "Linkage and association studies of single-nucleotide polymorphism-tagged tumor necrosis factor haplotypes in juvenile oligoarthritis." *Arthritis Rheum.* Dec;46(12):3304-11, (2002).
16. Coulter DM, Bate A, Meyboom RH, Lindquist M, Edwards IR. "Antipsychotic drugs and heart muscle disorder in international pharmacovigilance: data mining study." *BMJ.* May 19;322(7296):1207-9, (2001).
17. Zoutman DE, Ford BD, Bassili AR. "A call for the regulation of prescription data mining." *CMAJ.* Oct 31;163(9):1146-8, (2000).
18. Langmann T, Moehle C, Mauerer R, Scharl M, Liebisch G, Zahn A, Stremmel W, Schmitz G. "Loss of detoxification in inflammatory bowel disease: dysregulation of pregnane X receptor target genes." *Gastroenterology.* Jul;127(1):26-40, (2004).
19. Viguerie N, Clement K, et al "In vivo epinephrine-mediated regulation of gene expression in human skeletal muscle." *J Clin Endocrinol Metab.* May;89(5):2000-14, (2004).
20. Mundt JC, Freed DM, Greist JH., "Lay person-based screening for early detection of Alzheimer's disease: development and validation of an instrument." *J Gerontol B Psychol Sci Soc Sci.* May;55(3):P163-70, (2000).
21. Sanz EJ, De-las-Cuevas C, Kiuru A, Bate A, Edwards R. "Selective serotonin reuptake inhibitors in pregnant women and neonatal withdrawal syndrome: a database analysis." *Lancet.* Feb 5-11;365(9458):482-7, (2005).
22. Papadopoulos MC, Abel PM, Agranoff D, et. al., "A novel and accurate diagnostic test for human African trypanosomiasis." *Lancet.* Apr 24;363(9418):1358-63, (2004).
23. Ostermeier GC, Dix DJ, Miller D, Khatri P, Krawetz SA., "Spermatozoal RNA profiles of normal fertile men." *Lancet.* Sep 7;360(9335):772-7, (2002).
24. Goodwin LK, Iannacchione MA, Hammond WE, Crockett P, Maher S, Schlitz K., "Data mining methods find demographic predictors of preterm birth." *Nurs Res.* Nov-Dec;50(6):340-5. (2001).
25. Bate A, Lindquist M, Edwards IR, Olsson S, Orre R, Lansner A, et al. "A Bayesian neural network method for adverse drug reaction signal generation." *Eur J Clin Pharmacol;* 54:315-21 (1998).
26. Harris JM Jr. "Coronary angiography and its complications. The search for risk factors." *Arch Intern Med.* Feb;144(2):337-41 (1984)

SVM-Based Tumor Classification with Gene Expression Data

Shulin Wang^{1,2}, Ji Wang¹, Huowang Chen¹, and Boyun Zhang¹

¹School of Computer Science, National University of Defense Technology,
Changsha, Hunan 410073, People's Republic of China
jt_slwang@hnu.cn

²College of Computer and Communication, Hunan University,
Changsha, Hunan 410082, People's Republic of China

Abstract. Gene expression data that are gathered from tissue samples are expected to significantly help the development of efficient tumor diagnosis and classification platforms. Since DNA microarray experiments provide us with huge amount of gene expression data and only a few of genes are related to tumor, gene selection algorithms should be emphatically explored to extract those informative genes related tumor from gene expression data. So we propose a novel feature selection approach to further improve the SVM-based classification performance of gene expression data, which projects high dimensional data onto lower dimensional feature space. We examine a set of gene expression data that include sets of tumor and normal clinical samples by means of SVMs classifier. Experiments show that SVM has a superior performance in classification of gene expression data as long as the selected features can represent the principal components of all gene expression samples.

1 Introduction

The advent of DNA microarray technology provides biologists with the ability to measure the expression level of thousands of genes in a single experiment. With the development of this technology, a large quantity of gene expression data has been accumulating quickly, so a novel means should be explored to extract its biological functions that are unknown to us and to gather information from tissue samples regarding gene expression differences that will be useful in diagnosing disease.

In recent years, support vector machines [1][8], a supervised machine learning technique, have been shown to perform well in multiple areas of biological analysis including evaluating gene expression data, detecting remote protein homologies, and translation initiation sites, etc. Since DNA microarray data can be very high dimensional and have very few training datasets, this situation is particularly well suited for a SVMs approach. In this paper, our efforts are mainly to extract feature information from colon cancer samples and to apply SVMs to classify the extracted feature to help doctor to diagnose, treat, and predict tumor.

2 Related Works

A great deal of research has been done in the classification of gene expression data, the discovery of gene function, gene regulation network by utilizing unsupervised methods such as clustering [5] and self-organizing maps. While clustering the row or column vectors of gene expression data matrix, little prior biology knowledge is adopted, and we even don't know the biological meaning of the clustering results. In recent years, supervised methods such as decision trees, SVMs and multi-layer perceptrons have been broadly applied in order to classify normal and tumor tissues [2][3]. Sung-Bae Cho et al [10] have summarized the performance of many machine learning methods on tumor classification based on gene expression data.

However, feature selection plays a key role in the classification of gene expression data, for there are many noises and redundancies in gene expression data. Li et al [9] propose a GA/KNN method which is capable of selecting a subset of predictive genes from a large noisy dataset for sample classification. Furlanello et al [6] design a wrapper algorithm for fast feature ranking in classification problems which is an entropy-based recursive feature elimination method that can eliminate chunks of uninteresting features according to the entropy of the weights distribution of SVMs classifier. Kunihiro Nishimura et al [12] present a PCA-based method of gene expression which is a visual analysis with calculating PCA contribution axis. One drawback of PCA is, however, that class information is not utilized for class prediction. In fact, it is hard for single method to further improve the performance of tumor classification. Therefore, we propose an integrated feature extraction method which combines gene ranking method according to its classification ability with PCA to attempt to improve the classification performance of gene expression data using SVMs classifier.

3 The Classification Methods

3.1 Representation of DNA Microarray Data

Let $G = \{g_1, \dots, g_n\}$ be a set of genes and $S = \{s_1, \dots, s_m\}$ be a set of samples, where n is the number of gene set G , and m is the number of samples, and $s_i \in R^n, i = 1, 2, \dots, m$, and usually $n \gg m$. The corresponding gene expression matrix can be represented as $M = (x_{i,j})_{m \times n}$, where $x_{i,j}$ is the expression level of sample s_i on gene g_j . Hence each vector s_i may be thought of as a point in n -dimensional space, and each of n columns consists of an m -element vector for a single gene expression.

Our task is to classify all samples into tumor samples and normal samples, which is a binary classification problem. Suppose ω_1 and ω_2 be the two subsets of sample set S , satisfying $\omega_1 \cap \omega_2 = \phi, \omega_1 \cup \omega_2 = S$, which means that each sample belongs to one and only one class ω_1 or ω_2 .

3.2 Support Vector Machines

SVMs are a relatively new type of statistic learning theory, originally introduced by Vapnik and successively extended by a number of other researchers. The advantage of SVMs is that its general capability can be improved by using structural risk minimization principle. In another words, we can get a relatively small error rate on independent testing set under the circumstances of utilizing limited training set.

In standard linear classification problem we are looking for a weight vector $w \in R^n$ and scalar bias b of a linear classifying function: $f(x) = w \cdot x + b$, which satisfies the following set of inequalities:
$$\begin{cases} w \cdot x_i + b > 0, & x_i \in \omega_1 \\ w \cdot x_i + b < 0, & x_i \in \omega_2 \end{cases}$$
 where $w \cdot x$ is a

inner product. When the training set is linearly separable, there exists such a function. For the simplicity let us introduce a set of desired outputs $\{y_i \mid y_i = \begin{cases} +1, & x_i \in \omega_1 \\ -1, & x_i \in \omega_2 \end{cases} \mid i = 1, 2, \dots, m\}$. Suppose we are given labeled training data-

set $S = \{(x_i, y_i) \mid (x_i, y_i) \in R^n \times \{\pm 1\}, i = 1, 2, \dots, m\}$. Here $x_i \in R^n, y_i \in \{\pm 1\}$ is a label of sample x_i . Suppose we have a hyper-plane $w \cdot x + b = 0$, which separates the positive samples from the negative samples. The optimal hyper-plane should be a function with margin between the vectors of the two classes, which subject to the constraint as:

$$\begin{aligned} \max L(\alpha) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i y_i \alpha_j y_j K(x_i, x_j), \\ \text{Subject to: } &\sum_{i=1}^m \alpha_i y_i = 0, \alpha_i \in [0, C], i = 1, \dots, m \end{aligned}$$

Where α_i is Lagrange multiplier which can be computed by training SVMs, $K(x_i, x_j)$ is a kernel function, and parameter C is a given constant which adjusts the error rate of classification. For a test sample x , we could use decision function $f(x) = \text{sgn}(\sum_{i=1}^m \alpha_i y_i K(x_i, x) + b)$ to determine which class it belongs to. For cases in

which no linear separation is possible, they can work in combination with the technique of kernels, which automatically realizes a non-linear mapping to a feature space. The hyper-plane found by the SVMs in feature space corresponds to a non-linear decision boundary in the input space. Let $\phi(x) : I \subseteq R^n \rightarrow F \subseteq R^h, (n < h)$ be a mapping from the

input space I to a feature space F . The decision function can equivalently be expressed as $f(x) = \text{sgn}(\sum_{i=1}^m \alpha_i y_i K(\phi(x_i), \phi(x)) + b)$.

3.3 Feature Selection

The expression level of most genes measured in datasets is irrelevant to the distinction between tumor and normal tissues. To precisely classify tumor we have to select genes, which is called informative genes, highly related to tumor for classification. Therefore, to reduce unnecessary noise to the classification process, informative genes selection is of great importance in the analysis of gene expression data. Our proposed feature selection method called hybrid method integrates PCA with Feature Score Criterion (FSC) that was used in [4] to drastically reduce the dimension of gene expression data and to minimize the information loss before using the SVMs algorithm. The novel hybrid method exploits the advantages that each approach offers. FSC is a calculated ranking number for each gene to define how well this gene discriminates two classes, and the principal components can be found by calculating the eigenvectors of the covariance matrix of the gene expression data. The central idea of PCA is to reduce the dimensionality of the dataset while retaining as much as possible the variation in this dataset. The feature selection algorithm is as follows.

Step1 For each gene g_i in G , we firstly calculate the mean μ_i^+ (resp. μ_i^-) and standard deviation σ_i^+ (resp. σ_i^-) which correspond to the gene g_i of samples labeled +1(-1), respectively. Then we calculate a feature score $F(g_i) = |(\mu_i^+ - \mu_i^-) / (\sigma_i^+ + \sigma_i^-)|$ for each $g_i \in G$, and rank the genes according to their score values. At last, we simply select the genes with the highest $F(g_i)$ scores as our top genes G_{top} , satisfying $|G_{top}| \ll |G|$. Step 2 Applying Principal Component Analysis(PCA) to the top ranking genes G_{top} to calculate new r variables as represents of G_{top} , satisfying $r \ll |G_{top}|$.

4 Experiments

4.1 Sample Dataset

We mainly study the samples of colon cancer dataset which involves comparing tumor and normal samples of the same tissue [11]. The dataset consists of 62 samples of colon epithelial cells including 40 colon cancer samples and 22 normal samples. Gene expression level in these 62 samples was measured using high density oligonucleotide microarray. Among the 6000 genes detected in these microarrays, 2000 genes were selected based on the confidence in the measured expression level. The dataset is

available at web site <http://www.molbio.princeton.edu/colondata>. Among the published datasets, classifying colon dataset is more difficult than doing others, so in fact the colon dataset may be seen as the benchmark dataset of various tumor classification methods.

4.2 Experiment Method

In practice, we use the software LIBSVM [7] to classify the colon dataset into normal and tumor class. Training SVMs requires specifying the type of kernel and the regularization parameter C . However, finding the best choices for the kernel and parameters can be challenging when applied to real datasets. Generally, the recommended kernel for nonlinear problems is the Gaussian radial basis kernel $K(x, y) = \exp(-\sigma\|x - y\|^2)$ [9], because it resembles the sigmoid kernel for certain parameters and it requires less parameters than a polynomial kernel. The kernel parameter σ and C , which controls the complexity of the discriminant function versus the training error minimization, can be determined by running a 2-dimensional grid search, which means that the values for pairs of parameters (C, σ) are generated in a predefined interval. Performance was tested by utilizing a cross-validated method. Accuracy of a diagnostic test can be expressed with recognition rate, and good classifier requires high recognition rate.

4.3 Results and Analysis

Initially, experiments are carried out only using FSC method to select the 150, 50 and 25 top-ranked genes as represents of 2000 genes respectively, and then on the basis of the selected genes we employ PCA to generate 3 principal components to be used as the input of SVMs classifier. Figure 1 plots the first two principal components to have a visual expression of the similarities among samples. In figure 1 the label number represents the serial number of each sample in original dataset and the line obviously divides samples into two classes in which only 4 samples are divided by error.

Table 1 shows the experiment results of two feature selection methods that have the highest accuracy. From the classification results of experiments, we can conclude that our hybrid method is obviously superior to the single FSC method in reducing dimension for SVMs classification

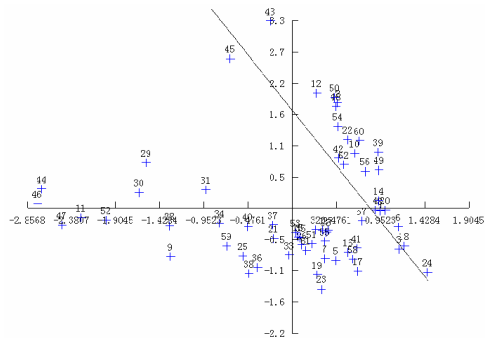


Fig. 1. 2-D scatter plot of the first two principal components

and in improving performance of SVMs classification when retaining high recognition rate. The cross validation accuracy of the hybrid method is approximately 90.32%. Comparing with other methods which was summarized in literature [10], our method is able not only to classify the dataset with higher recognition rate but also to visualize the classified results.

Table 1. Comparison of recognition rate with two methods

Method	#Genes	C	σ	Accuracy
FSC	150	1500	0.00008	95 %
	50	1500	0.00005	95 %
	25	500	0.0008	90 %
Hybrid	150	1500	0.03	95 %
	50	100	0.005	95 %
	25	1000	0.005	95 %

5 Conclusion and Future Work

SVMs is one of the prospective tools of analysis for gene expression data coming from DNA microarray. This is due to the fact that SVMs is particularly suitable to cope with small datasets, when the number of samples is much less than the number of genes. Experiments show that our hybrid method performs well in reducing dimension and improving the performance of SVMs classifiers. Another advantage of the hybrid method is that the classification results can be visualized in 2-D and 3-D. We will further focus on exploring how to autonomously select the parameters for SVMs classifier and developing the classification tool based on SVMs, which will integrate various feature selection methods, to help doctor to diagnose and predict cancer.

Acknowledgement

This research was funded by the National Natural Science Foundation of China under the grant No. 60233020 and Hunan Provincial Natural Science Foundation of China under the grant No. 04JJ6032. We also loyally thank the reviewers for their sound advice for this paper.

References

1. Vapnik V.N., Statistical learning theory. Springer, New York (1998).
2. Terrence S. Furey, Nello Cristianini, Nigel Duffy, David W. Bednarski, Michel Schummer, and David Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*. Vol. 16 No. 10 (2000) 906-914.
3. Guyon I.J. Weston S. Barnhill, and Vapnik V.. Gene selection for cancer classification using support vector machines. *Machine Learning* (2002) 46:389-422.

4. Golub T.R., Slonim D.K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., Bloomfield C.D., and Lander E.S.. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* (1999) 286:531-537.
5. Eisen M., Spellman P., and Botstein D.. Cluster analysis and display of genome-wide expression patterns. *PNAS*, (1998) 95:14863-14868.
6. Furlanello C., Serafini M., Merler S., and Jurman G.. An accelerated procedure for recursive feature ranking on microarray data. *Neural Networks* (2003) 16:641-648.
7. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines (2001), Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
8. Cristianini N. and Shawe-Taylor J.. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK (2000).
9. Li L., Weinberg C.R., Darden T.A., and Pedersen L.G.. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*(2001) 17(12):1131-1142.
10. Sung-Bae Cho and Hong-Hee Won. Machine learning in DNA microarray analysis for cancer classification. *Proceedings of the First Asia-Pacific Bioinformatics Conference on Bioinformatics* (2003) 189-198.
11. Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D., and Levine A., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues by oligonucleotide arrays. *Proc. Nat. Acad. Sci. USA*, (1999) 96:6745-6750.
12. Kunihiko Nishimura, Koji Abe, Shumpei Ishikawa, Shumpei Ishikawa, Shuichi Tsutsumi, Koichi Hirota, and Hiroyuki Aburatani. A PCA based method of gene expression visual analysis. *Genome Informatics* (2003) 14: 346-347.

GEPCLASS: A Classification Rule Discovery Tool Using Gene Expression Programming

Wagner R. Weinert and Heitor S. Lopes*

Bioinformatics Laboratory, CPGEI
Federal University of Technology – Paraná
Av. 7 de setembro, 3165 80230-901 Curitiba (PR), Brazil
weinert@cpgei.cefetpr.br, hslopes@pesquisador.cnpq.br

Abstract. This work describes the use of a recently proposed technique – gene expression programming – for knowledge discovery in the data mining task of data classification. We propose a new method for rule encoding and genetic operators that preserve rule integrity, and implemented a system, named GEPCLASS. Due to its encoding scheme, the system allows the automatic discovery of flexible rules, better fitted to data. The performance of GEPCLASS was compared with two genetic programming systems and with C4.5, over four data sets in a five-fold cross-validation procedure. The predictive accuracy for the methods compared were similar, but the computational effort needed by GEPCLASS was significantly smaller than the other. GEPCLASS was able to find simple and accurate rules as it can handle continuous and categorical attributes.

1 Introduction

Gene Expression Programming - GEP [3] is a novel evolutionary algorithm, recently proposed, that includes characteristics from Genetic Algorithms - GA and Genetic Programming - GP. The main difference from GEP to GA and GP is how individuals are encoded.

In GAs individuals are usually a fixed-size linear string of bits (chromosomes) whereas in GP, individuals are non-linear entities of different size and shapes, usually in the form of trees. In GEP, on the other hand, individuals are encoded as fixed-size strings of symbols (chromosome) that, in turn, are expressed as non-linear entities of different size and shapes known as “expression trees”. Since GEP has emerged recently, few applications have been published to date [8]. In this work we apply GEP to the data mining task of classification, where it is aimed to find comprehensible rules capable of modelling a given set of data. The objective of data classification is to predict the value of a goal attribute, giving a set of predicting attributes. Usually, rules are represented in the form IF *< antecedent >* THEN *< consequent >*, where “antecedent” is a logical

* This work was partially supported by a research grant from the Brazilian National Research Council – CNPQ (305720/04-0).

combination of the predicting attributes and their values, and the “consequent” part is the value of the goal attribute (class).

A GEP-based tool, named GEPCLASS (Gene-Expression Programming for CLASSification), was created for this purpose. We describe several modifications on the original GEP algorithm so as to efficiently cope with data classification. The application of GEPCLASS to a number of datasets is reported and its performance is compared with other published papers in recent literature.

2 Gene Expression Programming

Ferreira [3] proposed a new evolutionary algorithm with linear genotype/non-linear phenotype, denominated Gene Expression Programming, using concepts from both GAs and PG. In GEP, chromosomes are simple, compact, linear and relatively small entities, that are manipulated by means of special genetic operators (replication, mutation, recombination, translocation, etc). Expression trees (ETs) are the phenotypical representation of the chromosome. Selection, the central engine of evolutionary computation paradigms, operates over ETs rather than chromosomes. During the reproduction cycle, chromosomes, not ETs, are generated, modified and transmitted to the next generations. In [3], PEG is presented using individuals with only one chromosome and, henceforth, individual and chromosome are used as synonymous. Although it would be possible to use multiple-chromosome individuals, we used that same approach.

Similarly to other evolutionary algorithms, GEP starts with an initial population, either created at random or using some previous knowledge about the problem. Next, chromosomes are expressed into ETs that, in turn, are evaluated according to the specific meaning of the problem, yielding a fitness measure. If a stop criteria is not met, the best individual(s) is(are) kept and the rest are submitted to a fitness-based selection procedure. Selected individuals undergo modifications by means of genetic operators leading to a new generation of individuals. The whole process is repeated until a stopping criterion is satisfied.

2.1 GEP Encoding

The genome in GEP is a single chromosome which, in turn, is composed by one or more genes (that is, multigenic), as in nature. Every gene is divided into two parts: head and tail. The size of the head (h) is determined by the user, and the size of the tail (t) is computed as: $t = h(n - 1) + 1$, considering n the largest arity found in the function set for the particular problem.

The phenotypical representation of a genome is the set of sub-trees (ETs), each one being expressed by a gene, linked together by means of a linking function. This function is user-defined and connects the roots of all sub-trees. For instance, this linking function can be sum or multiplication, for symbolic regression problems, or logical AND or OR for classification problems. Like biological genes, only part of it is really expressed as an ET.

In the same way as GP, GEP also uses a function set and a terminal set as building blocks of possible solutions. Considering that genes have two parts,

namely head and tail, some restrictions apply: in the tail part only terminal elements can occur, whereas in the head of a gene, both terminals and functions are permitted (except for the first position, where only functions are allowed).

2.2 Selection Method and Genetic Operators

The main selection method in GEP is the well-known roulette-wheel, a popular procedure in the early implementations of GA. This method favors the fittest individuals of the population, whose chances to be selected are proportional to their relative fitness. During run, individuals that will undergo the transposition and crossover operators (see below) are randomly selected.

GEP also implements a simple elitist mechanism, where the best individual of a generation is copied to the next, by means of a cloning operator. This procedure guarantees that the best individual found throughout generations is kept.

In the original work of Ferreira [3], several genetic operators were defined.

The mutation operator works similarly as in GP and GA and aims to introduce new genetic material in the current population so as to increase genetic diversity. Due to the particular characteristics of the encoding in GEP, some integrity rules must be obeyed in order to avoid syntactically invalid individuals (see section 3.1).

GEP uses one-point or two-point crossovers, just like GA. The second type is somewhat more interesting since it can turn on and off noncoding regions within the chromosome. Also, a kind of uniform crossover was implemented, and is known as genic recombination. This operator randomly chooses genes of same position in two parent chromosomes to form two new offsprings.

There are three transposition operators: IS (insertion sequence), RIS (root IS) and genic. An IS element is a variable-size sequence of alleles extracted from a random starting point within the genome (even if the genome was composed by several chromosomes). Another position within the genome is chosen and it will be the place where the element will be inserted. This target site must be within the head part of a gene and cannot be the first allele (gene root). The IS element is sequentially inserted in the target site, shifting all alleles from this point forth. The same number of alleles inserted are deleted from the gene head, from the end backwards. This operator simulates the transposition found in the evolution of biological genomes. RIS is similar to the IS transposition, except that the insertion sequence must have a function as first allele and the target point must be also the first allele of a gene (root). Finally, genic transposition swaps genes within a chromosome.

3 Methodology

In this section we describe all modifications in the original GEP algorithm to develop the GEPCLASS system, specifically designed for data classification. Some of these changes were also used successfully in another GEP-based system [8].

3.1 Rule Encoding

Most literature in data-mining use rules in the form *if A then C*, for classifying data. The antecedent *A* is a set of conditions to be met and the consequent *C* is the predicted class. Conditions are t-uples in the form $\{A_i \text{ Op } V_{ij}\}$, where A_i is the *i*-th attribute, *Op* is a relational operator ($=$, \neq , $>$ or $<$), and V_{ij} is the *j*-th value belonging to the domain of attribute A_i . To combine several possible conditions in a rule, logical operators (*and*, *or*, *not*) are used. The consequent of a rule is simply a condition in the form $\{M_i = V_{ij}\}$, where M_i is one of the possible goal attributes, and V_{ij} is a possible value for this goal attribute.

Implementing data classification using evolutionary algorithms requires defining, a priori, whether an individual represents a single rule (Michigan approach) or a complete solution composed by a set of rules (Pittsburg approach) [4]. GEPCLASS can implement both approaches, either by an explicit decision of the user, or allowing the algorithm decide by itself which one is more adequate for a given classification task, during the evolutionary process. For instance, user can define that an individual will generate multiple rules, using more than one gene per chromosome and a logical *or* as linking function between genes. The use of GEP for data classification requires tight restrictions in the individuals' encoding, so as to avoid syntactically invalid individuals. In our approach, we define a closure property that states that any ET root must be a logical function. All logical functions have as offspring nodes other logical or relational functions. These, in turn, always have as offspring nodes attributes and correlated values.

To assure that the closure property of the encoding will be always met, we proposed in GEPCLASS some changes in the original encoding of GEP, at the phenotypical level, and we also defined filling rules for the chromosome. First, a crucial modification in the structure of the chromosome is regarding lengths of head and tail. The tail size (*t*) was defined before for symbolic regression problems [8], but for data classification, we propose a new way to compute it, as follows: $t = \text{int}([h \cdot (n - 1) + 1] / 2)$, where $\text{int}()$ returns the integer part of the argument. This new approach is justified considering the fact that every terminal element is attached to a set of possible values (domain), for instance ($a > 10$), not to another attribute or constant, as in symbolic regression, for instance ($a + b$). Therefore, a compact representation for both terminals and their values reduces half the size of the tail length in a gene. In words, at the implementation level (only), relational operators have arity 1.

Figure 1 presents a 2-genes chromosome with different lengths for heads and tails. Upward arrows show the points delimiting the coding sequence of each gene. The corresponding ET of the chromosome is presented. Notice that the ET maintains the original characteristics of GEP and, at the same time, guarantees rule integrity (syntactically valid rules). A filling rule was defined in GEPCLASS: the tail part of a gene always has only terminals (but head can have terminals and functions). An extension of this rule is the implicit precedence between functions (logical and relational) and terminals. In practice, this is accomplished by means of a constrained syntax, inspired in [2]. There is, the encoding must guarantee that a given operator will receive valid operands (terminals), and, therefore,

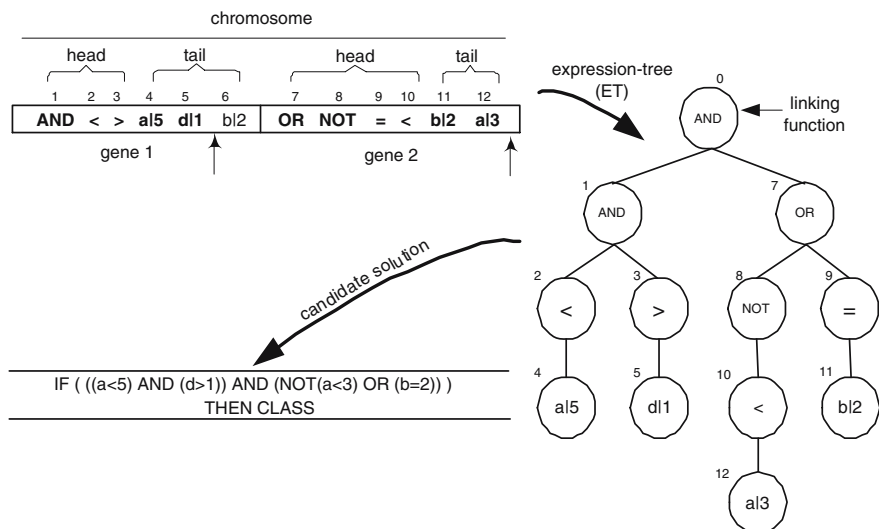


Fig. 1. Example of chromosome structure, ET and corresponding rule in GEPCLASS

will give a valid output to the parent node. Figure 1 clearly exemplifies this accomplishment: at root level (node 0) one can find a logical function (AND) linking two sub-trees; in the second level (nodes 1, 7 and 8) it is found logical operators (*and*, *or*, *not*); in next level (nodes 2, 3, 9 and 10) there are relational operators (<, >, =, <); finally, at leaf level (nodes 4, 5, 11 and 12) there are terminals and respective values. Notice that the arity of relational operators is not changed, only the representation in the tree.

A consequence of the filling rule is regards to the nature of the attributes. In a data set, one can have both continuous and categorical (nominal) attributes. If a given attribute is categorical, GEPCLASS uses only = or \neq as relational operators. Otherwise, if the attribute is continuous or ordered categorical (values mapped in a predetermined scale), all relational operators can be used. In GEPCLASS, any terminal can appear once, many times or none in a rule. This flexible characteristic allows one to find rules within a range, for instance, “ $A_i > 10$) AND ($A_i < 15$)”. On the other hand, this approach allows inconsistencies, such that: “ $A_i = 10$) AND ($A_i = 15$)”. Results obtained up to now (see section 5) have demonstrated that such characteristic is not a serious problem.

3.2 Chromosome Structure, Fitness Function and Genetic Operators

In the original GEP, determining an adequate length for the head of each gene is an open problem. It was stated that more complex problems may require lengthy gene heads, and, therefore, the ideal size is found by trial-and-error.

GEPCLASS uses variable-length chromosomes that can have one or more genes. Genes within a given chromosome are of the same size. Using chromosomes with different lengths in the population can introduce healthy genetic diversity during the search. This approach has also been proved beneficial for symbolic regression problems [8] and is a way to circumvent the open problem of the original GEP in defining adequate lengths for the head of a gene [3]. By default, 50% of the initial population is generated according to a user-defined parameter that defines the maximum and minimum head length of genes, and the rest is randomly generated in the same interval. This procedure was inspired in a similar technique proposed by Koza [6] for GP.

In GEPCLASS we propose two other selection methods over methods originally proposed for GEP. The first method always uses roulette-wheel for selecting individuals regardless of the genetic operator to be used further. This selection method introduces a strong selective pressure that can lead to fast convergence, usually to local maxima in the search space. The second implementation is the stochastic tournament, a successful method frequently used in GA. This strategy is guided by a user-defined parameter (k) that defines the number of individuals that will be randomly selected for a tournament. The individual with highest fitness value will be selected. This method is less aggressive than roulette-wheel and tends to make the algorithms less sensitive to local maxima.

Amongst the many fitness functions proposed before for data mining with evolutionary algorithms [4], we choose to implement in GEPCLASS that proposed by Lopes et al. [7]. This function is the product of two measures: sensibility (Se) and specificity (Sp). Sensitivity measures the fraction of positive instances that will be correctly classified by the system, and it is defined as $Se = tp/(tp + fn)$. Specificity measures the fraction of negative instances that will be classified as such, and it is defined as $Sp = tn/(tn + fp)$. These indicators take into account not only the number of correct classifications, but also the relationship between positive and negative classes. Therefore, the fitness function used has the advantage to maximize both Se and Sp at the same time. Sp and Se , are computed using the number of true-positive (tp), true-negative (tn), false-positive (fp) and false-negative (fn) scores of a rule.

Due to the way rules are encoded in GEPCLASS, most of the original operators needed functional modifications to comply with the closure property, maintaining the hierarchical structure of the ETs after the application of operators.

The mutation operator works in three different levels. When a logical function is selected for mutation, it will be substituted by another logical function. In the same way, when a relational function is selected, it will be replaced by another relational function. Finally, at the leaf level of the ET, when a terminal is selected for mutation, it will be changed by a new attribute and respective random value. There is an exception when the mutation operator is inoperative: when the node selected for mutation is a NOT function, since it is the only function with arity 1 and cannot be substituted by another logical function with higher arity.

The recombination operator works in the same way for chromosomes of different of the same lengths. This operator always swaps the genetic material between the head parts of two genes.

Both the IS and RIS transposition operators work over a single chromosome. IS transposition changes only the tail part of a gene (that is, terminals positions within the gene). Consequently, this operator can move genetic material from one gene to another or within a single gene. On the other hand, in RIS transposition, the donor site is in the tail of a gene, whereas the receptor site is always the first terminal of this same gene.

4 Computational Experiments and Results

Considering that GEP is an extension of GP, we understand that the most fair performance comparison would be GEPCLASS versus a GP-based system for data classification. Therefore, we compared GEPCLASS with a constrained-syntax genetic programming (CSGP) system proposed by [2], over four real-world data sets. In that work there is also a comparison with a “Booleanized” version of genetic programming (BGP) [1] that we reproduced here. The well-known C4.5 decision-tree induction algorithm [9] is often used as the baseline for performance comparisons in data classification literature, and we also included results for this algorithm using the same data. The data sets used for this work were: Ljubljana breast cancer (277/9/2), Wisconsin breast cancer (683/9/2), Dermatology (358/34/6) and Chest pain (138/161/12). Numbers within parenthesis correspond to the number of examples, attributes and classes, respectively. The first three data sets are available at the Machine Learning Repository (<http://www.ics.uci.edu/~mllearn/>), and the last one was described in [1].

The CSGP used in [2] had the following parameters: maximum tree depth: 15; population size: 500 individuals; stopping criterion: 50 generations; recombination, reproduction and mutation operators probabilities: 95%, 5% and 0%. GEPCLASS used the following parameters: population size: 30 individuals; stopping criterion: 50 generations; number of genes per chromosome: 2; linking function: logical *and*; head size range: 6-10; selection method: stochastic tournament; genetic operators: all those defined by the original GEP [3] with the same probabilities. All results reported in this work were obtained by performing a 5-fold cross-validation procedure [5], and using exactly the same data partitions as in [2]. Table 1 shows the accuracy rate for the four data sets, using C4.5, CSGP, BGP, and GEPCLASS.

Table 1. Comparison of accuracy rates (in %)

Data set	C4.5	BGP	CSGP	GEPCLASS
Ljubljana	71.4 ± 0.60	63.9 ± 5.67	71.8 ± 4.68	68.5 ± 12.73
Wisconsin	94.8 ± 0.06	89.3 ± 4.37	93.5 ± 0.79	93.8 ± 2.89
Dermatology	89.1 ± 0.13	86.2 ± 6.24	96.6 ± 1.14	90.5 ± 11.19
Chest pain	73.2 ± 0.77	78.1 ± 4.85	80.3 ± 3.90	88.9 ± 9.61

Table 2. Best rules found by GEPCLASS for Ljubljana, Dermatology and Wisconsin data sets and respective classes

Data set	class	#nodes	accuracy	Rule
Ljubljana	1	16	83.63%	<i>IF</i> ((\sim (<i>Node_caps</i> <> 1)) <i>AND</i> (((<i>Age</i> = 6) <i>OR</i> (<i>Deg_malig</i> <> 2)) <i>OR</i> (<i>Tumor_Size</i> = 3))) <i>THEN</i> (<i>class</i> = 1)
	2	15	94.54%	<i>IF</i> (((<i>Irradiat</i> <> 1) <i>OR</i> (<i>Inv_nodes</i> <> 0)) <i>AND</i> ((<i>Node_caps</i> = 0) <i>OR</i> (<i>Inv_nodes</i> < 5))) <i>THEN</i> (<i>class</i> = 2)
Dermatology	1	9	100%	<i>IF</i> ((\sim (<i>Thinning_suprapapil_epid</i> = 0)) <i>AND</i> (\sim (<i>Follic_horn_plug</i> = 1))) <i>THEN</i> (<i>class</i> = 1)
	2	12	82.85%	<i>IF</i> ((<i>Polygonal_papules</i> < 0) <i>OR</i> (<i>Saw_tooth_appearance_of_retes</i> < 1)) (\sim (<i>Spongiosis</i> < 2)) <i>THEN</i> (<i>class</i> = 2)
	3	15	100%	<i>IF</i> ((<i>Polygonal_papules</i> <> 0) <i>OR</i> (<i>Knee_and_elbow_involvement</i> <> 2)) <i>AND</i> ((<i>Vacuolisation_damage_basal_layer</i> > 0) <i>OR</i> (<i>Oral_mucosal_involvemente</i> <> 0))) <i>THEN</i> (<i>class</i> = 3)
	4	16	100%	<i>IF</i> ((<i>Koebner_phenomenon</i> <> 0) <i>OR</i> (<i>Disappearance_of_the_granular_layer</i> = 1)) <i>AND</i> (<i>Band_like_infiltrate</i> < 2)) <i>AND</i> (\sim (<i>Elongation_of_the_rete_ridges</i> > 0))) <i>THEN</i> (<i>class</i> = 4)
	5	12	100%	<i>IF</i> ((<i>Oral_mucosal_involvemente</i> <> 2) <i>AND</i> (<i>Vacuolisation_damage_basal_layer</i> <> 3)) <i>AND</i> (\sim (<i>Fibrosis_papillary_dermis</i> < 1))) <i>THEN</i> (<i>class</i> = 5)
	6	16	100%	<i>IF</i> ((<i>Disappearance_granular_layer</i> <> 1) <i>AND</i> (<i>Perifollicular_parakeratosis</i> <> 0)) <i>AND</i> (\sim ((<i>Follicular_papules</i> = 0) <i>AND</i> (<i>Perifollicular_parakeratosis</i> < 3)))) <i>THEN</i> (<i>class</i> = 6)
Wisconsin	1	13	97.03%	<i>IF</i> ((\sim (\sim (<i>Bare_Nuclei</i> < 4))) <i>AND</i> ((<i>Bland_Chromatin</i> < 2) <i>OR</i> (<i>Uniformity_of_Cell_Size</i> < 4))) <i>THEN</i> (<i>class</i> = 1)
	2	19	98.51%	<i>IF</i> ((<i>Bare_Nuclei</i> > 5) <i>OR</i> (<i>Uniformity_of_Cell_Shape</i> > 3)) <i>OR</i> (<i>Bland_Chromatin</i> = 7)) <i>AND</i> ((<i>Single_Epithelial_Cell_Size</i> > 1) <i>OR</i> (<i>Mitoses</i> > 3))) <i>THEN</i> (<i>class</i> = 2)

In order to show the simplicity of rules found by GEPCLASS, table 2 shows the best results found, over the five runs, for three out of the four data sets and respective classes. The chest pain data set has 12 classes and takes too much space to be reported. In this table it is shown the number of nodes of the best

solution, its accuracy rate and the composed rule itself. In each rule, the “|AND|” is the linking function and “ \sim ” means the logical “NOT”.

5 Discussion and Conclusions

In this work, we proposed a gene expression programming system for data classification and we compared its performance in four data sets.

The straight comparison for the accuracy rates between GEPCLASS and all other classifiers shows no statistically significant differences, except for the Chest pain data set comparing with C4.5 (t-test with confidence level 5%).

The standard deviations of GEPCLASS results were higher than those for CSGP. This fact could mislead to a false conclusion that GEPCLASS is unstable. In fact this is a direct consequence of the number of overall number of evaluations for each method. The computational effort can be measured by the product of the number of individuals in the population by the average number of generations to achieve the best result. Thus, GEPCLASS needed, at most $30 \times 50 = 1,500$ evaluations, whereas CSGP needed $500 \times 50 = 25,000$ evaluations. For all datasets, the computational effort needed by GEPCLASS was significantly smaller than that needed by CSGP. Therefore, it can be concluded that, for the datasets tested, GEPCLASS can achieve similar results to those found by CSGP, but with less computational effort. Accuracy is also similar to BGP and C4.5.

Comprehensibility (short rules), not only accuracy, is an important issue in data mining. Since GEPCLASS uses a single chromosome with user-defined number of genes, the concept of parsimony (simplicity) is intrinsically implemented, at the same time giving to the algorithm degrees of freedom to find flexible combinations of attributes (see in table 2 the small number of nodes for the rules found). These are the main advantages inherent to the use of the encoding scheme of GEP. On the other hand, CSGP needs an explicit parsimony term in the fitness function to favor smaller rules.

Some considerations are worth to be done about the underlying structure of CSGP and GEPCLASS. The model used in CSGP demands a logical OR in the root node of individuals. Therefore, an individual will represent, at least two rules (following the Pittsburgh approach). Another constraint of CSGP is that offspring nodes having logical AND as parent node, must have also a logical AND or a relational operator. These constraints were devised to simplify the representation of rule antecedents, as normal disjunctive form. In GEPCLASS, there are no such constraints over the structure of rules, allowing a more flexible combination of attributes. GEPCLASS can handle both Pittsburgh and Michigan approaches at the same time, increasing the possibility to find good solutions for complex classification problems. As a consequence, rules found by GEPCLASS can be, sometimes, substantially different (regarding the attributes of the antecedent) from those found by CSGP. Notwithstanding, table 2 shows that the best rules found have excellent accuracy.

The main contribution of this work is to propose a new methodology for the classification task in data mining inspired in the original GEP algorithm. The

developed system can handle both continuous and categorical attributes and is computationally efficient. Future work will include more experiments with other datasets and the implementation of user-defined fitness functions. Also, the usefulness of the original genetic operators for data classification tasks will be evaluated in depth. Authors intend to put GEPCLASS in public domain soon aiming at fostering further research and applications using this tool.

References

1. Bojarczuk, C.C., Lopes, H.S., Freitas, A.A.: Genetic programming for knowledge discovery in chest pain diagnosis. *IEEE Eng. Med. Biol.* **19** (2000) 38-44
2. Bojarczuk, C.C., Lopes, H.S., Freitas, A.A.: A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets. *Artif. Intell. Med.* **30** (2004) 27-48
3. Ferreira, C.: Gene expression programming: a new adaptive algorithm for solving problems. *Complex Syst.* **13** (2001) 87-129
4. Freitas, A.A.: *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag, Berlin, (2002)
5. Hand, D.: *Construction and Assessment of Classification Rules*. John-Wiley & Sons, New-York (1997)
6. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, Cambridge (1992)
7. Lopes, H.S., Coutinho, M.S., Lima, W.C.: An evolutionary approach to simulate cognitive feedback learning in medical domain. In: Sanchez, E. et al. (eds.): *Genetic Algorithms and Fuzzy Logic Systems*. World Scientific, Singapore (1997) 193-207.
8. Lopes, H.S., Weinert, W.R.: EGIPSY: an enhanced gene expression programming approach for symbolic regression problems. *Int. J. Appl. Math. Comput. Sci.* **14**:3 (2004) 375-384.
9. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kauffmann, San Mateo, (1993).

CBR-Based Knowledge Discovery on Results of Evolutionary Design of Logic Circuits*

Shuguang Zhao^{1,3}, Mingying Zhao², Jin Li¹, and Change Wang¹

¹ School of Electronic Engineering, Xidian University
Xi'an 710071, P.R. China

² School of Mechanic-Electronic Engineering, Xidian University
Xi'an 710071, P.R. China

³ College of Information Sciences and Technology, Donghua University
Shanghai 201620, P.R. China

Abstract. Automated design of circuits is a vital task, which becomes more and more challenging due to the conflict of ever-growing scales and complexities of circuits and slow acquisition of relevant knowledge. Evolutionary design of circuit (EDC) combined with data mining is a promising way to solve the problem. To improve EDC in the aspects of efficiency, scalability and capability of optimization, a novel technique is developed. It features an adaptive multi-objective genetic algorithm and interactions between EDC and data mining. The proposed method is validated by the experiments on arithmetic circuits, showing many exciting results especially some novel knowledge discovered from the EDC data.

1 Introduction

As electronic circuits grow rapidly in scale and complexity, automated design of them becomes even more desirable but challenging. Beside lack of relevant experience and knowledge, the extraordinary difficulties of knowledge acquisition make matters even worse. Evolutionary design of circuits (EDC) is a primary branch of Evolvable Hardware (EHW) [1, 2] that applies artificial-evolution based techniques, especially Genetic Algorithms (GAs) [3], to circuit design for the sake of optimal or feasible solutions (i.e. structures and parameters of circuits) in accordance with the design objectives. EDC is capable of automated design of circuits in theory, requiring neither knowledge nor human interventions. But in fact, domain knowledge is helpful or necessary for it to become applicable to the large-scale and/or complex tasks.

This paper is focused on gate-level evolution (GLE) [4-6] of logic (digital) circuits regarding multiple objectives. A GLE usually takes logic gates as building-blocks and evaluates individual's fitness by software simulation. As compared with a function-level EDC [1, 2, 7] that usually employs building-blocks of larger scale and more complex functions and hardware-based fitness evaluation, it has a wider application range, more analyzable results but a lower scalability. So far, most works reported in GLE are concentrated on combinational circuits especially arithmetic circuits, mainly

* This work was supported by China Postdoctoral Science Foundation (No. 2005037783).

for the sake of obtaining novel or efficient building-blocks. But they seldom managed to deal with either multiple design objectives (e.g., circuits' function, gate count and operating speed) or large-scale circuits [4-6, 10]. To realize multi-objective GLE of large-scale circuits, we proposed a novel approach that features an adaptive multi-objective GA that interacts with CBR-based data mining. It is introduced in detail along with some experimental results hereafter.

2 Modeling and Representing of Logic Circuits

As shown in Fig. 1, the circuit model adopted is an $R \times C$ geometry of logic units with P inputs and Q outputs, where each unit has K inputs, M outputs and N functional configurations. It can be used to represent either a combinational circuit or a sequential one, depending on if inner feedback is prohibited or not [12].

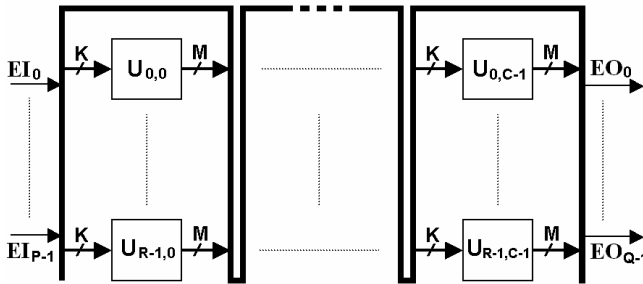


Fig. 1. The abstract model for gate-level evolution

All units including the virtual ones stand for external inputs are orderly numbered. A unit located in row i and column j , $0 \leq i \leq R-1$, $0 \leq j \leq C-1$, is assigned a sequence number $un = P + i + j \cdot R$ and encoded as $[IS_{un,1}, \dots, IS_{un,K}, TS_{un}]$, where $IS_{un,x}$ ($IS_{un,x} \in UI$, $0 \leq x \leq K-1$) equals the sequence number of a unit that feeds signal to its x -th input, and TS_{cn} ($0 \leq TS_{cn} \leq N-1$) corresponds to its current function. By linking up all units' encoding strings with an encoding string indicates sources of the array's outputs, $[OS_0, \dots, OS_{Q-1}]$, the array's encoding or a chromosome is comprised as

$$[IS_{P,1}, \dots, IS_{P,K}, TS_P] \dots [IS_{un,1}, \dots, IS_{un,K}, TS_{un}] \dots [IS_{GN,1}, \dots, IS_{GN,K}, TS_{GN}] [OS_0, \dots, OS_{Q-1}] \quad (1)$$

where $GN = P + R \cdot C - 1$. As to the binary encoding adopted in this paper, $IS_{cn,x}$ as well as OS_j employs $LS = \text{IGE} \lceil \log_2 (GN + 2) \rceil$ bits and TS_i employs $LT = \text{IGE} \lceil \log_2 N \rceil$ bits, and the array's encoding or chromosome amounts to $CL = R \cdot C \cdot (2 \cdot LS + LT) + Q \cdot LS$ bits, where $\text{IGE}[x]$ is a *rounding function* that gives the least integer not less than x . To decrease the problem scale and to improve efficiency of problem solving, it is helpful to adopt as simple as possible configurable logic units, considering the research intention or specific tasks. For most EDC tasks concerning combinational circuits, it is feasible to adopt a two-input logic unit with 4 functional configurations, *AND2*, *OR2*, *NOT* and *XOR* (i.e., *exclusive-OR*), which are universal and most often used in conventional designs. But for evolution of sequential circuits, it is usually necessary

to include some additional memory functions such as registers and latches in the set of functional configurations of logic units.

3 Dynamic Multi-objective Fitness Evaluation

Circuit design is in nature a difficult multi-objective optimization problem like

$$\text{Maximize } f(X) = (f_1(X), f_2(X), \dots, f_n(X)), \quad X \in \Omega \tag{2}$$

To solve the problem efficiently and conveniently, the following fitness function that converts the problem into its single-objective equivalent, optimization of the *sum of weighted objective functions*, was commonly used in EA-based solutions,

$$\text{Maximize } \text{Fit}(X) = \sum_{i=1}^n w_i \bullet \text{Fit}_i(X) \tag{3}$$

where $\text{Fit}_i(X)$ is the normalized objective function corresponding to object function $f_i(X)$; w_i denotes the relevant weight factor to express user-preferences. To lead the GA to co-optimization of the objectives, w_i is designed to vary as follows

$$w_i(t+1) = \alpha \bullet w_i(t) + (1-\alpha) \bullet [2/N - \overline{\text{Fit}_i(t)} / \sum_{j=1}^N \overline{\text{Fit}_j(t)}] \tag{4}$$

where, $\alpha \in [0, 1]$ is a constant parameter ($\alpha = 0.8$ is recommended); $\overline{\text{Fit}_i(t)}$ denotes the average fitness of all individuals in the population. If $\sum w_i(0)=1$ is satisfied, $\sum w_i(t)=1$ will hold forever. Thus, the more an objective optimized the less the relevant weight factor and resultant optimizing pressure on the objective, and vice versa.

As to gate-level EDC, the design objectives mainly include compliance with the desired circuit behaviors, minimization of the gate count and maximization of the operating speed. The first objective function can be expressed as a ratio of Number of Matched Operations (*NMO*) to Total Number of defined Operations (*TNO*)

$$\text{Fit}_1 = \text{NMO} / \text{TNO} \tag{5}$$

where an operation corresponds to a specific input-output combination. It will be scored 1 if the actual response revealed by simulation complies with the expected one, or it will be scored 0. To get a smoother fitness landscape that is consequently easier to search, each output variable is counted independently when computing Fit_1 . That is, a row of a truth (or state) table with m outputs specifies m operations.

Though every candidate circuit seemingly occupies all units (gates) in the array, there exist some *Unused Gates* which have no effective effect on the circuit behavior, e.g. even numbers of *NOT* gates connected in serial, a gate whose output is not referred, etc. They can be identified with the predefined features and the simulation results. So, it is reasonable to express the second objective function in terms of the Number of *Unused Gates* (*NUG*) and the Total Number of Gates (*TNG*), e.g.,

$$\text{Fit}_2 = k_1 \bullet \text{NUG} / \text{TNG} \tag{6}$$

To evaluate a candidate circuit's performance of operating speed, *active time* of each comprised logic gate will be estimated firstly, based on its location in a signal path and the assumption that propagation-delay of a gate is independent of its logic function. Then, the Maximal input-output Propagation-Delay (*MPD*) of a candidate circuit can be figured out and used to express the third objective function

$$Fit_3 = k_2 / MPD \tag{7}$$

where k_1 and k_2 are user-defined parameters for scaling. Then, the fitness function for GA to synthetically evaluate individuals (candidate circuits) can be defined as

$$Fit(t) = \sum_{i=1}^3 w_i(t) \bullet Fit_i(t) \tag{8}$$

4 Adaptation of GA Parameters

The probability of crossover and mutation, P_c and P_m , are enabled to self-adapt to genetic-procedure and individuals' diversity in this paper, because they have great effects on the GA's performance and their optimal values are hard to be preset. The genetic-procedure is estimated with *relative generation number*, a ratio of the current generation number, t , to the maximal one allowed, t_{max} . The individuals' diversity is measured with the concentration degree of current individuals

$$f_d(t) = \overline{f}(t) / [\varepsilon + f_{max}(t) - f_{min}(t)] \tag{9}$$

where $f_{max}(t)$, $f_{min}(t)$ and $\overline{f}(t)$ are maximal, minimum and average fitness of all individuals in the current population, respectively. Thus, $0 < f_d(t) < +\infty$, and $f_d(t)$ will vary simultaneously with the individuals' diversity or distribution. On these bases, P_c and P_m are ordered to adapt themselves in the following manner

$$P_c = \begin{cases} P_{c0} / f_d(t) & t < t_0 \\ P_{c0} \bullet e^{-k_3 \bullet (t-t_0) / t_{max}} / f_d(t) & t_0 \leq t \leq t_1 \\ P_{c0} \bullet e^{-k_3 \bullet (t_1-t_0) / t_{max}} / f_d(t) & t_1 \leq t \leq t_{max} \end{cases} \tag{10}$$

$$P_m = \begin{cases} P_{m0} \bullet f_d(t) & t < t_0 \\ P_{m0} \bullet e^{-k_4 \bullet (t-t_0) / t_{max}} \bullet f_d(t) & t_0 \leq t \leq t_1 \\ P_{m0} \bullet e^{-k_4 \bullet (t_1-t_0) / t_{max}} \bullet f_d(t) & t_1 \leq t \leq t_{max} \end{cases} \tag{11}$$

where, P_{c0} and P_{m0} are initial values of P_c and P_m , respectively; t_0 and t_1 are user-defined parameters, $0 \leq t_0 \leq t_1 \leq t_{max}$; k_3 and k_4 are positive constant parameters. It is usually feasible to let $P_{c0} = 0.8$, $P_{m0} = 0.1$, $t_0 = 0.2t_{max}$, $t_1 = 0.8t_{max}$ and $k_3 = k_4 = 3$. In this way, P_c and P_m will decrease slowly in the evolution process, meanwhile they will respond to the changes of individuals' diversity. With such a self-adaptation mechanism inspired by some principles of bionomics and developmental biology [8, 9], P_c and P_m will be probably suitable for the whole of GA process.

5 CBR-Based Extractions and Reuse of Principles

The most serious problem that hinders EDC from becoming practical is that it is temporarily incompetent for the large-scale and complex tasks. To solve this problem, we tried to integrate EDC with Case-Based Reasoning (CBR) so as to extract novel principles or modules from EDC results and reuse them afterwards.

CBR is a well-known AI technique that solves new problems by using or adapting past solutions. In CBR systems knowledge is embodied in a library of past cases (i.e., case-base), instead of being encoded in classical rules. Each case typically contains a description of the problem, plus a solution containing implicit knowledge and/or the outcome. To solve a current problem with CBR, it is matched against the available cases, and the relevant cases are retrieved and used to suggest a solution to be reused and tested for success and then revised if necessary. Finally the current problem and the final solution are retained as part of a new case-base. Therefore, CBR quite suits EDC requirements of extracting and reusing principles or building-blocks exist in EDC results, as demonstrated by some preliminary results [11, 12].

The primary difficulties herein exist in building a case-base. While CBR relies on cases that have known structure, e.g. attribute value pairs, evolved circuits lack any *understanding* incorporated in their structures. As a result, all knowledge except their functionality must be identified before building a useful case-base. Instead of using a GA as a *knowledge lean* method to generate knowledge for a case-base, a GA with the efficient and flexible genotype-phenotype mapping is used to produce efficient solutions and consequently ease the case-base creation. Moreover, each case stores just its own information on its similarity to all other cases, which is represented by four indexes correspond to its function, behavior, structure and sub-circuits respectively. These indexes are to be calculated only once, and additional cases can be indexed in linear time proportional to the size of the case-base.

The creation procedure can be outlined as follows: 1) Remove redundant information and duplicate circuits in EDC results, compress (in terms of the spaces left in the genotypes) and normalize (in terms of the cells' order) the resultant circuits so as to facilitate the CBR functions of matching, retrieval and adaptation. 2) Split the normalized circuits into sub-circuits and calculations of their structure, behavior and functionality. 3) Separate the sub-circuits into perfect solution elements and imperfect ones for the given requirements. 4) Index the circuits according to their function, behavior, structure and sub-circuits, by using a case-based indexing mechanism.

With the case-base built, matching of cases can be achieved efficiently by using the *Nearest Neighbor Matching* function that ranks the cases, making it possible to discover novel knowledge from the EDC results, including equivalent circuits or logic expressions, transform formulas and optimal circuits (in terms of gate count, operating speed, etc). On these bases, imitating the ways that enable human designers to solve large-scale problems, e.g., decompose a circuit into several modules and/or implement the circuit(s) by assembling the verified building-blocks, appropriate cases or sub-circuits in the case-base can be retrieved, assembled and tested automatically towards larger and more complex circuits that have the desired behaviors and specifications. In this way, as illustrated in Fig. 2, the interaction between EDC and CBR-based data mining can be realized and utilized to help us analyze and understand the EDC results and partly solve the scalability problem of EDC.

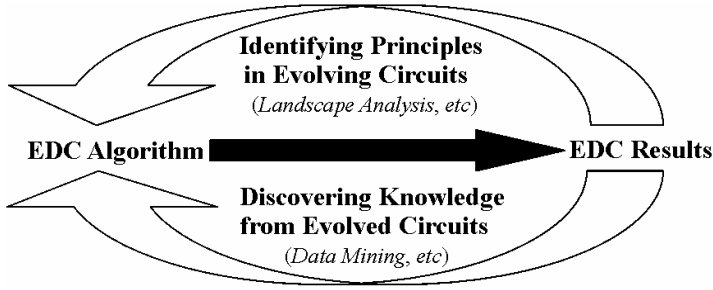


Fig. 2. Interaction between EDC and CBR-based data mining

6 Experimental Results and Discussions

Our approach to scalable EDC is basically a combination of the above ideas and an EGA framework, which is theoretically guaranteed to converge with probability 1. To verify the approach, some experiments were performed on some benchmark problems of gate-level evolution [5, 6, 10], especially arithmetic circuits such as multipliers and even-parity checkers. That is, the EDC experiments were performed firstly and the CBR-based data mining technique was applied to the EDC results afterwards. It is exciting that some evolved circuits are the best ones ever reported and some useful principles and building-blocks have been discovered.

On an n -bit digital multiplier that *outputs the product of two groups of n -bit binary numbers*, a set of experiments was carried out that of increasing inputs ($n=2, 3, 4$). For a 4-bit multiplier, which is rather difficult for human experts to design, a circuit evolved from scratch is depicted in Fig. 3, where NG is short for Number of Gates and ND is short for Number of Delay. It exhibits wondrous reuses of inner outcomes and vastly excels its competitors in performance, including the best one designed by human-experts (NG=64, ND=24) and that evolved by Miller *et al* [6, 10] from a human-designed one (NG=62, ND=24). Herein the latter was evaluated in its reverted form, as each *nonstandard AND gate* in it is equivalent to 2 standard gates (refer to Fig. 4). Moreover, the following expressions for efficient implementation of the carries of a binary addition, the core of a binary multiplication, were identified as

$$CF_{n+1}=CF_n \oplus [I_{n+1} \cdot (I_1 \oplus \dots \oplus I_n)] \tag{12}$$

$$CS_n = \begin{cases} 0 & n < 4 \\ CS_4 = I_1 \cdot I_2 \cdot I_3 \cdot I_4 & n = 4 \\ CS_5 = CS_4 \oplus \{I_5 \cdot [I_1 \cdot I_2 \cdot (I_3 \oplus I_4) \oplus I_3 \cdot I_4 \cdot (I_1 \oplus I_2)]\} & n = 5 \\ CS_6 = CS_5 \oplus \{I_6 \cdot \{I_1 \cdot I_2 \cdot (I_3 \oplus I_4) \oplus I_3 \cdot I_4 \cdot (I_1 \oplus I_2) \\ \oplus I_5 \cdot \{I_1 \cdot I_2 \oplus I_3 \cdot I_4 \oplus [(I_1 \oplus I_2) \cdot (I_3 \oplus I_4)]\}\}\} & n = 6 \end{cases} \tag{13}$$

where CF_n and CS_n denote respectively the least-bit and the secondary-least-bit of the carry derived from n bit-addends, I_1, \dots, I_n ; CF_{n+1} denotes the successor of CF_n with an additional input I_{n+1} , $n \geq 1$. $CF_1=0$. Equation (12) is a universal iterative formula for

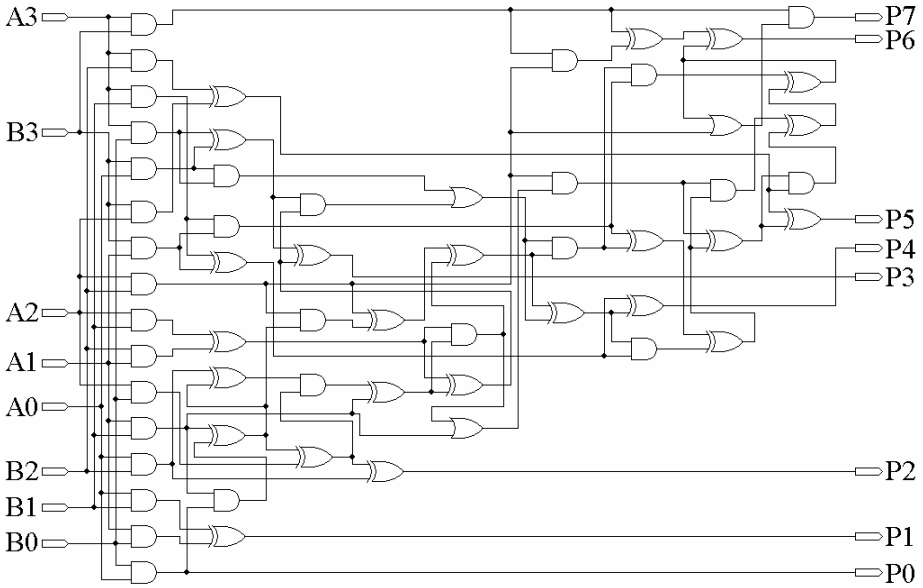


Fig. 3. A 4-bit multiplier evolved (NG=58, ND=18)

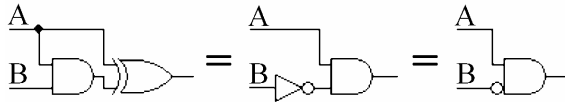


Fig. 4. Standard implementations of the AND gate with an inverted input, as $A \oplus AB = AB'$

efficient implementation of CF_n . Equations (13), although which seem complex and limitedly applicable, are also useful for efficient implementations of multipliers.

Similarly, by making experiments on even-parity checkers of increasing inputs and analyzing the EDC results, which *output '1' iff inputs contain nonzero even numbers of '1'*, a universal iterative formula to design them has been identified as

$$F_{n+1} = I_{n+1} \cdot (I_1 + \dots + I_n) \oplus F_n \tag{14}$$

where F_n denotes the output of a n -bit even-parity checker with n inputs, I_1, \dots, I_n ; F_{n+1} denotes the successor of F_n updated by I_{n+1} joining; $F_1 = 0$. In addition, some novel transform formulae regarding exclusive-OR logic have been identified, e.g.,

$$A \oplus B = (A + B) \cdot (A \cdot B)' = (A + B) \oplus (A \cdot B) \tag{15}$$

$$(A + B) \cdot (A + C) = B \oplus C \oplus (A \cdot B) \oplus (A \cdot C) \tag{16}$$

$$A \cdot (B \oplus C) = (A \cdot B) \oplus (A \cdot C) \tag{17}$$

$$A \oplus B + A \cdot B = A + B \tag{18}$$

Although the above extracted principles are rather difficult for human experts to deduce, with knowledge of Boolean algebra it is not difficult to prove that they are correct and reusable in EDC of larger scale. These facts argue that combining CBR-based data mining with EDC is a promising way to analyze, understand and reuse the EDC results in order to solve the scalability problem of EHW.

To conclude, a novel EDC approach was proposed and validated in this paper, and the experimental results encourage us to release its potential in automated design of and knowledge discovery on circuits of larger scale and/or other types in future.

References

1. Yao X., Higuichi T.: Promises and Challenges of Evolvable Hardware. *IEEE Trans. On Systems Man and Cybernetics-Part C*. 1 (1999) 87–97
2. Zhao S. G.: Study of the Evolutionary Design Methods of Electronic Circuits. Ph.D. dissertation (in Chinese), Xidian University, China (2003)
3. Goldberg D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA (1989)
4. Koza J. R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA (1992)
5. Vassilev V. K., Job D., Miller J. F.: Towards the Automatic Design of More Efficient Digital Circuits. In: *Proceedings of EH'00*. IEEE, PaloAlto (2000) 151–160
6. Coello Coello A. C., et al.: Use of Evolutionary Techniques to Automate the Design of Combinational Circuits. *Inter. J. of Smart Engineering System Design*. 4 (2000) 299–314
7. Zhao S. G., Yang W. H.: Intrinsic Hardware Evolution Based on a Prototype of Function Level FPGA. *Chinese Journal of Computers*. 6 (2002) 666–669
8. Shang Y. C., Cai X. M.: *General Bionomics*. Beijing University Press, Beijing (1992)
9. Gilbert S. F., *Developmental Biology*. 6th edn. Sinauer Associates Inc., Sunderland (2000)
10. Miller J. F., Job D., Vassilev V. K.: Principles in the Evolutionary Design of Digital Circuits: Part I. *J. of Genetic Programming and Evolvable Machines*. 1/2 (2000) 7–35
11. Miller J. F., Job D., Vassilev V. K.: Principles in the Evolutionary Design of Digital Circuits: Part II. *J. of Genetic Programming and Evolvable Machines*. 3 (2000) 259–288
12. Zhao S. G., Jiao L.C., Zhao J.: Multi-objective Evolutionary Design and Knowledge Discovery of Logic Circuits with an Improved Genetic Algorithm. In: Y. Hao et al. (Eds.): *CIS2005, Part I, LNAI 3801*. Springer-Verlag, Berlin Heidelberg (2005) 273-278

Data Summarization Approach to Relational Domain Learning Based on Frequent Pattern to Support the Development of Decision Making

Rayner Alfred^{1, 2} and Dimitar Kazakov¹

¹University of York, Computer Science Department, Heslington,
YO105DD York, United Kingdom
{ralfred, kazakov}@cs.york.ac.uk
<http://www-users.cs.york.ac.uk/~ralfred,~kazakov>

²On Study Leave from Universiti Malaysia Sabah,
School of Engineering and Information Technology,
88999, Kota Kinabalu, Sabah, Malaysia
ralfred@ums.edu.my

Abstract. A new approach is needed to handle huge dataset stored in multiple tables in a very-large database. Data mining and Knowledge Discovery in Databases (KDD) promise to play a crucial role in the way people interact with databases, especially decision support databases where analysis and exploration operations are essential. In this paper, we present related works in Relational Data Mining, define the basic notions of data mining for decision support and the types of data aggregation as a means of categorizing or summarizing data. We then present a novel approach to relational domain learning to support the development of decision making models by introducing automated construction of hierarchical multi-attribute model for decision making. We will describe how relational dataset can naturally be handled to support the construction of hierarchical multi-attribute model by using relational aggregation based on pattern's distance. In this paper, we presents the prototype of "Dynamic Aggregation of Relational Attributes" (hence called DARA) that is capable of supporting the construction of hierarchical multi-attribute model for decision making. We experimentally show these results in a multi-relational domain that shows higher percentage of correctly classified instances and illustrate set of rules extracted from the relational domains to support decision-making.

1 Introduction

The processing power to acquire and store large amount of data on documents has increased dramatically over the last few years. Despite the growing of computational power of modern computers, our abilities, to analyze these data for decision-making, are limited for data stored in relational model (multiple tables). We need to join these multiple tables in order to get more information about a specific record stored in target table that has *one-to-many* relationship with data stored in another table.

However, most traditional data mining tools cannot handle relational dataset with high-dimensional of *one-to-many* relationship, unless pre-processing task is applied to the data for data conversion. For instance, Fig 1 depicts two tables involved in the

hepatitis database. The patients' various exams are not directly related, so joining these tables for a common analysis fails to provide a suitable dataset for discovering rules based on traditional data mining algorithm such as Apriori [32]. In other words, the results deriving from joint tables may lead to data redundancy and thence to distortions in the discovering the rules.

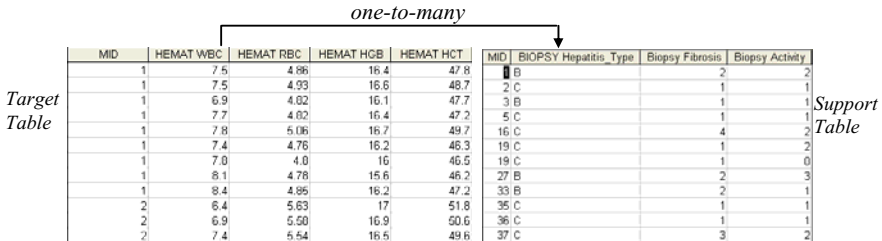


Fig. 1. Hepatitis dataset tables (KDD CUP 2005)

Data representation stored in a relational model differs from the traditional feature-vector (single table) representation in which the relational model is more expressive than attribute-value model in capturing and describing complex structure and relationships in the business/medical domain. Mining knowledge from relational databases to support *decision-making* process has a few advantages over mining knowledge from a single table, namely: *standardized relational format for most database*, *highly expressive power in capturing complex data*, and *the ability to integrate background knowledge* [16], [17]. The integration of relational data mining and decision support is not new, however not much research done in supporting their integration. This work presents a novel approach that is capable of learning relational domain and generating automated hierarchical multi-attribute model to support the development of decision-making system. In this approach, we describe the technique of generalizing data with *one-to-many* relationships using granularity computing as a means of data summarization to automate and support the construction of hierarchical multi-attribute modeling (HMAM) for decision-making [2]. In section 2 we introduce related works in relational data mining. Section 3 covers the concept of hierarchical multi-attribute model in decision modeling. Section 4 describes the pattern-based aggregation approach to relational data mining and discusses the pre-processing procedure. Experimental results are presented in section 5 and this paper is concluded in section 6.

2 Related Works in Relational Data Mining

Relational learning research is not a new research area and it has a long history. Muggleton and DeRaedt [13] introduce the concept of Inductive Logic Programming (ILP) and its theory, methods and implementations in learning multi-relational domains. ILP methods learn a set of existentially unified first-order Horn clauses that can be applied as a classifier [5].

In a relational learner based on logic-based propositionalization [11], instead of searching the first-order hypothesis space directly, one uses a transformation module

to compute a large number of propositional features and then apply a propositional learner. Both ILP and binary propositionalization lack support for numerical aggregation. In general, propositionalization approaches may outperform ILP or MRDM systems, as suggested before in the literature [4], [20]. The choices of aggregation methods and parameters also have significant effects on the results in noisy real-world domains [9]. [12] have conducted a comparative evaluation of approaches to Boolean and numeric aggregation in propositionalization; however their results are inconclusive. In contrast, [16], [9] find that logic-based relational learning and logic-based (binary) propositionalization perform poorly in a noisy domain compared to numerical propositionalization.

Distance-based methods [8], [16] are another variant of relational learning. Their central idea is that it is possible to compute the mutual distance [7] for each pair of object for clustering [3], [15]. Probabilistic Relational Models (PRMs) [6], [10] provide another approach to relational data mining that is grounded in a sound statistical framework. [6], [10] introduce a model that specifies for each attributes of an object, its (probabilistic) dependence on other attributes of that object and on attributes of related objects. Propescul et al. [18] propose a combined approach called Structural Logistic Regression (SLR) that combines relational and statistical learning. Database numeric aggregation [9] techniques propose a method in which aggregation is done by using some of the built-in functions of common relational database system such as *count*, *min*, *max*, *sum*, *avg* and *exist*. Another approach proposed by [17] uses vector distances for dimensionality reduction and is capable of aggregating high-dimensional categorical attributes that traditionally have posed a significant challenge in relational modeling.

3 Decision Support and Hierarchical Multi-Attribute Model

3.1 Decision Support

The term “*decision support*” has a variety of meanings depending on the context on how you use it. Marko [2] outlined the literature review of decision support in details. Decision support can be categorized into *human decision sciences* [24] and *machine decision-making* [23]. [24] defines *human decision sciences* as an interdisciplinary field which addresses three possibly overlapping aspects of human decision making: normative, descriptive and decision support itself. There are some other definitions of decision support that focus on specialized disciplines (Fig. 2), such as operations research and management science [25], decision analysis [26], decision support systems [23], and others including data warehousing [27], group decision support systems [23],[28] and computer-supported cooperative work. Decision analysis introduced by [26] applied decision theory. Decision analysis provides a framework for analysing decision problems by structuring and breaking them down into more manageable parts, and explicitly considering the possible alternatives, available information, uncertainties involved and relevant preferences. In [26], Clemen introduces three models in decision-making, which are *influence diagram*, *decision tree* and *multi-attribute* models.

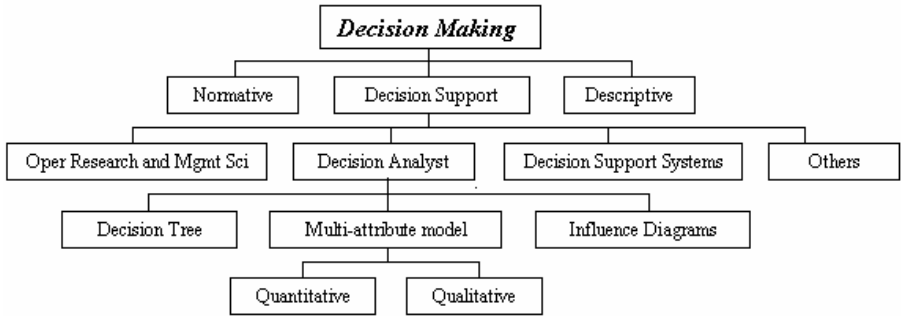


Fig. 2. Decision Making

3.2 Multi-Attribute Modeling

In principle, a multi-attribute model (MAM) [26] represents a decomposition of a decision problem into smaller and less complex sub-problems. A model consists of *attributes* and *utility functions*, as shown in Fig.3. *Attributes* are variables corresponding to decision sub-problems and all attributes at the leaf are basic attributes and attribute at the node is aggregate attribute. *Utility functions* define the relationship between the attributes at different levels in the tree and they serve for the aggregation of partial sub-problems into the overall evaluation or classification of options.

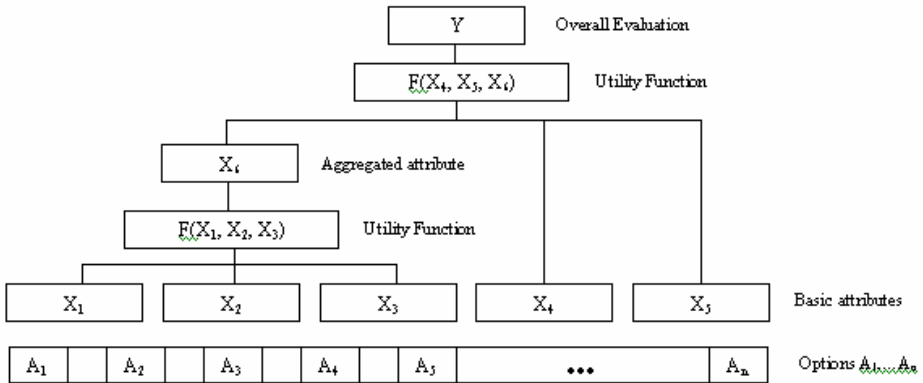


Fig. 3. Components of a Multi-Attribute model

The overall evaluation (utility) of an option is finally obtained as the value of one or more root attributes (Y) in Fig. 3. There are two types of MAM: *quantitative decision model* and *qualitative decision model*. In *Quantitative decision model*, all attributes are continuous and the utility functions are typically defined in term of attributes' weights, such as a weighted average of lower-level attributes [29], [30], [31]. In contrast, in *qualitative decision model*, all attributes are either nominal or ordinal [2], whose values are usually string values rather than numbers and the utility functions

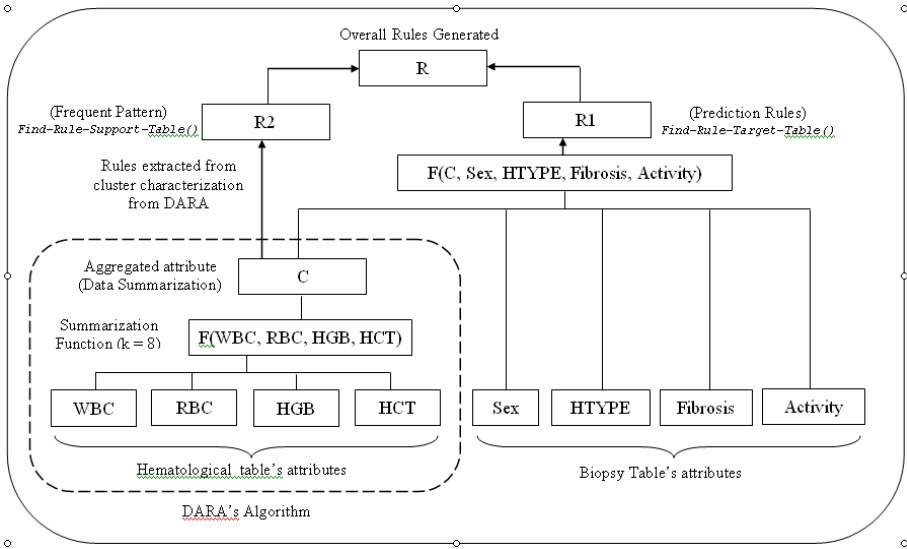


Fig. 4. Components of Multi-Attribute Decision Model for Hepatitis Datasets

use clustering functions to summarize data. This paper emphasizes the *qualitative decision model* in constructing decision making model. Fig. 4 illustrates how learning relational domains using dynamic aggregation based on patterns' distance (DARA) algorithm provides a concrete foundation for bridging relational data mining and MAM. DARA algorithm (Fig. 5) uses data summarization as the utility function to automate the construction of multi-attribute model to support decision-making.

4 Pattern-Based Aggregation

A common method to aggregate a single categorical attribute with numerous patterns is the selection of a subset of pattern that appears most often or distribution based approach. In this approach, each record (row) is viewed as a vector whose dimensions correspond to patterns occurrence in the target table stored in relational domain. Each pattern will have it's own component magnitudes. The component magnitudes are the *pf-irf* weights, as describes in (1), of the patterns which is adapted from *tf-idf* weights [12].

$$pf-irf = pf(p, r) \cdot irf(p) \tag{1}$$

$$irf(p) = \log \frac{|R|}{rf(p)} \tag{2}$$

$$sim(r_i, r_j) = \frac{r_i \cdot r_j}{||r_i|| \cdot ||r_j||} \tag{3}$$

Pf-irf is the product of *pattern frequency Pf(p, r)*, and the *inverse record frequency* (2). *Pf(p, r)* refers to the number of times pattern *p* occurs in the corresponding record

r . In (2), $|R|$ is the number of records in the table and $rf(p)$ is the number of records in which pattern p occurs at least once. Therefore, given two vectors (records) with component magnitudes described in (1), the similarity between two records is then computed in (3), where r_i and r_j are vectors with pf - irf coordinates as described above. Aggregation can be defined as a summarization of the underlying pattern or distribution from which the related objects were sampled. Once we compute the pf - irf weights, then we can compute the distance between each record and cluster them based on their weights. By grouping them into clusters or segments (validated by [1]), we are generalizing or aggregating them based on the underlying pattern or distribution from which the related objects were sample.

```

Input: A relational database
Output: a set of rules distinguish class label.
Procedure:
    Rule set R = empty
    Create-Pattern();
    Compute-Similarity-And-Transform()
    Update-Target-Table()
    Rule r1 = Find-Rule-Target-Table()
    Add r1 to the R.
    Rule r2 = Find-Rule-Support-Table()
    Add r2 to the R
    Return R
End Procedure

```

Fig. 5. Dynamic Aggregations of Relational Attributes Algorithm (DARA)

The process of data summarization is done using DARA's algorithm (Fig. 5) through the data generalization. The generalization task is done by converting each record's measurement into patterns, (in `Create-Pattern()` from Fig. 4). Then, data summarization is done for each record, by first computing the *pattern-frequency* and *inverse-record frequency* (2) and then grouping them based on the distance between records (3). This individual-centered concept, in which all rows belonging to a specific record is considered as a pattern that characterizes the individualism of each record. For instance, Fig. 6 depicts the summarization process for each record using (1) and (3). Firstly, each record is characterized by patterns of WBC, RBC, HGB and HCT measurements and they are converted into binary codes (01 = below normal, 10 = normal, 11 = above normal). Then, using (1), we compute the magnitude weight for each pattern. For example, given $p = 10101010$ and $r = 1$, pf - $irf(10101010,1) = 4 \times \log(5/4) = 0.387$. All records are then clustered (using `Compute-Similarity-And-Transform()` in Fig. 5), based on the records' component magnitudes (1). The component magnitude for each pattern is computed repeatedly for all records.

The set of overall rules, R , (in Fig. 4) is obtained from the combined set of rules, $R1$ and $R2$. $R1$ is obtained by using the function `Find-Rule-Target-Table()`, where existing attribute-value classifiers such as C4.5, Conjunctive Rules and Naïve Bayes are applied. We use the *Weka* software [21] to extract $R1$. $R2$ is induced by using `Find-Rule-Support-Table()` as shown in Fig. 5 that describes each cluster/segment/

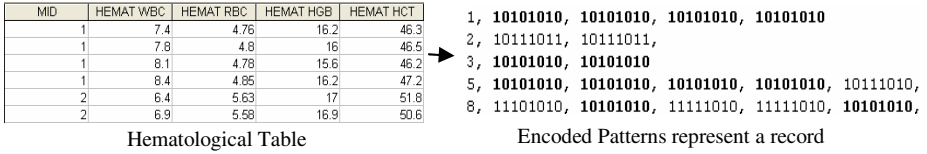


Fig. 6. Data Generalization for one-to-many relationship

group by finding pattern that has the maximum component magnitude for each cluster. The result of our experiment in Hepatitis dataset is discussed in the next section, in which rules generated from the HMAM using DARA is more efficient in terms of the percentage of correctly classified instances. In Fig. 4, we illustrate the integration of the relational learning algorithm, DARA, and HMAM in supporting the construction of decision support system.

5 Experimental Results on Hepatitis Dataset

The database was collected at Chiba University hospital contains information on patients’ exam dating from 1982 to 2001. Among the topics suggested by the Hepatitis dataset, we proposed to evaluate whether the level of biopsy activities and the type of hepatitis can be estimated based on laboratory tests namely WBC, RBC, HGB and HCT. These laboratory tests were chosen based on the work reported by [33]. The approach adopted here consisted of analyzing blood tests together with the biopsy results, seeking patterns that might indicate a correlation between the patients’ exam results and the degree of their activities and also the type of their hepatitis.

The accuracy estimates from 10-fold cross validation result shown in Table 1 using the k-means clustering. In Table 1, the percentage of correctly classified instances for type of hepatitis increases significantly by 1.16% using DARA algorithm using k-means clustering when number of clusters is 8 or 45. In contrast, in Table 2, the percentage of correctly classified instances for Biopsy Activities increases significantly by 1.45% when number of clusters is 40 and 35.

Figure 7 and 8 depicts set of rules, R1, induced using C4.5 [21] for classifying the type of hepatitis and also the Biopsy Activity. For each case, we also get the data

Table 1. Classifiers’ performance for classifying Type of Hepatitis

The Percentage of Correctly Classified Instances for Type of Hepatitis																	
No. of Clusters	0	2	4	6	8	10	15	20	25	30	35	40	45	50	55	60	65
C4.5	70.39	69.96	69.96	69.81	71.55	70.39	70.54	69.81	69.96	70.68	71.41	70.54	71.26	70.10	69.67	69.52	70.25
Naive Bayes	70.39	69.96	70.39	70.39	70.54	70.83	68.94	69.81	68.80	70.39	69.96	71.12	70.39	69.96	70.10	70.10	70.39
Conjunctive Rules	70.39	70.39	70.39	70.39	70.39	70.39	70.39	70.39	70.39	70.25	70.39	70.39	70.39	70.39	70.39	70.39	70.39

Table 2. Classifiers’ performance for classifying Biopsy Activities

The Percentage of Correctly Classified Instances for Biopsy Activities																	
No. of Clusters	0	2	4	6	8	10	15	20	25	30	35	40	45	50	55	60	65
C4.5	61.54	61.54	60.52	60.52	60.52	60.67	60.96	60.96	61.39	60.52	62.26	62.99	61.39	60.96	61.68	61.68	62.26
Naive Bayes	59.65	60.52	61.25	60.23	60.96	61.39	61.54	61.54	61.54	61.39	61.25	63.14	61.39	60.96	61.54	62.12	61.39
Conjunctive Rules	61.54	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65	59.65



Fig. 7. R1 obtained using C4.5 for the type of Hepatitis (k = 8)

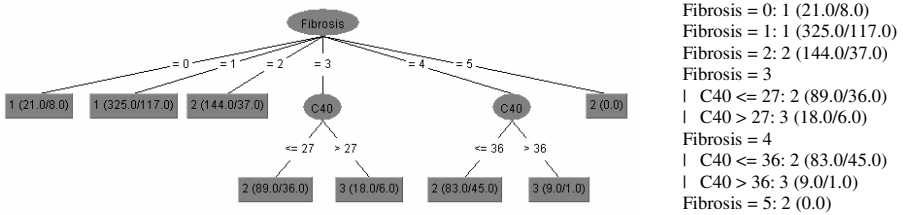


Fig. 8. R1 obtained using C4.5 for the Fibrosis Activity (k = 40)

summarization (R2) for each cluster as shown in Table 3 and 4, where N = Normal, AB = Above Normal, and BN = Below Normal.

Table 3. Type of Hepatitis

Clusters	WBC,RBC,HGB,HCT	Weight
C > 2	N,N,N,N	10391.23
C <= 2	BN,N,BN,BN	2149.39
	N,AN,N,N	2445.45
C > 7	N,N,BN,BN	1395.72
C <= 7	N,N,N,N	10391.23

Table 4. Level of Biopsy Activities

Clusters	WBC,RBC,HGB,HCT	Weight
C <= 36	N,AN,N,N	1947.89
C > 36	N,N,N,N	715.0297
C <= 27	N,AN,N,N	1947.89
C > 27	N,N,N,N	715.0297

Below we summarize the findings based on R1 and R2 for classifying the type of hepatitis in Table 5 and classifying the activities of virus in Table 6.

Table 5. Finding for Classifying type of Hepatitis

TYPE	FINDINGS
Hepatitis C	a) Fibrosis level is F2 or lower and the activity of virus is A1,
	b) Fibrosis level is F2 or lower and the activity of virus is A2 with WBC, RBC, HGB and HCT at normal level
	c) Fibrosis level is greater than F2 and the activity of virus is A3 or lower with WBC, RBC, HGB and HCT at normal level
	d) Fibrosis level is greater than F2 with WBC, RBC, HGB and HCT at normal level
Hepatitis B	a) Fibrosis level is F2 or lower and the activity of virus is A3 or greater,
	b) Fibrosis level is F2 or lower and the activity of virus is A2 with WBC, HGB and HCT at below normal and RBC at normal level
	c) Fibrosis level is greater than F2 with WBC, RBC at normal level but HGB and HCT at below normal level

Table 6. Finding for Classifying Level of Virus Activities

LEVEL	FINDINGS
1	a) Fibrosis level is F0 and F1
2	a) Fibrosis level is F2 b) Fibrosis level is F3 or F4 with WBC, HGB and HCT at normal level and RBC at above normal level
3	a) Fibrosis level is F3 or F4 with WBC, RBC, HGB and HCT at normal level

6 Conclusion and Future Works

In this paper, we propose Dynamic Aggregation of Relational Attributes (DARA), an efficient approach to learning relational domain and integrate DARA with the HMAM to support the modeling of decision support for Hepatitis dataset. The results revealed that *DARA algorithm* generates rules that improve the performance of the classifier. There are some other techniques that can be used to perform the transformation such as Self Organizing Map (SOM) technique. SOM is very effective to be used when we have a lot of missing data and this could improve the transformation-based approach in multi-relational domain. In the future, we would proceed to validate the clinical reasonability of the results and validate the usefulness of the system on other datasets.

References

1. J.C. Bezdek. Some new indexes of cluster validiy, *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 301-315, 1998
2. B. Marko. 2001. *Decision Support*. In D. Mladenic, N. Lavrač, Bohanec, M., and Moyle, S. 2003. *Data Mining and Decision Support: Integration and Collaboration*, Kluwer Aca. Publishers.
3. W. Dillon and M. Goldstein. *Multivariate analysis*, pages 157-208. John Wiley and Sons, Chichester, 1984.
4. S. Džeroski, H. Blockeel, B. Kompare, S. Kramer, B. Pfahringer, W. Van Laer, *Experiments in Predicting Biodegradability*, In Proceedings of ILP '99, 1999
5. S. Džeroski and N. Lavrač, editors. *Relational Data mining*. Springer-Verlag, 2001. ISBN 3540422897.
6. L. Getoor, N.Friedman, D. Koller, and A. Pfeffer. Learning Probabilistic relational models. In S. Džeroski and N. Lavrač, editors. *Relational Data mining*. Springer-Verlag, 2001.
7. T. Horvath, S. Wrobel, and U. Bohnebeck. Relational instance-based learning with lists and terms. *Machine Learning*, 43(1/2): 53-80, 2001.
8. M.Kirsten, S. Wrobel and T. Horvath. Distance based approaches to relational learning and clustering. In S. Džeroski and N. Lavrač, editors. *Relational Data mining*. Springer-Verlag, 2001.
9. A. Knobbe, M. De Haas, and A. Siebes. Propositionalization and aggregates. In *LNAI*, volume 2168, pages 277-288, 2001.
10. D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *AAAI/IAAI*, pages 580-587, 1998.
11. S. Kramer, N. Lavrač and P. Flach. Propositionalization approaches to relational data mining. In S. Džeroski and N. Lavrač, editors. *Relational Data mining*. Springer-Verlag, 2001. ISBN 3540422897.

12. M.A. Krogel, S. Rawles, F. Železny, P.A. Flach, N. Lavrač, and S.Wrobel. Comparative evaluation of approaches to propositionalization. In *13th International Conference on Inductive Logic Programming (ILP)*, pages 197-214, 2003.
13. S.H. Muggleton and L. DeRaedt. Inductive Logic programming: Theory and Methods. *The Journal of Logic Programming*, 19 & 20:629-680, May 1994.
14. S.H. Muggleton. Inverse Entailment and Progol. *New Generation Computing*, 13:245-286, 1995.
15. J. McQueen. Some Methods of classification and analysis of multivariate observations. In *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281-293, 1967
16. C. Perlich and F. Provost. Aggregation-based feature invention and relational concept classes. In *Proceedings of the Ninth ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2003.
17. C. Perlich and F. Provost. ACORA: Distribution-based aggregation for relational learning from identifier attributes. *Journal of Machine Learning*, 2005.
18. A.Propescul, L. H. Ungar, S. Lawrence, and D. M. Pennock. Structural Logistic Regression: Combining relational and statistical learning. In *Proceedings of the workshop on Multi-Relational Data Mining (MRDM-2002)*, pages 130-141. University of Alberta, Edmonton, Canada, July 2002
19. A. Srinivasan and R.D. King. Feature Construction with Inductive Logic Programming: A Study of Quantitative Predictions of Biological Activity Aided by Structural Attributes. *Data Mining and Knowledge Discovery*, 3(1):37-57, 1999.
20. A. Srinivasan, R.D. King, D.W. Bristol, An Assessment of ILP-Assisted Models for Toxicology and the PTE-3 Experiment, In *Proceedings of ILP '99*, 1999
21. I. Witten, and E. Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufman.
22. G. Salton, J. Michael, McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, 1986.
23. D.J. Power, 1999. *Decision Support Systems Glossary*, <http://DSSResources.COM/glossary/>
24. [INSEAD, 2003] INSEAD, 2003. *Decision Sciences*. PhD Program Description, <http://www.insead.edu/phd/program/decision.htm>
25. F.S. Hillier and G.J. Lieberman, 2000. *Introduction to Operation Research*, McGraw Hill.
26. R. T. Clemen, 1996. *Making Hard Decisions: An introduction to Decision Analysis*, Duxbury Press
27. J. Han and M. Kamber, 2001. *Data Mining: Concept and Techniques*, Morgan Kaufman.
28. E.G. Mallach, 1994. *Understanding Decision Support Systems and Expert Systems*, Irwin, Burr Ridge.
29. DAS, 2001. *Decision Analysis Software*. <http://faculty.fuqua.duke.edu/daweb/dasw.htm>
30. H.L.S. Younes, 2001. *Current tools for assisting intelligent agents in real-time decision making*, MSc Thesis, <http://www-2.cs.cmu.edu/~lorens/papers/mscthesis.html>
31. G. Parmigiani, 2002. *Modelling in Medical Decision Making: A Bayesian Approach*. John Wiley & Sons, Ltd.
32. R. Agrawal and R. Srikant, Fast algorithms for mining association rules. In *Proc. of the International Conference on Very Large Databases*, Santiago de Chile, Chile, 1994
33. T. Watanabe, H. Suzuki, and L. Takabayashi. Application of prototipeline to chronic hepatitis data. In *Working core of ECML/PKDD 2003 Discovery Challenge*, p.166-177, 2003.

Extreme Value Dependence in Problems with a Changing Causation Structure

Marlon Núñez and Rafael Morales

Department of Languages and Computer Sciences,
University of Málaga, Málaga 29071, Spain
{mnunez, morales}@lcc.uma.es

Abstract. We explore the role of sequences of extreme values for measuring tail-dependence between times series. The proposed measure concentrates on searching extreme cause-effect fluctuation pairs in the recent time interval and requires much less data than current causality and dependence approaches. The target applications of this approach are those in which there is the necessity of rapidly recognizing the interval time in which a time series may be influenced by other time series characterized by sudden and unpredictable extreme changes. This paper presents the tail-dependence measure in the field of stock markets and compares it to known causality and dependence measures. An application of the mentioned measure in the field of space physics is also presented.

1 Introduction

Time series may describe the behaviour of several physical and economic phenomena. When we have to deal with two time series, the question often arises for describing the interrelationships existing between them. In economics, for example, before applying sophisticated methods for describing relationships between two time series, it is important to check whether they are independent (or serially uncorrelated in the non-Gaussian case) or not. In order to study dependence and causality, a main line of research in complex systems attempts to take into account several effects that give rise to the complex behaviour in general and to extreme events in particular: some of these extreme behaviours rely entirely on a probabilistic description, others emphasize deterministic models with many degrees of freedom, and some attempt to combine these two. In its classical version, the probabilistic theory of extremes (Reiss, R.D. and M. Thomas, 1997), (Beirlant *et al.* 1996), (Embrechts, Resnick & Samorodnitsky 1998) is concerned with the statistical properties of sequences of extreme events. There is increasing evidence, that frequency-size distributions of events extracted from observed time series, whether natural or socio-economic, often deviate significantly from a Gaussian distribution and exhibit “heavy tails” (i.e., more events of large size, a.k.a. “extreme events”) than normally expected (i.e., than would appear in a Gaussian distribution); and the individual extreme events are often correlated with each other.

Observed distributions of extreme events have been fitted, more or less well, to power laws (earthquakes or forest fires), log-Pearson laws of Type III (for floods),

Generalized Pareto Distribution, and so on. For that reason, extreme value theories try to forecast the distribution of extreme events, assuming that future extreme events will have the same distribution as observed in the past. But a problem arises with problems in which the causality structure changes over time. That is, when the distribution of extreme events of the past has changed to an unknown distribution due to a change in the influent time series. In this case, the Extreme Value Theory is not reliable because the statistical properties have changed. On the other hand, a recent sudden change of causality gives no data to find the new distribution and dynamics parameters. This same problem has also to be faced by dependence tests and causality measures when they are dealing with sudden changes in the causality structure.

This paper proposes a strategy for facing measuring tail-dependences in problems with sudden changes in the causality structure. It consists of the continuously measuring of the tail dependence among pairs of time series in order to discover which time series could be a good predictor of the other in a new scenario. The paper is organized as follows: Section 2 introduces the proposed tail-dependence measure between two time series. Section 3 shows comparative results in the field of stock markets. Section 4 presents an application of the mentioned measure in the field of space physics. The conclusions and future directions are presented in Section 5.

2 Measuring Tail Dependence

A strategy to deal with time series in a problem with sudden changes in the causality structure is to provide a quick and continuous measure of the tail-dependence between two series. In order to measure the tail dependence from a time series A to the time series B , the proposed approach will search for “cause-effect” event *pairs*. Our main assumption is that an extreme fluctuation of A might cause an extreme fluctuation in B within a posterior time interval. Our goals are that the measure of tail dependence at a given time t be proportional to the number of hypothetical “cause-effect” pairs found in the immediate past of t and inversely proportional to the standard deviation of the temporal distance between both events, that is, the *separation* of the pairs.

This strategy is very simple, since it only requires a short sequence of past extreme values of the analyzed time series. There is no model to discover or to construct. It uses a simple model: a big event might cause another big event within a certain interval time. If this *cause-effect* sequence is found with some persistence in two time series, then there is a tail-dependence. The rest of the section formalizes the above strategy.

The proposed measure $TailD_{A \rightarrow B, t} \in [-1, 1]$ calculates the dependence of extreme fluctuations of time series B due to extreme fluctuations of time series A , abbreviated as $A \rightarrow B$. This measure will be positive if there exists a direct tail-dependence, that is: extreme fluctuations of time series A cause extreme fluctuations of the same direction (sign) in time series B . $TailD$ will be negative if there exists a contrary tail-dependence, that is: extreme fluctuations of time series A cause extreme fluctuations of contrary direction in time series B . Let us introduce four associated terms:

- ${}^+TailD_{A \rightarrow B, t}$ measures how much influence positive extreme fluctuations in time series A have on extreme positive fluctuations in time series B at time t .
- ${}^-TailD_{A \rightarrow B, t}$ measures how much influence negative extreme fluctuations in time series A have on extreme positive fluctuations in time series B at time t .

- ${}^+TailD_{A \rightarrow B, t}$ measures how much influence positive extreme fluctuations in time series A have on extreme negative fluctuations in time series B at time t .
- ${}^-TailD_{A \rightarrow B, t}$ measures how much influence negative extreme fluctuations in time series A have on extreme negative fluctuations in time series B at time t .

In order to measure the tail dependence at every instant t from time series A to time series B , we construct the ${}^+A$ and ${}^+B$ with all extreme positive first differences greater than a user-defined u percentile. We also construct the ${}^-A$ and ${}^-B$ time series with all extreme negative first differences lower than the $(1-u)$ percentile. .

Our main assumption is that a time series A has influence on a time series B when a sequence of extreme fluctuations of A causes a sequence of extreme fluctuations in B , with a very similar time lag between each of the *cause-effect* fluctuation pairs. The user sets two parameters, L_{min} and L_{max} , assuming that the effects of the time series A in the time series B will occur with a *minimum time lag* of L_{min} time steps and a *maximum time lag* of L_{max} time steps. That is, depending on the theory of the domain, the user sets L_{min} and L_{max} to determine the effect window, in which the influence of one time series over the other is assumed to be located. In order to measure ${}^+TailD_{A \rightarrow B, t}$ at time t the following terms will be used:

- i is an element of the ${}^+A$ time series, with the most positive extreme values of A .
- $time(i)$ is the time of the element i .
- $Pair_i$ is conformed by the element i and the element j of ${}^+B$ (see Figure 1) such that:
 - $time(i) \geq t - L_{max}$, and
 - $time(i) + L_{max} \geq time(j) \geq time(i) + L_{min}$, and
 - j is the most close event to $time(i) + L_{min}$ than any other element of ${}^+B$, and
 - there is no element of ${}^-B$ between $time(j)$ and $time(i) + L_{min}$, and
 - j has not been selected as paired element of any event of ${}^+A$
- $PairSeparation_{i, t, L_{max}, L_{min}}$ is the separation of the discovered *pairs*
- $PairSeparationMean_{t, L_{max}, L_{min}}$ is the mean of all the pair separations.
- $PairSeparationVar_{t, L_{max}, L_{min}}$ is the standard deviation of all the pair separations.
- $PairCount_{t, L_{max}, L_{min}}$ is the number of pairs found in the interval $[t, t - L_{max}]$
- $OddCount_{t, L_{max}, L_{min}}$ is the number of unpaired elements of ${}^+A$ within the interval $[t - L_{max}, t - L_{min}]$ plus the unpaired elements of ${}^+B$ within the interval $[t - L_{max} + L_{min}, t]$

The TailD measure of tail dependence at a given time t will be proportional to the number of hypothetical “cause-effect” pairs found in the immediate past of t and inversely proportional to the standard deviation of the temporal distance between both events. Then the positive-positive tail dependence ${}^+TailD_{t, L}$ is calculated as follows (L is an abbreviation of the parameters L_{min}, L_{max}):

$${}^+tailD_{t, L} = \frac{PairCount_{t, L}}{PairCount_{t, L} + OddCount_{t, L}} \times \frac{PairSeparationMean_{t, L}}{PairSeparationMean_{t, L} + PairSeparationVar_{t, L}} \quad (1)$$

The rest of the terms ${}^-tailD_{t, L}, {}^+tailD_{t, L}, {}^-tailD_{t, L}$ are calculated accordingly. Finally, the measure *TailD* will be the dominant tail-dependence of its four influence measure, that is:

$$tailD_{t,L}^{i,j} = \begin{cases} \max\{^+tailD_{t,L}, ^-tailD_{t,L}\} & \text{if } \max\{^+tailD_{t,L}, ^-tailD_{t,L}\} > \max\{^+tailD_{t,L}, ^-tailD_{t,L}\} \\ -\max\{^+tailD_{t,L}, ^-tailD_{t,L}\} & \text{if } \max\{^+tailD_{t,L}, ^-tailD_{t,L}\} > \max\{^+tailD_{t,L}, ^-tailD_{t,L}\} \end{cases} \quad (2)$$

Figure 1 shows the extreme-value time series ^+A and ^+B to illustrate the selection of the hypothetical “cause-effect” fluctuation pairs. Since $TailD$ at time t is the measure of the tail dependence of the interval time $[t-L_{max}, t]$, which is a short interval, it is necessary an associated measure for larger periods. $averagedTailD$ measures the average of $TailD$ for a user-defined period $[t_1, t_2]$:

$$averagedTailD(t_1, t_2, L_{max}, L_{min}) = \frac{1}{n} \sum_{t=t_1}^{t_2} tailD_{t, L_{max}, L_{min}}$$

The above formula will be useful in Section 3 to compare this approach with two causality measures in a problem of stock market time series.

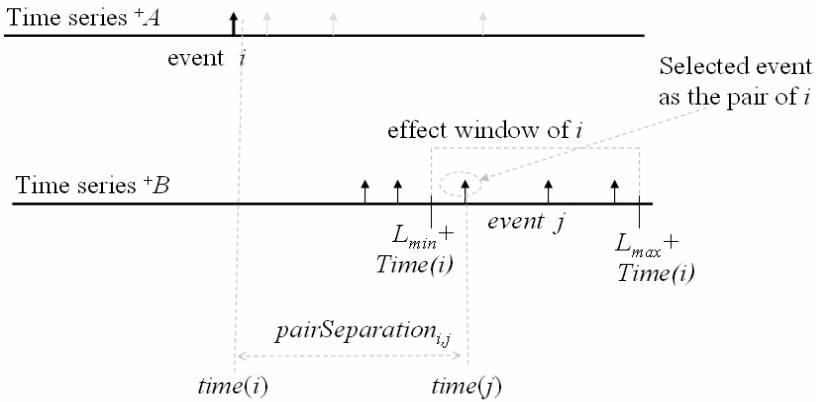


Fig. 1. Illustration of the searching process of hypothetical cause-effect pairs

In order to illustrate the ^+TailD measure, Figure 2b shows a short a portion of two time series. The goal of this example is to measure the positive-positive tail dependence at every time t . Let us assume that the domain expert assures that the effect of time series A to time series B is in at least 6 time steps and at most 20 time steps further. That is, we need to produce a time series of $^+TailD_{t, 6, 20}$. Figure 2a shows three time series: ^+A , ^+B and ^+TailD .

Figure 2a shows that the positive-positive tail dependence ^+TailD measure (dashed line) has detected a high tail dependence because of the existence of three fluctuations of time series A beginning at time step 170 that probably cause other three fluctuations in 16, 16 and 17 time steps later in time series B. Note that the tail dependence was measured with a few data within a short interval time of 20 time steps width. The other associated measures $^-tailD_{t, L_{min}, L_{max}}$, $^+tailD_{t, L_{min}, L_{max}}$, $^-tailD_{t, L_{min}, L_{max}}$ are necessary to be calculated in order to apply the final formula (2).

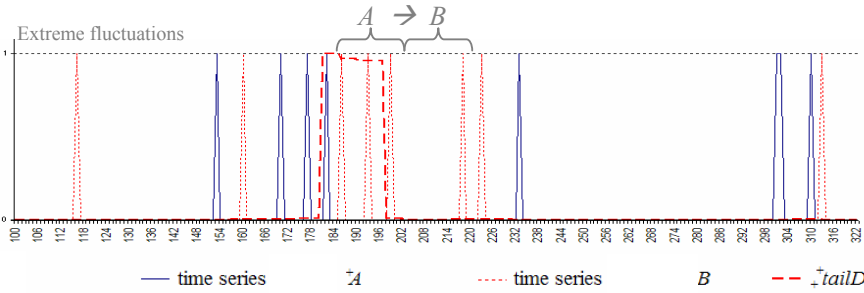


Fig. 2a. Extreme fluctuations of the time series A and B of Figure 2b and the corresponding tail-dependence ${}^+TailD_{A \rightarrow B}$ measure (dashed line)

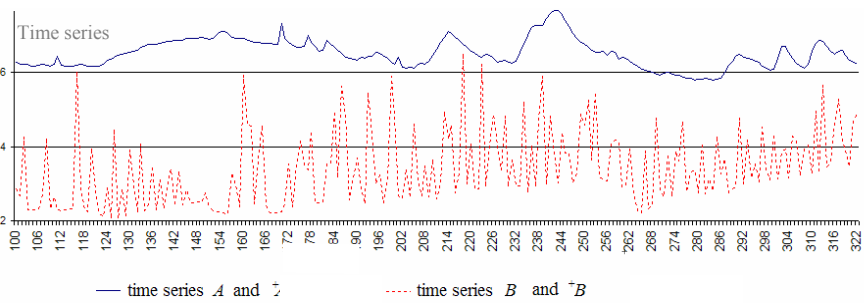


Fig. 2b. Two time series (A and B)

The following sections presents the experimentation using the TailD measure in the fields of stock markets (Section 3) and space physics (Section 4).

3 Tail-Dependence Measure in the Field of Stock Markets

This section investigates empirically the interrelationships between the daily stock market returns of the stock exchanges of Frankfurt (Germany) and New York (USA). Because of sudden structural breaks due to macro economical and political reasons, it is not possible to use any models for *ex ante* prediction, because the strength of the *causalities* is not constant over time. The purpose of the following analysis is to show that, within a limited temporal interval, the dependence measures simply may be used for knowing when extreme fluctuations of one market may be useful for the prediction of the extreme fluctuations of the other market.

Measuring the tail dependence of a short interval is useful in this problem because of a changing structure in the system of global stock markets. In the observation period from 1985 to 1997, international stock markets were subject to changes of economic background as well as a changing psychological behaviour of investors.

In order to compare different approaches, this section compares the proposed measure *TailD* with known models: Granger (1995) and Hosoya (1991) causality

models. The purpose of this experimentation is to compare the discovered changing causality structure between the markets.

To investigate the interactions between these markets in an econometric model correctly, we have to consider the fact that the stock markets are situated in different time zones. Also, their floor trading hours do not overlap. The order of the markets – as given by time - also dictates the sequence in which each market can process new information. The data used for estimation consist of the daily floor trading closing values of the Dax and the Dow Jones Industrial stock indices.

3.1 Data and Market Scenario

Country specific bank holidays led to the omission of these dates from the whole sample. In order to calculate the Granger and Hosoya measures the fluctuation (stock return) was the change in the logarithm of the stock index.

In order to calculate the TailD measure, the fluctuation was the first difference of the stock index. Using first differences of the data, stationarity is assured, which is required for estimation of TailD. Dornau R. (1999) proposed 13-year observation period for analysing possible structural causality changes. Significant breaks were found for the 1987 crash, the breakdown of the Japanese bubble economy beginning 1990 and the stabilization of the Nikkei at the turn of 1992. The last period proposed ends just before the market breakdown October 1997. This leads to the analysis of four separated periods:

- Period 1 : 10/15/85 to 10/15/87 (499 sessions)
- Period 2 : 03/01/88 to 10/02/89 (400 sessions)
- Period 3 : 03/05/90 to 12/29/92 (705 sessions)
- Period 4 : 01/04/93 to 10/20/97 (1253 sessions)

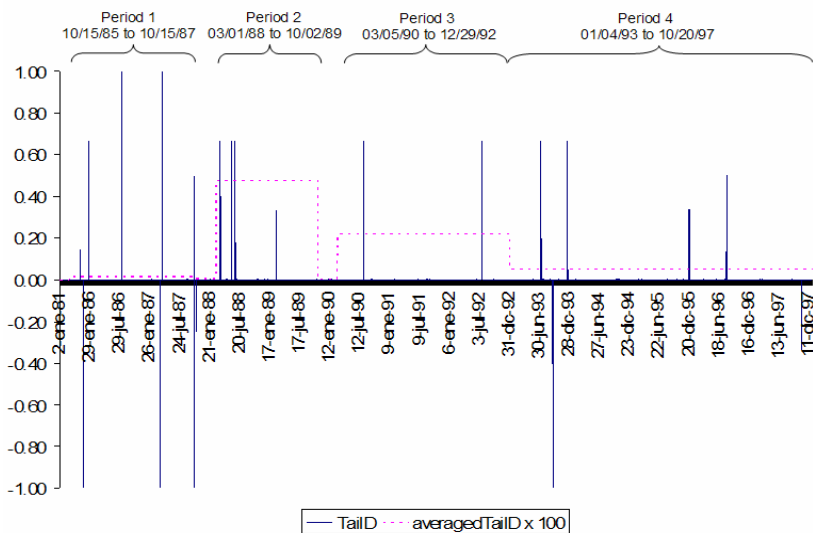


Fig. 3. Tail dependence measures (*TailD* and *averagedTailD*) from the Dax to the Dow Jones

In order to calculate TailD, every daily stock session was divided by two, in order to simulate the order of the markets. That is, a half of a session corresponds to a time-step. Since stock markets are very reactive we consider that an extreme fluctuation in one market might cause an extreme fluctuation in the other in the same half-session, at least, and in four time steps later, at most. That is: $L_{min}=0$ and $L_{max}=3$. A positive extreme fluctuation was the first difference greater than the 95th percentile. A negative extreme fluctuation was the first difference lower than the 5th percentile.

The *averagedTailD* was calculated with the boundary times of the periods selected by Dornau (1999) in his study. Figure 3 shows the measure *TailD* and the *averaged-TailD* from the Dax to the Dow Jones. Figure 4 shows the measure *TailD* and the *averaged-TailD* from the Dow Jones to Dax.

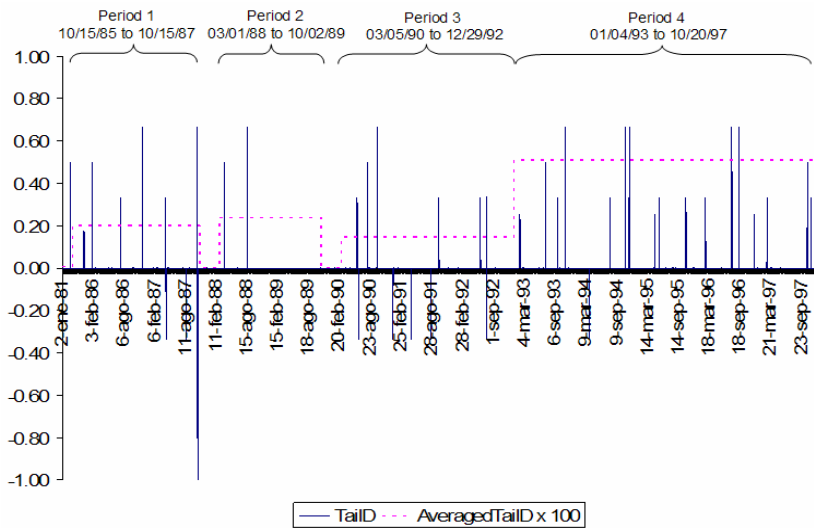


Fig. 4. Tail dependence measures (*TailD* and *averagedTailD*) from the Dow Jones to Dax

The following notes summarize the results of Figure 3 and 4:

- During the first interval (10/15/85 to 10/15/87), the tail dependence from the Dax Jones to the Dow was almost inexistent. This result is consistent with the Granger causality (Dornau, 1999) measure (see Figure 5) and (von Fuerstenberg, Jeon 1989), that reported that the US stockbrokers did not care about foreign markets because of the strong good behaviour of the Dow Jones during this period.
- During the second period (03/01/88 to 10/02/89), the Dax influences the Dow Jones with regard to extreme fluctuations. This result is consistent with the Granger causality measure (Dornau, 1999). During October 1987 there was a crash in stock markets. After a volatile period of 100 days (03/01/1988), investors in the Dow Jones index were sensitive to shocks in foreign markets like the German Dax.

- During the third and fourth periods (03/05/90 to 12/29/92) and (01/04/93 to 10/20/97) the influence of the Dax to the Dow Jones was decreasing, whereas it increases in the opposite direction (see Figure 4). This result is consistent with the Granger causality measure (Dornau, 1999).

3.2 Comparative Results

The theory of global information does not support the theory of causality between the market returns (Dornau, 1999). It has to be assumed that if new global information is available, the reaction of one market is not caused by the change of the previous index but by global information itself. One market uses to follow the other for psychological reasons without any economic or political background. According to Granger’s concept of causality, a test for causality should employ all relevant information. In order to face this requirement and to facilitate the calculations Dornau (1999) used an approximation: the succession of the markets in time of the major stock markets in Japan, Europe and the USA is all the relevant information. This leads to the examination of trivariate models.

In order to test the Granger-causality (Dornau, 1999) used a method introduced by Boudjellaba *et al.* (1992) to give a formulation of the concept of Granger-causality in a general multivariate situation. Another known causality measure was proposed by Hosoya (1991) who introduced a measure for the strength of causality that was extended by Granger and Lin (1995). Implementing Hosoya’s idea leads to the regression of several equations assuming a linear dependence between the markets.

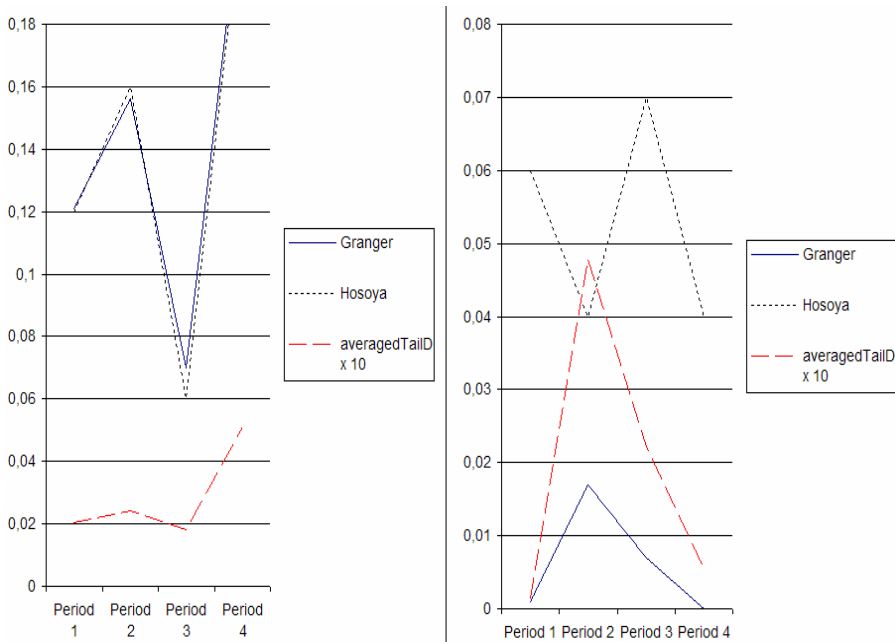


Fig. 5. Comparative results of the analysed three approaches: TailD, Granger and Hosoya causality measures: Left graph) Dow to Dax. Right graph) Dax to Dow

Note that no extreme value behaviour is taken into account in these two known causality analysis. Two of the reasons why Granger and Hosoya methods are somewhat sophisticated are that they take into account all (extreme and non-extreme) behaviour in multivariable analysis, and, on the other hand, all the possible causes have to be included to the study. Therefore several assumptions have to be made. The *TailD* measure unlike above methods, ignores the unpredictable "not extreme" behaviour concentrating simply in the search of extreme cause-effect pairs. However, it is interesting to note that the *averagedTailD* measure yields results (see Figure 5) with similar shape to Granger causality without making sophisticated calculations and assumptions.

All three models give similar relative measures, with the exception of the Hosoya measure of $Dax \rightarrow Dow Jones$ in the period 2. In this period, Hosoya measures gives a low causality measure compared with the rest of the periods. *TailD* and Grangers measures produces similar shapes, but with different absolute values.

4 Tail-Dependence Measure in the Field of Space Physics

This section presents an application of *TailD* as a measure to correlate time series, with the purpose of identifying periods of causation in real space physical phenomena.

4.1 A Very Dynamic and Changing Space Environment

The Sun is a star-sized magnet; its magnetic field permeates the solar system all the way from Mercury to Pluto and beyond. In interplanetary space, the Sun's magnetic field rules. Solar explosions hurl particles across the solar system at nearly light speed. Those particles are guided by the Sun's magnetic field.

Because the Sun rotates on its axis, the Sun's magnetic field out among the planets has a spiral shape. Researchers call it "the Parker spiral" after the physicist who first described it. The energetic solar particles follow the spiral paths all the way. Using the Parker spiral, theoretically, forecasters may predict where energetic solar particles will go. That's a good thing, say, for spacewalking astronauts who want to know when a radiation storm is coming so they can duck inside their spaceship.

But the Parker spiral is not stable; it is dynamic. Therefore the sun-earth magnetic connection becomes intermittent, complicating any prediction. We propose to use the *TailD* to empirically measure this *intermittent influence*. Kiplinger (1995) reported a high correlation between several associated solar flares, which have high X-ray fluxes, with high proton fluxes at Earth. For this reason, an approach for detecting a sun-earth magnetic connection is to estimate when a high X-Ray solar activity at the sun is correlated with a high proton flux behaviour measured at the Earth. Because of the mentioned correlation, only *direct* tail-dependence makes sense in this domain. That is, the factors ^-TailD and ^+TailD (see equation 3) are set to 0 in this domain.

4.2 Results

The goal is to measure the direct tail-dependence between the solar activity in terms of *X-Rays time series* and the arrived particles at the earth in terms of *proton flux time series*. If there is a high correlation, that is, a high *TailD* value, one may assume that there exists a cause-effect due to a magnetic connection.

Figure 6 shows the intermittent magnetic connection level (bottom time series) for two situations occurred during August 2002, when two high proton fluxes occurred. These two hazardous situations are called Solar Proton Events (SPE). In the morning of August 22 a sequence of extreme fluctuations of X-Ray fluxes (see a high and long slope) apparently produced a sequence of extreme fluctuations of proton fluxes afterwards. Note that our approach estimated a high TailD value during the very beginning of the SPE, 4.5 hours before the official onset. In the evening of August 23 and morning of August 24, two sequences of extreme fluctuations of X-Ray fluxes (see two high and long slopes) apparently produced two sequences of extreme fluctuations of proton fluxes afterwards. Again, our approach estimated a high TailD value 1.5 hours before the onset of the SPE.

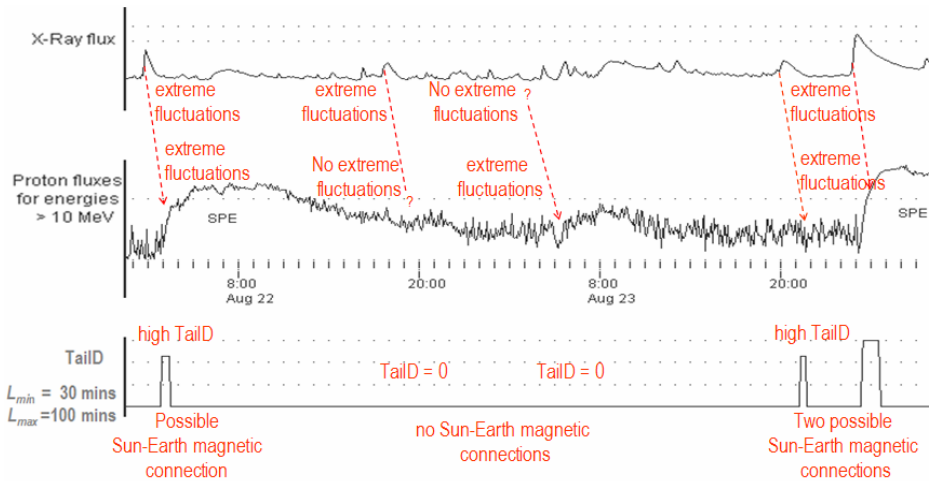


Fig. 6. Tail dependence between X-Ray and proton flux time series. The arrow label “extreme fluctuations” points to a time period with a sequence of several extreme fluctuations.

Currently, there is no space instrument for measuring the *intermittent* sun-earth magnetic connection. Therefore, there is no direct way to evaluate the previously presented approach, but there is an indirect way to evaluate it: The most dramatic consequence of a magnetic connection is the arrival of high proton fluxes; if we use *TailD* as an intermediate step for forecasting the onset of high proton fluxes, we may empirically validate the *TailD* measure for this domain.

Núñez et al. (2006) proposed an algorithm which uses *TailD* as a parameter for identifying magnetic connections. In order to predict the onset of very high proton fluxes, this algorithm takes into account the occurrence of a magnetic connection (a high *TailD* value) and a recent high X-Ray flux¹. Although, the forecasting algorithm

¹ X-ray and proton flux time series were downloaded from <http://goes.ngdc.noaa.gov/data/avg>. In every “Z” file the columns labelled as XL, P2,...P7 are the X-ray and proton flux time series for the specific energies.. We measured *TailD* between the X-Ray time series and each one of proton time series and select the maximum value of every *TailD* value at every time.

is out of the scope of this paper, the low number of false predictions obtained by that algorithm help to validate the intermediate factor *TailD*. Table 1 shows a summary of these results in terms of successfully forecasted very high proton fluxes or SPEs (third column), and the number of false forecasted cases (fourth column) for the period 2002-2005. This table also shows the proportion of successful cases and false warnings that the forecasting algorithm produced for the analyzed historic period (fifth column). We think that these results indirectly validate *TailD* as a measure of intermittent influence in this domain.

Table 1. Summary of the results for forecasting very high proton fluxes (SPEs) obtained by Núñez *et al.* (2006) that uses *TailD* to empirically identify time periods of magnetic connection

	Number of SPEs	Successful forecasted cases (number of SPE onsets successfully forecasted)	Non successful forecasted cases (SPE onset not forecasted)	False alarms. Wrongly forecasted cases
2002	14	6 (42% of SPEs)	8 (57% of SPEs)	15
2003	8	5 (63% of SPEs)	3 (38% of SPEs)	8
2004	6	3 (50% of SPEs)	3 (50% of SPEs)	19
2005	7	6 (85% of SPEs)	1 (14% of SPEs)	19

5 Conclusions and Future Directions

We have given one very general and easily implementable method for measuring tail dependence by searching extreme cause-effect fluctuation pairs in the recent time interval. For this reason it could be used in problems with a sudden changing causality structure. In markets time series, for instance, other common and popular causality measure methods (i.e. Granger and Hosoya causality measures) includes several analysis steps on large datasets. However, the *averagedTailD* give results similar to those obtained by using the Granger causality measures, but without making domain-dependent assumption and large calculations.

Note that the proposed measure depends on two parameters, L_{min} and L_{max} , that determine the position and size of an effect window, in which the influence of one time series over the other is assumed to be located. A future direction is to include a method for an automatic calculation of L_{min} and L_{max} . Anyway, these two parameters should be consistent with the theory (e.g. econometrics, space physics) around the problem, and for this reason, they should always be revised by a domain expert.

An interesting line of research is to explore the use of the tail dependence as part of a forecasting task, and particularly in problems where the causality structure changes dramatically over time. In these problems there is no information about the time of change. Current autoregressive models or partial-memory data mining have problems for adapting fast to new dynamics (Hulten, *et al.*, 2001) (Núñez, *et al.*, 2005). We think that *TailD* offers the capability of knowing when the dependence has changed. Research could be done by using tail-dependence measures as a part of sophisticated forecasting methods in the field of time series (i.e. as a fast dependence test applied to short time intervals) or in the field of data mining (i.e. as another variable to be taken into account for constructing a decision/regression tree).

Acknowledgements

This work was supported in part by CICYT project Moises-TA TIN2005-08832-C03, Spain. I am grateful to Professor Igor Veselovsky (Scobel'syn Institute of Nuclear Physics - Moscow State University) for revising the results of *TailD* as an empirical measure of magnetic connections and his encouragement to continue this research.

References

1. Beirlant, J., Teugels, J. & Vynckier, P.: Practical analysis of extreme values, Leuven University Press, Leuven (1996)
2. Boudjellaba, H., Dufour, J.M., and Roy R.: Testing Causality Between Two Vectors in Multivariate ARMA Models. J. of the American Statistical Assoc., Vol. 87 (1992) 1082-1090
3. Dornau R. Shock around the clock - on the causal relations between international stock markets, the strength of causality and the intensity of shock transmission. Intl. J. of Intelligent Systems in Accounting, Finance & Mgmt, Vol. 8, 4, John Wiley & Sons (1999)
4. Embrechts, P., Kluppelberg, C. , and Mikosch, T.: Modelling extreme events for Insurance and finance, Springer, Berlin (1997)
5. Granger, C., and Lin, J.: Causality in the Long Run. Econometric Theory, Vol. 11 (1995)
6. Hosoya, Y.: On Granger Condition for Non-Causality. Econometrica, Vol. 45, (1991) p1735
7. Hulten G, Spencer L. and Domingos P.: Mining time-changing data streams. Proceedings of the KDD Intl. Conf. on Knowledge Discovery and Data Mining, ACM Press, NY (2001)
8. Kiplinger, A.L.: Comparative Studies of Hard X-Ray Spectral Evolution in Solar Flares with High-Energy Proton Events Observed at Earth. Astrophysical Journal, Vol. 453 (1995)p.973
9. Núñez, M., Fidalgo, R., Morales, R.: On-Line Learning of Decision Trees in Problems with Unknown Dynamics, Lecture Notes in Artificial Intelligence, Vol. 3789, (2005) 443-453
10. von Fuerstenberg G., Jeon B.: International Stock Price Movements: Links and Messages. Brooking Papers on Economic Activity (1989) 125-179
11. Reames, D. V.: Solar Energetic Particle Variations. Adv. Space Research. Vol. 34 (2004)
12. Reiss, R.D. and Thomas, M.: Statistical Analysis of Extreme Values, Birkhauser Verlag, Boston, MA. (1997)
13. Tylka A. J., Boberg P. R., Cohen C.M., Dietrich W. F., Macleannan C. G., Mason G. M., Ng C. K., and Reames D. V.: Flare- and Shock-accelerated Energetic Particles in the Solar Events. The Astrophysical Journal, Vol. 581, part 2, (2002) L119-L123

A Study on Object Recognition Technology Using PCA in the Variable Illumination

Jong-Min Kim and Hwan-Seok Yang

Computer Science and Statistic Graduate School, Chosun University, Korea
mrjjoung@chosun.ac.kr

Abstract. Object recognition technologies using PCA(principal component analysis) recognize objects by deciding representative features of objects in the model image, extracting feature vectors from objects in an image and measuring the distance between them and object representation. Given frequent recognition problems associated with the use of point-to-point distance approach, this study adopted the K-Nearest Neighbor technique(class-to-class) in which a group of object models of the same class is used as recognition unit for the images inputted on a continual input image. However, we propose the object recognition technique new PCA analysis method that discriminates an object in database even in the case that the variation of illumination in training images exists. Object recognition algorithm proposed here represents more enhanced recognition rate to change of illumination than existing methods.

1 Introduction

Object recognition is one of the most actively researched areas in computer vision [1]. An object recognition system can be described in various ways, but it simply finds objects in a given image that match models of known objects in the database [2][3]. In this study, an image containing a single object in absence of background clutter was tested. The image of objects can be acquired in either two-dimensional (2D) image or three-dimensional (3D) image. A 2D shape recognition approach maintains images of objects in one stable position, making it useful for flat objects, whereas a 3D shape recognition approach obtains images of objects from different viewpoints, making it useful for every kind of object. Because objects appear differently from the point of view, 3D images demand more complex object recognition systems than 2D images.

In this study, a collection of images was developed by rotating a 3D object 5 degrees at a time and making a full turn. The performance of recognition systems using principal component analysis is very sensitive to rotation, translation, scale and illumination [4][5]. It is therefore necessary to create many images of objects to be tested and normalize the size of images to keep the performance of recognition system stable. Because of frequent recognition errors involved in the use of point-to-point approach[6], this study used K-Nearest Neighbor approach (class-to-class), in which a group of object models of the same class is used as recognition unit for images inputted on a continual basis, to improve the recognition quality. Normalization and histogram equalization were also performed in object images to ensure a stable recognition rate under varying illumination conditions.

2 Object Recognition Algorithm

Normalization was applied to visual images obtained on a real time basis, and recognition was performed using principal component analysis. A local eigenspace was designed to apply the K-Nearest neighbor decision rule rather than simple distance measures. The proposed algorithm is presented in Fig.1.

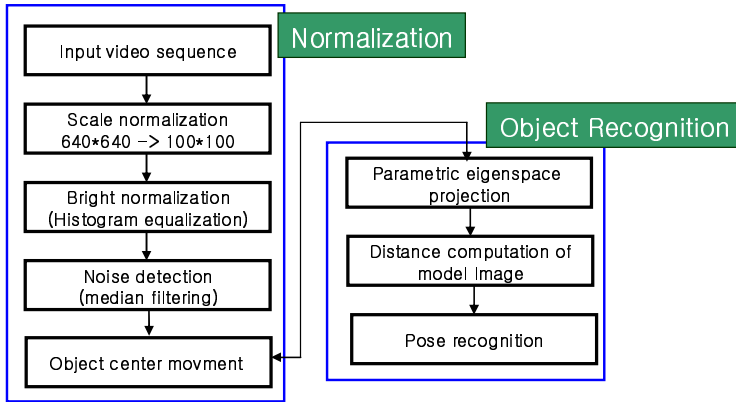


Fig. 1. Composition of proposed algorithm

3 Normalization

Histogram equalization was performed to normalize varying illumination conditions, and median filters were also used to clean up noise. The outcome of histogram equalization is presented in Fig.2.

To eliminate the noise added to the images as a result of the process of histogram equalization, the following equation was employed for median filtering.

$$med(x_i) = \begin{cases} x_{v+1}, & n = 2v + 1 \\ \frac{1}{2}(x_v + x_{v+1}), & n = 2v \end{cases} \quad (1)$$

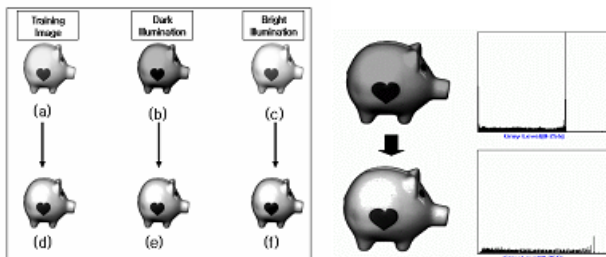


Fig. 2. Object images under variable illumination and result after histogram equalization

4 Object Recognition Using Principal Component Analysis

It is much difficult to show images of object that is rotating continually using a camera. This study proposes a solution based on principal component analysis to obtain images of object that is rotating a certain degree at a time and making a full turn. The solution involved creating a low dimensional vector space to model the overall appearance of object using the equalized output.

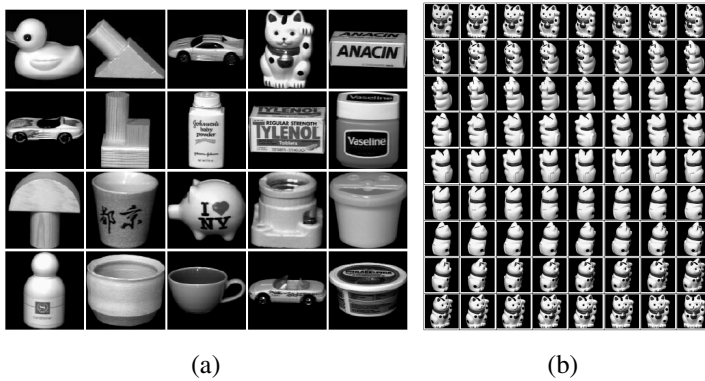


Fig. 3. (a) Object Set (b) Image set obtained by rotating object 5°

The images of objects used for this study is presented in Fig.3(a) and 72 images of an object that rotated 5° degrees at a time is presented in Fig.3(b).

4.1 Eigenspace Calculation with the Principal Component Analysis Method

In order to calculate eigenvectors, the average image was calculated as the center value and then the difference from this center to each image was calculated using the following formulas (2) and (3):

$$C = (1 - N) \sum_{i=1}^N x_i \tag{2}$$

$$X = \{ x_1^{(1)} - c, x_2^{(2)} - c, \dots, x_R^{(p)} - c \} \tag{3}$$

Where C is average image and X is a set of images.

The image matrix X is $N \times M$, where M is the total number of images in the universal set, and N is the number of pixels in each image.

Next, we define the covariance matrix :

$$Q = XX^T \tag{4}$$

That is, eigenvalue λ and eigenvector e of the covariance matrix Q of the images were calculated according to the following equations:

$$\lambda_i e_i = Q e_i \tag{5}$$

Among the studied matrixes, the matrix used as eigenvector is U as it size equals to matrix X . Eigenvectors yielded from the process of SVD(Singular Value Decomposition) can be reconstructed in the order of descending with eigenvalues. An eigenvector's eigenvalue reflects the importance of eigenvector and is calculated using the formula(6). At this time, it is possible not to include every eigenvector into eigenspace and select only main eigenvectors representing features of objects.

$$\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^N \lambda_i} \geq T_1 \tag{6}$$

Where T_1 is the threshold value used for determining the number of vectors.

K was 5 for the low dimensional space used for learning and pose evaluation.

4.2 Image Correlation and Distance in Eigenspace

Once object models are decided by normalized images in the eigenspace, the next step needed for recognition is very simple.

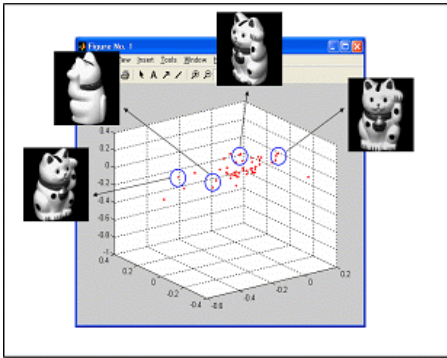


Fig. 4. The distribution of cat images in the eigenspace

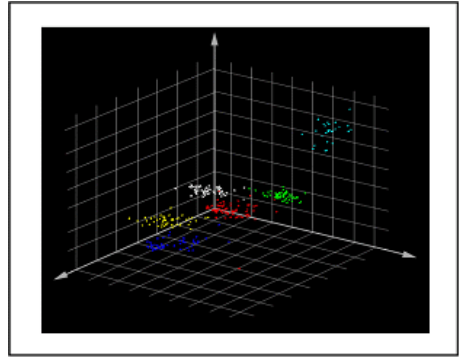


Fig. 5. The distribution of images of all objects in eigenspace

After the difference between the input image X and the average image C was calculated and reflected in the eigenspace according to the following formula:

$$f_j = [e_1, e_2, e_3, \dots, e_k]^T (x_n - c) \tag{7}$$

The new image f_j was represented as a point in the eigenspace.

When reflected, a set of scattered points in the eigenspace correspond to all the objects. Vectors that have a similar feature or value were closely clustered in the ei-

genspace, meaning the images of two identical objects have a similar value and are situated in the same area. The distribution of feature vectors for a set of images of a rotating object is presented in Fig.4, and the distribution of feature vectors for images of all the rotating objects used for the study is presented in Fig.5. The closer distance between the fixed point to a point in the eigenspace is interpreted as a higher correlation between the new image and the previously stored image.

4.3 Distance Measures and Object Recognition with Improved k-Nearest Neighbor

The use of point-to-point approach in pattern classification frequently resulted in a recognition error even if a new image was matched to the previously stored image because of features that were unnecessarily detected in the new image. To solve this recognition problem associated with this individual feature matching technique, features can be analyzed by using a group of object models of the same class as recognition unit for images inputted on a continual basis (Class to Class).

$$w = \frac{(\arg S(M_j) - \text{Min}(\arg S(M_j)))}{d(k-1)} \tag{8}$$

Where $\arg S(M_j) = j$ is an operator meaning the number of object model. K - Nearest Neighbor matching algorithm was used as presented in formulas (8) and (9).

$$\frac{\sum \sum w(I_j - M_j)}{k} \tag{9}$$

Where $K = 3$.

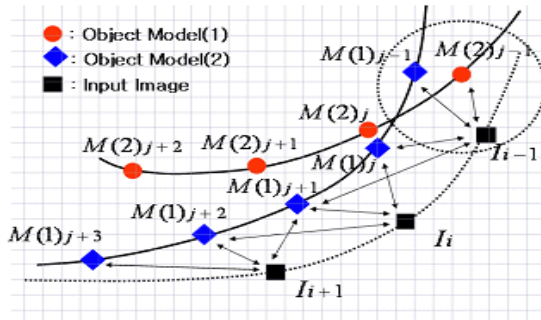


Fig. 6. Recognition based on K -Nearest Neighbor matching algorithm

Recognition of model image and the input image is decided by the value obtained from the formula (9). Based on these formulas, input images were matched to model images as illustrated in the eigenspace(Fig. 6). It was found in the same eigenspace that the input image can be identified as a different object despite its proximity to the model image. To solve this recognition problem resulted from the distance measures between points, this study used K -Nearest Neighbor matching algorithm in which

features are analyzed for matching by using a group of object models of the same class as recognition unit for images inputted on a continual basis. As a result, Recognition quality improved.

5 Experiment Results and Conclusions

5.1 Results from k-Nearest Neighbor Algorithm

The images were taken while each object was rotating 5° at a time and making a full turn. A set of these images is called the image of the object. The images in the size of 640×480 pixels were normalized to the size of 100×100 pixels. Eigenvectors were calculated from a set of object images, and the five-dimensional vectors showing high probabilities were defined as a feature space. As a result, 1000 dimensional images (100×100) were compressed to 5-dimensional images, which were suitable for real-time object recognition. Matching rates of point-to-point approach and edited K -nearest neighbor rule are compared in Table 1. As shown in the table, the matching rates were high with edited k-nearest neighbor rule. A larger number of mismatches were corrected when using the k-nearest neighbor rule.

Table 1. Comparison of matching rates between two classification techniques

Matching	input image	A failure to find a match	Mismatching	Matching rate
Distance measure (Point to Point)	With object models	10.5 %	11 %	78.5 %
	Without object models	15.8 %	20.2 %	62 %
K-Nearest Neighbor (Class to Class)	With object models	6.1 %	3.7 %	90.2 %
	Without object models	13.2 %	16.8 %	70 %

5.2 Effects of Illumination on Object Recognition

Effects of varying illumination conditions on recognition rates are presents in Table 2. Illumination conditions were defined as the unaltered illumination condition (simple principal component analysis), normalized brightness and the condition after histogram equalization.

It was found that matching rates were high when features were classified with k-nearest neighbor rule, compared with those obtained from the use of point-to-point measures. The study findings provided the evidence that k-nearest neighbor was more effective for simple and stable recognition than other techniques using geometric information or stereo images. The images of objects were taken by rotating objects 5° at a time to recognize 3D objects from 2D images, and features of objects of the same class were grouped to be used as recognition unit for 3D object recognition.

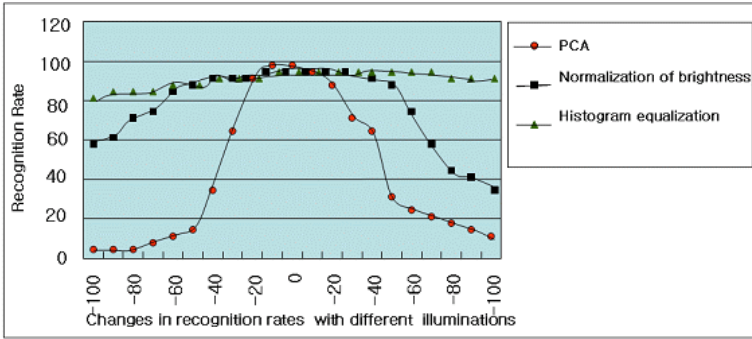


Fig. 7. Comparison of recognition

Table 2. Comparison of recognition rate in the varialbe illumination

Changes in recognition rates with different illuminations	PCA	Normalization of brightness	Histogram equalization
-100	3.33%	57.00%	81.23%
-90	3.33%	61.33%	85.00%
-80	3.34%	70.67%	86.00%
-70	5.00%	76.00%	86.33%
-60	7.67%	83.67%	88.67%
-50	11.33%	88.67%	92.00%
-40	35.00%	91.33%	92.67%
-30	65.66%	91.33%	93.33%
-20	88.20%	91.67%	93.33%
-10	97.00%	92.33%	92.00%
0	96.25%	93.33%	94.67%
10	95.33%	91.33%	92.67%
20	90.67%	90.67%	92.67%
30	68.30%	90.00%	92.67%
40	62.67%	89.67%	92.67%
50	30.33%	84.33%	92.67%
60	25.00%	73.00%	91.00%
70	20.33%	57.66%	91.33%
80	17.00%	44.00%	91.00%
90	14.33%	38.00%	91.00%
100	12.67%	33.67%	90.77%
Total	40.61%	75.70%	90.65%

It is known that recognition systems using principal component analysis undermines recognition quality because of their vulnerability to changes in illumination conditions. The edited k-nearest neighbor rule proposed in this study maintained

recognition rate of more than 90% under varying illumination conditions caused by histogram equalization, and the recognition rate was higher than that obtained from the illumination condition in which brightness was normalized. However, mismatches often occurred during the process of 3D object recognition with objects rotated by 90° . In addition, it was difficult to separate characteristics features of objects such as area from cluttered background. It is therefore important to develop a more stable algorithm for 3D object recognition by solving these problems.

References

1. J.Weng, N.Ahuja, and T.S.Huang, "Learning recognition and segmentation of 3-D object from 2-D images." Proc. of Fourth Int'l Conf. on Computer Vision, pp. 121-128, Belin, May 1993.
2. Paul Viola, M. Jones, "Robust real-time object detection", *International Conference on Computer Vision*, 2001.
3. Hiroshi Murase and Shree K. Nayar, "Visual Learning and Recognition 3-Object from appearance", *international journal of Computer Vision*, Vol,14,1995.
4. Daisaku Arita, Satoshi Yonemoto and Rin-ichiro Taniguchi. Real-time Computer Vision on PC-cluster and Its Application to Real-time Motion Capture. 2000 IEEE.
5. J. Yang, D. Zhang, "Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition," *IEEE Transactions on Pattern analysis and Machine Intelligence* Vol. 26, No. 1, 2004. 1.
6. F. Bourel, C.C Chibelushi and A.A Low, "Robust facial expression recognition using a state-based model of spatially localised facial dynamics", *Proceedings of Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pp.106-111, 2002.
7. A. S. Georghiadis, P. N. Belhumeur and D. J. Kriegman, "From Few to Many : Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.23, no.6, pp.643-660, June, 2001.
8. J. Segen and S. Kumar, "Shadow Gestures: 3D Hand Pose Estimation Using a Single Camera," *CVPR99*, vol. 1, pp. 479-485, Fort Collins, Colorado, June, 23-25, 1999
9. Hwan-Seok Yang, Jong-Min Kim, and Seoung-Kyu Park, "Three Dimensional Gesture Recognition Using Modified Matching Algorithm", *Lecture Notes in Computer Science LNCS3611* pp224-233, 2005
10. P. N. Belhumeur, J. P. Hefanpha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces : Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.19, no.7, pp.711-720, July, 1997.

Pattern Recurring in Three-Dimensional Graph Based on Data Mining

Yanbing Liu^{1,2} and Menghao Wang¹

¹ Chongqing University of Posts and Telecommunications,
Chongqing 400065, P.R. China

² School of Computer Science, UEST of China, Chengdu 610010, P.R. China
liuyb@cqupt.edu.cn

Abstract. With the availability of pattern recognition in digital image processing, it becomes essential to automate the data mining and information processing of graph expression patterns. This paper focuses on mining three-dimensional (3D) patterns of the binocular graphs that are collected through two cameras as the left and right cameras. A new algorithm for the stereo matching and 3D pattern recurring based on data mining is presented. The experimental results show that the algorithm can effectively recur the pattern of scene graphs.

1 Introduction

Data mining is defined as the exploration and analysis, by automatic or semiautomatic means, of large volume of data in order to discover meaningful patterns or rules [1,2]. Using a combination of statistical analysis, modelling techniques and database technology, data mining finds patterns and subtle relationships in data and infers rules that can reconstruct the graphs. Data mining has been applied in a wide range of areas in processing graphs. Graph mining, which is also considered as geographical knowledge discovery, is a branch of data mining that has attracted much attention in recent researches. It puts emphasis on extraction of interesting and implicit knowledge such as spatial patterns or other significant correlations between graphs. Image recurring is one of the key technologies of industrial computed tomography. The goal of pattern recurring research can be briefly stated as to create a remote monitor that is able to see the scene. To define what such seeing ability exactly means, to our purposes here, it entails the pattern extraction from graphics, the information that would allow the correct interpretation of the depicted scene. In the most general case of three-dimensional vision, such interpretation necessitates the reconstruction of 3D scenes, i.e., the attributes of shape, location, pose, and perhaps the movement, of all graph entities. There is a host of computer vision techniques, which aim at the estimation of these intrinsically recorded attributes. Usually, there are three methods to recover a 3D scene structure: shape from shading and photometric stereo, structure from motion and recovery from stereoscopy [3]. The exploitation of remotely sensed multi-sensor (camera) imagery for agricultural,

military, and civilian applications has become an important research area in recent years. Most of the works on shape from shading [4,5] consider the problem as the restoration of the spatial shape of a smooth continuous surface when a continuous function representing the brightness at each point of the surface is given. The motion method has the advantage that the corresponding problem is relatively easy to solve because adjacent graphs are alike. However, the retrieved 3D structure is often inaccurate due to the fact that the baseline between views is small, unless the graph sequence is long enough. Consequently, the estimated depth is sensitive to graph noise. In contrast, with a long baseline the stereo cue could allow very accurate 3D reconstruction [6].

The purpose of this paper is to present a method based on data mining, stereoscopy, and pattern recurring of 3D scene objects. The remaining of this paper is organized as follows. Section 2 describes the system architecture and framework of our method. Section 3 discusses the theory and algorithm. Experimental results are presented in Section 4. Conclusion is drawn in Section 5.

2 System Architecture and Framework of Our Method

Pattern predictive analysis has been used for mining multimedia data and scientific researches, such as remote monitor, astronomy, and geo-scientific researches. Data mining applications often have varying quality and performance [7]. We introduce a new approach for binocular matching and 3D graph pattern recurring of objects based on data mining. The proposed approach considerably reduces the stereo correspondence ambiguity and improves the matching speed by data mining and precisely optimizing the size of stereo matching zoom in this paper. Fig. 1 displays an overview of the developed method. We refer to two cameras as the left and right cameras. First, the stereo matching is completely recurred by combining the left and right monocular correspondences together with approximate knowledge of the cameras, intrinsic parameters, which could be estimated from some off-line data processes of camera calibration and stereo calibration. We stress the fact that the 3D scene and the stereo correspondences are both unknown. Second, with some stereo matches and the stereo geometry, we can mine all the sample-points scene coordinates. Third, all the mined sample-points scene coordinates with the patch-inserted method will recover 3D scene. We consider a pattern to be a rigid sub-structure that may occur in a graph after allowing for an arbitrary number of rotations and translations as well as a small number of operations in the pattern or in the graph.

Taking scene graph that have been carefully classified by user as the data cube, we can construct models for the pattern recurring of the objects, based on properties like magnitudes, areas, shading, and intensity. Preparing the binocular graphs for mining involves two stages of preprocessing. In the first step, binocular graphs from different sensors (camera) is used by means of a 3D site model (shape buildings) and inter-band contrasts of the binocular graphs enhanced through double-camera processing to produce a single graph visualization in 3D. In addition, a binocular graphs may be of intermediate imagery suitable

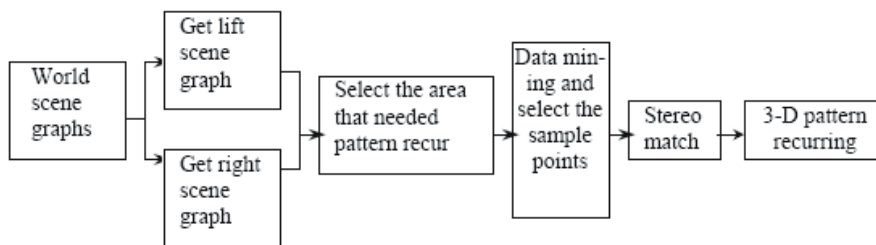


Fig. 1. Illustration of the developed system architecture

for input to a classifier of graph content is created. Next step of double-camera preprocessing enhances extended contours and textural features in the original graph from the processing stage, to enable mining for feature foundation data [8].

The remote mining tool of pattern recurring is a client-server application consisting of a remote client computer with two cameras connected via a network to a local server application. The client application represents the interface with the camera, while the server performs classifier training and the actual mining of the binocular graphs data cube. In order to keep network traffic to a minimum, only sample pixels selected by the client and pattern-match results obtained by the server base on data mining. The pattern mining tool interface provides an intuitive means by which an analyst can get the sample points for pattern recur targets of graph. This interface supports a variety of binocular graphs inspection options including, the changing of display characteristics for optimal viewing of individual graph planes in the data cube and a zoom function.

3 Theory and Algorithm

This paper makes the elucidation of the algorithm to recur the pattern of scene graph based on data mining.

First stage is data mining procedure on the binocular graphs.

Aside from standard methods used in pattern recurring such as edge detection and Hough transformations. For pattern recurring, data mining can help to extract important feature from the binocular graphs captured by camera. Data mining as a process is depicted in Fig. 2 and consists of an iterative sequence of the following steps: data collection, data cleaning, data selection, data transformation, data mining and so on. After the collection of the data of graph, the data cleaning process can help to remove noise and inconsistent data in stereo matching. Then the data relevant to the recurring task are retrieved from the graph cube. Before data mining, the binocular graph data should be transformed into forms appropriate for mining. With the pre-process of the graphs' data, the data mining process can easily and efficiently get all the sample points. Before pattern recurring could happen, some preprocessing should be done depends on the algorithm and graph attributes that has been decided to match and recur. Some stereo operation must be performed to extract the descriptions of the geometric features that will be involved in the process of data mining.

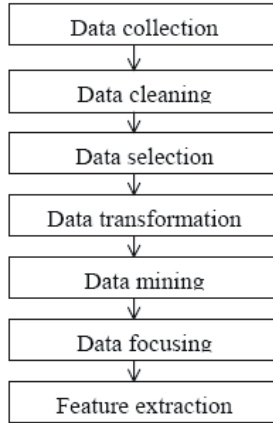


Fig. 2. Flow chart of the data mining process

Second stage is selecting the sample points and running stereo match on the bin-ocular graphs.

At the phase of data selection, Uncertainties mainly stem from a sub-graph of the binocular graph data according to the task of data mining, including what sample point data should be selected, and how much sample point data is enough, also these data necessarily recur scene graphs or uncertainties. We can use data discretization to divide a given continuous graph-attribute data (like light intensity and shading) into discrete values, and this operation is a main origins of uncertainties in the whole process of data mining. Uncertainties from data mining mainly refer to the limitation of recur-ring models, and reconstructing algorithm may further propagate, enlarge the uncertainty during the mining process [9]. To calculate the scene coordinates, we need to do stereo matching that is to select the same point on the binocular graphs. The stereo method requires that the shift of each point be known in one graph with respect to the other. But it is not possible to find the shift of each pixel, as it leads to a lot of incoherent matches. Hence, a mining window is chosen around the point and the sub-graph thus obtained is used to locate the best possible occurrence of the same in the other graph of the stereoscopic pair [3,10]. To find out corresponding point in the right graph, a mining window with a size of $W \times W$ is chosen and the middle point (i, j) is the point we want to find the same on the right graph. $f(i, j)$ denotes the light intensity of the point (i, j) and the point on right graph is chosen using plots of deviation Δd calculated using in Equation (1) for all candidate points along a horizontal row of pixels for various mining window. When Δd is the least one with different point (k, l) on the right graph, (k, l) is the points that we need. Obviously, $k - l$ should be an even number.

$$\Delta d = \sum_{n=-(k-l)/2}^{(k-l)/2} |f(i+n, j+n) - f(k+n, l+n)| \quad (1)$$

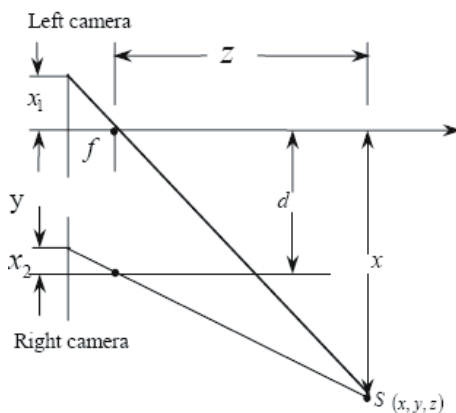


Fig. 3. Triangle law

The optimization of the window size is a key for the highest speed of matching. Let $S = W \times W$ denote the window size. With the larger S , the pattern matching speed is slower and the matching precision is more accurate, namely the miss-matching rate is lower. With the smaller S , the pattern matching speed is higher and the miss-matching rate is higher. So we should choose a suitable mining window size S to get the good matching speed with the enduring miss matching rate.

Third stage is data processing and computing for the scene coordinate. The appropriate recurring of the data within each binocular graph can support data processing. For example, light intensity data with missing values could lie at different levels in binocular graph from the 'clean data'. Associated light intensity to indicate the boundaries between binocular graphs affords the graph mining remove noisy or missing values. The matching point (x_2, y_2) on the right graph can be selected from the specified point (x_1, y_1) on the left one by stereo matching. Denote its scene coordinates by (x, y, z) . The simplest model is two identical camera of equal focal length f and separated by a distance of d .

Two sub-graphs are taken by two same camera as the scene object precisely moved a known distance. Using the laws of triangulation from similar triangles in Fig. 3, we can conclude Equation (2), Equation (3) and Equation (4).

$$x = -\frac{d \cdot x_1}{|x_1 - x_2|} \tag{2}$$

$$y = -\frac{d \cdot y_1}{|y_1 - y_2|} \tag{3}$$

$$z = \frac{d \cdot f}{|x_1 - x_2|} \tag{4}$$

If the coordinates (x_1, y_1) and (x_2, y_2) are estimated by stereo matching, the scene coordinates (x, y, z) is obtained.

Fourth stage is pattern recurring of the scene graph.

Though the above three steps, all the samples' scene coordinates can be obtained. By the transformation from five points to a polyhedron, and connect the

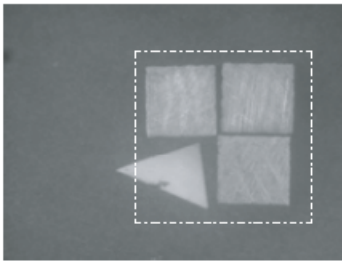
necessary point with line and 3D shape, we can get recurred-pattern of the scene graphs according the points' scene coordinates system [3,10].

4 Experimental Results

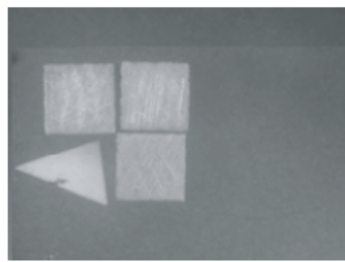
In experiments, the data used for image reconstruction are in bitmap(BMP) format. The reconstructed results are also displayed and saved in bitmap format.

Table 1. Part of the sample-point matching data

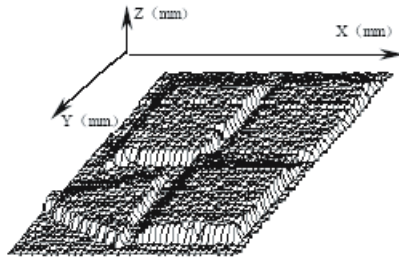
<i>x</i> of left graph's coordinate (<i>pixel</i>)	<i>y</i> of left graph's coordinate (<i>pixel</i>)	<i>x</i> of right graph's coordinate (<i>pixel</i>)	<i>y</i> of right graph's coordinate (<i>pixel</i>)	<i>x</i> of scene coordinate (<i>mm</i>)	<i>y</i> of scene coordinate (<i>mm</i>)	<i>z</i> of scene coordinate (<i>mm</i>)
140	68	12	58	0.5145	0.2499	42.38
146	86	10	76	0.5366	0.3161	39.90
152	122	10	112	0.5586	0.4484	41.73
158	74	27	63	0.5806	0.2720	41.39
164	92	35	82	0.60273	0.3381	42.06
170	98	40	88	0.6248	0.3602	41.73
176	104	46	94	0.6488	0.3822	41.73
182	212	52	203	0.6689	0.7791	41.76
218	242	89	232	0.8012	0.8894	42.06



(1) Left camera graph



(2) Right camera graph



(3) The pattern-recurred graph

Fig. 4. The binocular graphs and the recurred graph

Using bitmap format makes program easier process simulated scanning data because the uncompressed image format. The algorithm is implemented in Java programming language. Table 1 shows data that compute from the given parameters in different binocular graphs [3,10]. Then the system can use sample points in the area that are of interests. Second, with the matching method, all the sample points on the left graph can be matched to the right graph, and then all sample-points parallaxes are obtain. With the pattern recurring method, the third graph is drawn. The third graph pattern is recurred with different size of area and different amount of sample points.

The area needing to be recovered is selected on the left graph. We select the rectangle field those vertex are (132,68), (132,225), (366,68), (366,225). In Fig. 4, the experimental result shows that, this algorithm can efficiently recur and recover 3D shape well from the selected area with different binocular graphs.

5 Conclusion

Together with visualization techniques, graph mining is becoming an effective approach to pattern recurring in 3D graphs. In this paper we presented a pattern recurring algorithm of the scene graph based on the binocular graphs. The trial applications of the algorithm have verified the feasibility and effectiveness that recur the pattern of scene graph.

Acknowledgement

The authors wish to express their sincere thanks to all those who have worked or are currently working with them for their helpful discussions. The work is supported by the Natural Science Foundation of CQUPT, CSTC under Grant No.2005BB2060 and the Natural Science Foundation of CQMEC under Grant No.KJ050507.

References

1. David L. Olson, Desheng Wu, Decision Making with Uncertainty and Data Mining. First International Conference on Advanced Data Mining and Applications (ADMA 2005). Wuhan, China, July, (2005)1-9.
2. M. Berry, G. LinHoff, Data Mining Techniques for Marketing, Sales and Customer Support, Chapter 1, Wiley Computer, Publishing, (1997).
3. Wang Menghao, Liu Yanbing and Zhang Xiaofeng, 3-D surface reconstruction of monolayer grinding wheel topography based on stereo vision. Opto-Electronic Engineering, Vol. 32, 2,(2005)26-29.
4. Horn, B.K.P., Obtaining shape from shading information. In: Horn, B.K.P. and Brooks, M.J. (eds.): Shape from shading. Cambridge, Massachusetts, The MIT Press (1989).
5. Brooks, M.J. and Choinacki, W., Direct computation of shape from shading. Rapport derecherche n 2176, INRIA, France, January (1994).

6. Yu Hongbo, Zhao Rongchun, Wang Bing. 3D Surface Reconstruction Based on Shadow Computer Engineering and Application, 10,(2004)30-33.
7. Parthasarathy, S., Towards Network-Aware Data Mining, Parallel and Distributed Processing Symposium., Proceedings 15th International. 23-27 April, (2001) 1589 -1597.
8. Streilein, W., Waxman, A. and Ross, W., et al., Fused Multi-Sensor Image Mining for Feature Foundation Data, Information Fusion, 2000. Proceedings of the Third International Conference on Volume 1, 10-13, vol.1, July, (2000)TUC3/18 - TUC3/25.
9. Binbin He, Tao Fang and Da-zhi Guo, Uncertainty In Spatial Data Mining, Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August, (2004)1152-1156.
10. Zhang Xiaofeng, Tong Hua and Huang Hua, Journal of Nanchang Institute of Aeronautical Tech-nology (natural science), The matching algorithm of moulding boards with binocular images based on the skeleton angular point, Vol.17,4,(2003)62-64.
11. Han, J., Kamber M.(2001). *DATA MINING: Concepts and Techniques*. Beijing: High Education Press and Morgan Kaufmann Publishers,(2001).
12. Zhong Qu, ART of Image Reconstruction with Narrow Fan-Beam Based on Data, irst In-ternational Conference on Advanced Data Mining and Applications (ADMA 2005). Wuhan, China, July, (2005)407-414.

Mining the Useful Skyline Set Based on the Acceptable Difference

Zhenhua Huang and Wei Wang

Fudan University, China
{051021055, weiwang1}@fudan.edu.com

Abstract. The efficiency of skyline query processing has recently received a lot of attention in database community. However, researchers often ignore that the skyline set will be beyond control in the applications which must deal with enormous data set. Consequently, it is not useful for users at all. In this paper, we propose a novel skyline reducing algorithm, i.e. *SRANF*. *SRANF* algorithm adopts the technique of noise filtering. It filters skyline noises directly on the original data set based on the acceptable difference, and returns the objects which can not be filtered from the original data set. Furthermore, our experiment demonstrated that *SRANF* is both efficient and effective.

1 Introduction

Recently, there has been a growing interest in so-called skyline queries [1, 2]. The skyline of a set of points is defined as those points that are not dominated by any other point. A point dominates another point if it is as good or better in all dimensions and better in at least one dimension.

Numerous algorithms have been proposed for skyline retrieval. Borzsonyi et al.[1] propose the methods based on divide-and-conquer(DC) and block nested loop(BNL). Specifically, DC divides the dataset into several partitions that can fit in memory. The skylines in all partitions are computed separately using a main-memory algorithm, and then merged to produce the final skyline. BNL essentially compares each tuple in the database with all the other records, and outputs the tuple only if it is not dominated in any case. The sort-first-skyline (SFS) [3] sorts the input data according to a preference function, after which the skyline can be found in another pass over the sorted list. Tan et al.[2] propose a solution that deploys the highly CPU-efficient bit-operations by computing the skyline from some bitmaps capturing the original dataset. The authors also provide another method based on some clever observations on the relationships between the skyline and the minimum coordinates of individual points. Kossman et al.[4] present an algorithm that finds the skyline with numerous nearest neighbor searches. An improved approach following this idea appears in[5]. Balke et al. [6] study skyline computation in web information systems, applying the “threshold” algorithm of [7].

In [8], Bentley et al. established and proved that the average number of skyline tuples is $O((\ln N)^{d-1})$. This is the cardinality bound most often cited and employed in

skyline work. And in [9], it is established that $\Theta((\ln N)^{d-1} / (d-1)!)$. From these two papers, we can find that the result set of skyline queries will increase exponentially when the size of the original data set and the number of dimensions increase gradually. Hence, it is lack of manageability. It is important to note that if skyline queries return over a half size of the original data set, the result set is not useful for users at all.

Motivated by these facts, we propose a novel skyline reducing algorithm (i.e. *SRANF*) based on the acceptable difference between tuples in the original data set in this paper. In general, the acceptable difference is given by users. *SRANF* algorithm adopts the technique of noise filtering. It filters skyline noises directly on the original data set based on the acceptable difference, and returns the objects which can not be filtered from the original data set. Furthermore, our experiment demonstrated that *SRANF* is both efficient and effective.

2 Skyline Noise Filtering Algorithm

2.1 Skyline Set Noise Filtering

Intuitively, we can process the skyline set directly, and eliminate those tuples which are dominated by the others in the range of the acceptable difference.

Example 1. (Acceptable Difference) Let P, Q are 3-dimensional objects, where $P=(3, 5, 8)$ and $Q=(2, 9, 16)$. It is easy to see that Q does not dominate P , so the skyline result is $\{P, Q\}$. Obviously, the skyline query returns the whole original data set. Q does not dominate P since the value of the first dimension of Q equals to 2 which is smaller than the one of P (i.e.3). And the difference between these two values is only 1 (i.e. $3-2=1$). If users omit this slight difference, then the skyline query only returns the object Q .

Definition 1. (Skyline Noise Coefficient) Let Ω be the set of N -dimensional tuples. Then the skyline noise coefficient T is defined as N -dimensional vector $(\chi_1, \chi_2, \dots, \chi_N)$, where χ_j is the acceptable difference which is given by users, $1 \leq j \leq N$.

Definition 2. (T-Skyline Noise) Let Ω be the set of N -dimensional tuples, and $\nabla(\Omega)$ be the skyline set on Ω , and $T(\chi_1, \chi_2, \dots, \chi_N)$ be the skyline noise coefficient. Then a tuple O is defined as *T-Skyline Noise* if it satisfies:

- 1) $O \in \nabla(\Omega)$;
 - 2) $\exists O', O' \in \nabla(\Omega) \wedge O' + (\chi_1, \chi_2, \dots, \chi_N) \succ O \wedge \sum_{i=1}^N (\Pi_i(O')) > \sum_{i=1}^N (\Pi_i(O))$.
- And the formula $\sum_{i=1}^N (\Pi_i(O'))$ is the sum of values of all dimensions of O' .

From definition 2, we see that the number of the skyline noises will increase with the increasing of the skyline noise coefficient, so the result set becomes smaller.

Definition 3. (T-Skyline Noise Filter) A T-Skyline Noise Filter is a functional component which has three parameters. And we denote it as $f^T(T, Input, Output)$. f^T is an algorithm module M which realizes the function of T-skyline noise filtering. Each element is defined as follows:

- 1) T : Skyline Noise Coefficient;
- 2) Input: the basic skyline set, that is, $\text{Input} = \nabla(\Omega)$;
- 3) Output: the result set by filtering T -skyline noise, that is, $\text{Output} \subseteq \nabla(\Omega)$;
- 4) M :
 1. $\text{list} := \text{sortSet}(\text{Input})$;
 - /* sort the basic skyline set based on $\sum_{i=1}^N (\Pi_i(O))$ */
 2. $Z := \emptyset$;
 3. While ($\text{list.count} > 0$) do
 4. $Z := Z \cup \{\text{list}[0]\}$;
 5. $O := \text{list}[0]$;
 6. $\text{list.delete}(0)$;
 7. For $i := 1$ to list.count do
 8. if $\forall w \in [1, N], \Pi_w(O) + \chi_w \geq \Pi_w(\text{list}[i])$ then
 9. $\text{list.delete}(i)$;

From definition 3, we see that the T -skyline noise filter can further delete those tuples which are not the local optimal choices for users. And each deleted tuple can be obtained by returning the approximate tuple which is not deleted by the T -skyline noise filter.

2.2 Original Data Set Noise Filtering

Generally, we only have the original data set in the beginning. So, we must delete the tuples that are not the basic skyline set, and filter those tuples which are in the basic skyline set but do not satisfy the local optimal choices for users. In this case, we combine the skyline computing with the technique of the skyline noise filtering, and get the result set directly. This thought of skyline reduction is based on noise filtering. In fact, it is also a method which considers the acceptable difference in the data set. *SRANF* (Skyline Reducing Algorithm based on Noise Filtering) only need to modify the parameter *Input* and improve the algorithm module *M* in definition 3. The pseudocode of *SRANF* algorithm is showed below.

Algorithm *SRANF*(T , Input)

```

/*  $T$  is the Skyline Noise Coefficient; Input is the set of
 $N$ -dimensional tuples; Output is the result set by filtering
 $T$ -skyline noise */
list := sortSet(Input);
Z :=  $\emptyset$ ;
 $\delta := 1$ ;
Z := Z  $\cup$  {list[0]};
list.delete(0);
While(list.count > 0) do

```

```

f := false;
For j := 1 to δ do
    if ∀w∈[1,N], Πw(O)+χw ≥ Πw(list[i]) then
        f := true;
        break;
if f=false then
    Z := Z ∪ {list[0]};
    δ := δ+1;
list.delete(0);

```

SRANF algorithm first sorts the *Input* based on $\sum_{i=1}^N (\Pi_i(O))$; then *SRANF* compares each tuple ϑ in the list with every element in the set Z , and if there's no tuples in the set Z which cause the tuple ϑ to be a T-skyline noise, then *SRANF* puts ϑ into the set Z . Moreover, the theorem 1 shows that all the tuples in the set Z are the local optimal choices for users.

Theorem 1. (Correctness of *SRANF*) The tuples in the set Z are the local optimality for the preference of users.

Proof Sketch. First, *SRANF* sorts the *Input* based on $\sum_{i=1}^N (\Pi_i(O))$, so $\forall O \in Z, \neg \exists O' (O' \in (\Omega - Z) \wedge O' \succ O)$. This formula can be obtained by reductio ad absurdum. If the formula is not correct, $\exists O' (O' \in (\Omega - Z) \wedge O' \succ O) \Rightarrow \forall w \in [1, N], \Pi_w(O') > \Pi_w(O)$. It shows that there exists a tuple O' in the list which is better than the tuple O in the set Z on each dimension, that is, $\sum_{i=1}^N (\Pi_i(O')) > \sum_{i=1}^N (\Pi_i(O))$. And it is impossible. So, the set Z is the subset of the basic skyline set. Secondly, according to *SRANF* algorithm, we see $\forall O \in Z, \neg \exists O' (O' \in \Omega \wedge \sum_{i=1}^N (\Pi_i(O')) > \sum_{i=1}^N (\Pi_i(O)) \wedge O' + (\chi_1, \chi_2, \dots, \chi_N) \succ O)$, so the tuples in the set Z are not T-skyline noise, that is, the tuples in the set Z can not be filtered. Hence, these tuples are the local optimality for the preference of users.

3 Experiments

In this section, we report the results of our experimental evaluation in terms of three aspects:

- a. Skyline manageability: the size of the result set is whether or not fit for users;
- b. Skyline distance: the tuples in the result set are whether or not acceptably dissimilar;
- c. Skyline efficiency: the runtime performance is whether or not acceptable.

The databases used in all our experiments are generated in a similar way as described in [1]. And we only use the independent databases, where attribute values of tuples are generated using an uniform distribution.

All the experiments are carried out on a PC with a 1.6 GHz processor and 512 MB of main memory running the Windows 2000 operating system. All algorithms are implemented in JBuilder and Oracle9i.

3.1 Experiment 1: Comparing the Skyline Manageability

In this experiment, we examine the size of the result set produced by our algorithm and basic skyline method denoted as *BSM*. We compare them using tuples of dimensions 5 and 10. The size of original data set changes from 5×10^5 to 3.0×10^6 . The noise coefficient $T(\chi_1, \chi_2, \dots, \chi_N)$ equals to $(5, 5, \dots, 5)$, that is, $\forall w \in [1, N], \chi_w = 5$. Figure 1 shows the result of experiment 1.

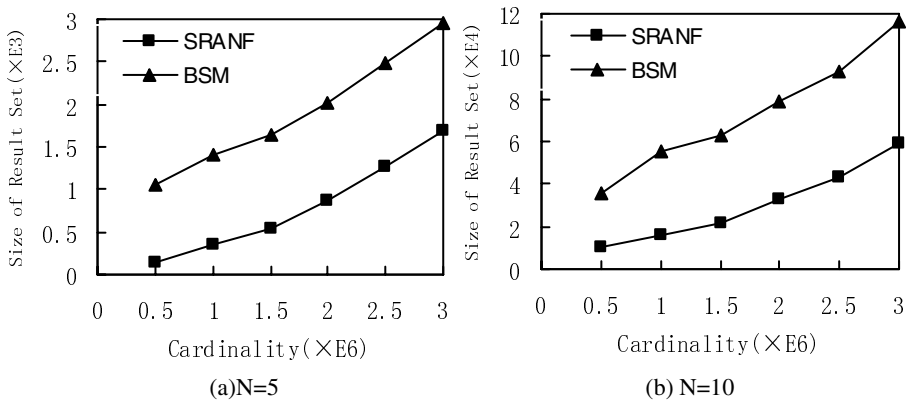


Fig. 1. Size of Result Size vs. Cardinality

From experiment 1, we observe that the size of the Skyline set will increase with the increasing of the size of the original data set and the number of dimensions. In Figure 1 (b), when the size of the original data set equals to 3.0×10^6 and the number of dimensions equals to 10, the size of the skyline set is about 1.2×10^5 . It is obvious that users can not select the optimal object from such enormous result set. As a result, the skyline set is not useful for users' decision at all. And we observe that the result set produced by our algorithm is about 40% of the basic skyline set. If we increase the noise coefficient $T(\chi_1, \chi_2, \dots, \chi_N)$, the result set produced by our algorithm will become smaller than 40% of the basic skyline set.

3.2 Experiment 2: Comparing the Skyline Distance

The database and the parameters which are used in this experiment are the same as the ones used in experiment 1. And we let the measurement of skyline distance be ξ :

$$\sum_{i=1}^L \left(\sum_{j=1 \wedge j \neq i}^{L-1} \left| \sum_{k=1}^N \Pi_k(O^i) - \sum_{k=1}^N \Pi_k(O^j) \right| \right) / L(L-1);$$

The symbol N denotes the number of dimensions of tuples, and the symbol L denotes the cardinality of the result set. It is not difficult to see that the skyline distance increases with the increasing of ξ , and the result set is more useful for users. Figure 2 shows the result of experiment 2.

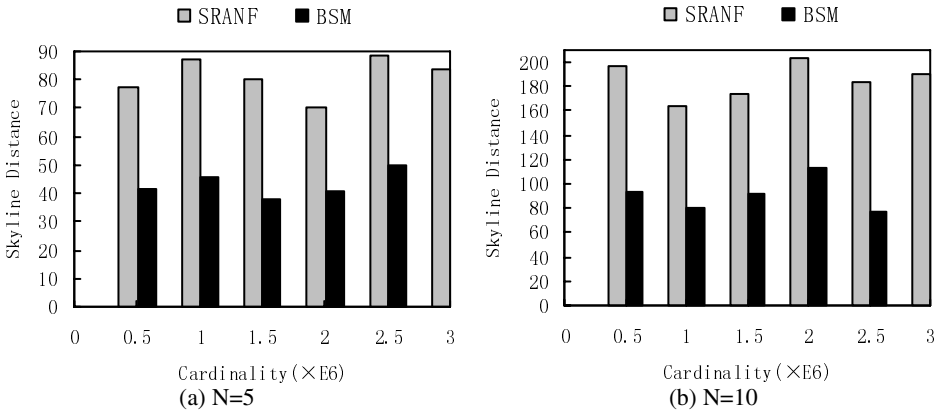


Fig. 2. Skyline Distance vs. Cardinality

From experiment 2, we observe that the skyline distance of tuples in the result set produced by *SRANF* is larger than the one produced by *BSM*. And it just conforms to our expectation. Since *SRANF* filters those tuples (i.e. skyline noises) which are similar to the ones which are the local optimality for users, the distances between the objects in the result set produced by *SRANF* are much larger than the one produced by *BSM*. In figure 2, we can see that the skyline distance produced by our algorithm is about 2.4 times larger than the one produced by *BSM*. Furthermore, if we increase the noise coefficient $T (\chi_1, \chi_2, \dots, \chi_N)$, the skyline distance will become larger.

3.3 Experiment 3: Comparing the Skyline Efficiency

Similarly, the database and the parameters which are used in this experiment are the same as the ones used in experiment 1. And in this experiment, we examine the total amount of time needed by each algorithm under different cases. Figure 3 shows the result of experiment 3.

In Figure 3 (a), we observe that runtime of *SRANF* algorithm almost equals to that of basic skyline method. However, when the number of dimensions increases, the distance of two curves will become larger (such as Figure 3(b)). This is since when the number of dimensions increases, the difference of the size of the result sets produced by these two algorithms increases gradually.

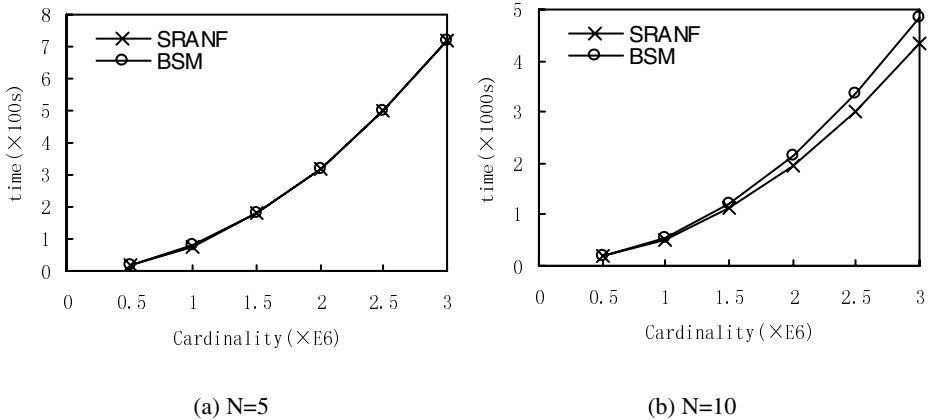


Fig. 3. Runtime vs. Cardinality

4 Conclusions

Skyline query processing, in particular, the efficiency of skyline query algorithms, has recently received a lot of attention. However, researchers often ignore that the skyline set will be beyond control in the applications which must deal with enormous data set. Consequently, it is not useful for users at all. Motivated by these facts, we propose a novel skyline reducing algorithm (i.e. *SRANF*) based on the acceptable difference between tuples in the original data set in this paper. *SRANF* algorithm adopts the technique of noise filtering. It filters skyline noises directly on the original data set based on the acceptable difference, and returns the objects which can not be filtered from the original data set. Furthermore, our experiment demonstrated that *SRANF* is both efficient and effective.

References

1. Borzsonyi, S., Kossmann, D., Stocker, K.: The Skyline Operator. *International Conference on Data Engineering ICDE*, 2001.
2. K.L. Tan, P.K. Eng and B.C. Ooi. "Efficient Progressive Skyline Computation", *VLDB*, 2001.
3. J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. "Skyline With Pre-sorting", *International Conference on Data Engineering ICDE*, 2003.
4. Kossmann, D., Ramsak, F., Preparata, F. P.: Shooting Stars In The Sky : An Online Algorithm For Skyline Queries. *VLDB*, 2001.
5. Papadias, D., Tao, Y., Fu, G., Seeger, B.: An Optimal And Progressive Algorithm For Skyline Queries. *SIGMOD*, 2003.
6. Balke, W-T., Guntzer, U., Zheng, J.X.: Efficient Distributed Skylining For Web Information Systems, *EDBT*, 2004.
7. Fagin, R., Lotem, A., Naor, M.: Optimal Aggregation Algorithms For Middleware. *PODS*, 2001.
8. Preparata, F. P., Shamos, M. I.: Computational Geometry: An Introduction. *Springer-Verlag*, 1985.
9. Bentley, J. L., Kung, H. T., Schkolnick, M., Thompson, C. D.: On The Average Number Of Maxima In A Set Of Vectors And Applications. *JACM*, 1978.

Modeling Information-Sharing Behaviors in BitTorrent System Based on Real Measurement

Jinkang Jia and Changjia Chen

School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing
100044, P.R. China
jinkangjia@yahoo.com.cn, changjiachen@sina.com.cn

Abstract. Logs of a BitTorrent (BT) site on campus are used to characterize the information sharing properties of BT users. The characteristics of (1) user's publishing and downloading behaviors, (2) the life span of the torrent and (3) user's downloaded, uploaded amount are addressed in the paper. Our main contributions are: (1) the distribution of users publishing characteristics has a flat head which is similar to that of user's information fetch behaviors, but cannot be explained by the well-known fetch-at-most-once principle. An indirect selection model is proposed to explain the phenomenon. (2) The lifetime y of a torrent file is tightly correlated with user's interests x (the number of fetches) (3) Distributions of user's downloaded amount and uploaded amount are essentially different. Approximately the former is exponential but the latter is power-law. The Cobb-Douglas like utility (CDLU) function is applied to study the relationship between them and a simple bound is found in user's CDLU.

1 Introduction

BitTorrent (BT) [1] is the most popular P2P information sharing system on Internet nowadays. As an unofficial statistics shows, BT has accounted for more than half of total Internet traffic in China recently, and there are millions of users who use BT to exchange files all over the world. The application has exploded rapidly in short period since it appeared in 2001, and it should be a vital research field for researchers on Internet. But contrary to prevalence of the BT's usage, the number of papers which has been published on BT is relatively small, and we guess that the main reason for the hindrance of widely and deeply research on BT is lack of the support of the real data. Due to various reasons the administrators of BT websites are unwilling to share their data for research. In order to overcome these difficulties, we constructed our own BT server on our campus to collect data and persuaded administrators of another BT station which is in operation a little earlier than ours to share its database. We extracted the log files lasting for about 200 days from the database.

Based on analyses of the real data we put the research emphasis on the description of the information sharing characteristics in BT system. Our main contributions are: Firstly, we try to depict how BT users publish files and choose interested files to download. We cannot observe the power-law derivate mechanism of monkey typing [7] in publishing behaviors, but we do find there appears obvious fetch-at-most-once like phenomenon no matter in the number of published files or in the amount of bytes

published by users. These phenomena are hard to be explained by FAMO which was first discovered in the measurement of KaZaa [6]. Therefore, the indirect selection principle is proposed to explain these phenomena. Secondly, we study the relationship between the life span of the torrent files and the behaviors of BT users, and it is revealed there exists obvious positive correlation between them, which is coincident with the conclusions in [3]. Finally we present the characteristics of the sharing ratio and uploaded/downloaded amounts statistics from our data. We try to adopt the theory of public goods in economics to analyze the relationship between uploaded bytes and downloaded bytes, and get the CDLU function of them.

The remainder of this paper is structured as follows. Some related work on BT is presented in Section 2. In section 3 we will introduce BT and describe our datasets. Base on analyses on users' behaviors of four kinds we show indirect selection and fetch-at-most-once characteristics in section 4. Then we discuss the characteristics of torrent life span and the uploaded/downloaded amount in section 5 and 6 respectively. Finally some conclusions and discussions are presented in Section 7.

2 Related Work

As far as we know, the research work on BT can be classified into two categories: one contains measurement and modeling work which try to reveal characteristics of BT networks [2-5, 13]. A fluid model is introduced to model the population evolution of the torrent and to reveal the performance in steady state in [2]. In [3] the authors try to depict one torrent's life span through the modification of the users' arrival distribution based on real measurement, and the cooperation among different torrents is raised for prolonging each torrent's life. In [13] based on rigorous mathematic deduction we provide improved deterministic models to describe the torrent evolution, downloading process of peers, and the generating process of seeds as well. While in some other early works [4, 5] some statistics or some phenomena are found empirically, such as the peers' evolutional curves, the serious flash crowd phenomenon, data availability and integrity, and the distribution of upload/download rate of peers, etc.

The other category of research pays great attention on the improvement of BT performance [10-12]. Free-riding and incentives are the problems mostly addressed. In [10] the effect of tit-for-tat algorithm is shown by experiments and two community specific mechanisms in use are proved to be good external incentives to promote cooperation. In [11] the iterated prisoner's dilemma in game theory is adopted for modifications of the TFT, and it's shown the new mechanism is more robust against free-riders. Recently in [12] the authors make comprehensive analysis on performance through simulation-based study and the performance of BT system is evaluated from several vital metrics (upload link utilization, fairness among peers). A tracker-based change and piece-based TFT policy are proposed to improve fairness among peers.

3 BT Mechanism and Our Datasets

A typical BT system is composed of three components: a website for publishing files, a tracker and lots of users. The files published on website are the small-sized torrent files

Table 1. Data statistics on different categories

	▽	▷	◀	△	○	✱	+	•	×	□	◇	☆	⊗	▽	▷
# of files	648	482	462	334	77	693	628	220	389	302	380	215	202	127	30
Avg. size(MB)	991	2800	349	554	102	669	1118	915	1710	1163	1106	2241	326	830	1371
Max. size(MB)	39177	18455	7414	8022	794	34005	13804	30379	34994	7614	18807	29158	4428	7981	5832
Min. size(KB)	1	27903	5	3	235	11	35	1765	17	19	1	4038	201	89	362
Total size (GB)	642	1349	161	185	7	464	702	201	1007	351	420	481	65	107	41

▽	▷	◀	△	○	✱	+	•	×	□	◇	☆	⊗	▽	▷
Chinese Movie	Chinese TV	Software	Chinese Music	Documents	Talk Shows	Games	Sports	Cartoons	Education	Foreign Movie	Foreign TV	Others	Foreign Music	Reseedings

Fig. 1. Icons we adopt to represent different categories in the paper

which only contain the meta-info of the real resource files to be shared. A torrent file which includes the tracker’s URL and the INFOHASH of the resource file is made and uploaded by the publisher. A tracker is a central server which is responsible for recording all users’ IP addresses and helping them find each other. After the torrent file is published on the website, the publisher runs his client to connect to the tracker and becomes the first seed of the torrent. Other users browse the site and find files in which they are interested. On downloading the torrent file users will run their clients for users’ list from the tracker and start downloading of the real resource file.

There are four kinds of users’ behaviors our data recorded, and they are publishing torrent files, browsing web pages which briefly introduce the contents of the resource files, downloading torrent files and downloading resource files. It should be noted that users can download torrent files from the links which web pages supply to users or directly from the file directory of the website. That is, it isn’t necessary for users to browse the web page to download torrent file. Our data contain the publisher of each file, number of times each page was browsed and each torrent file was downloaded, and number of completed times for each resource file.

There are two datasets of the BT log files which we use in the paper. The first dataset records all the information on the published files between Mar 24, 2005 and Oct 22, 2005. There are 5389 items in total and each item records the information of one file published. The information includes the ID of each web page, the filenames of the torrent file and the resource file, the category and size of the resource file, the publishing and ending time of torrent files, and some statistics on behaviors which are mentioned in last paragraph. Here ending time refers to the time when a torrent file is downloaded for the last time. The second dataset records each user’s IP address and total uploaded/downloaded bytes as well. There are totaling 10111 records in this dataset which is collected from Oct 9, 2004 to Oct 22, 2005.

There are 5389 files and 7858 users involved in our datasets. Among all the users there are 1543 publishers, which indicates only about one fifth of the users are willing to publish files for sharing. According to the contents of these files, they are classified into 15 categories upon publishing. We present the statistics on each category in Table 1 and show the icons that we adopt to differentiate various categories in Fig.1.

4 Indirect Selection and FAMO Characteristic of Users' Behaviors

Generally speaking, the selection behavior of users follows power-law distribution which has been proved by many works on network measurements. After the analyses of our data, we find much more than power-laws in users' behaviors. For example, it is revealed that the interests of BT users are broad-spectrum, which may help us in making assumptions about how files are selected by a single user when we try to explain the power-laws of the mass behaviors. In [7] we assume that all users have identical interests' spectrum, and each user independently selects files according to the same power-law. However, the random selection of exponentially-distributed interested users also results in power-law distribution if the size of the object groups is exponentially distributed, too. This has been proved by the model of monkey typing in [8]. From our analysis, we confirm that the broad-spectrum interests' assumption in [7] is acceptable. As a result, the mechanism of monkey-typing cannot be observed in the publishing behaviors of BT users.

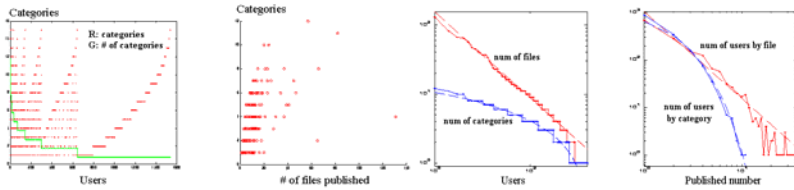


Fig. 2. Characteristics of the number of files or categories published by BT users are plotted. We name these four graphs as (a)~(d) from left to right. Details are described in the text.

It should be pointed out that it is shown the obvious deviation from power-laws in both users' downloading and publishing behaviors from our data. The former can be explained by FAMO but the latter cannot. An Indirect Selection Principle (ISP) is proposed to explain the deviation in users publishing behavior. We believe that ISP will play an important role to explain the flat head phenomenon in power-laws as FAMO does.

4.1 Broad-Spectrum Characteristics of Interests

We cannot study the interests of downloading behaviors because of deficiency of real data. But our datasets do contain the behaviors of the publishers, so we turn to study the publishing behaviors. We think it is reasonable that the interests of downloading correlate with those of publishing. We plot the publishing statistics in Fig.2. Fig.2 (a) shows the category distribution published by each publisher. All publishers are sorted in decreasing order by the number of categories they published, and each red point denotes the publisher in column has ever published some files of the category in row. The green line reveals the total number of categories which each publisher published. It can be observed that the interests of users are broad enough and more than 42% of users published files in at least two categories. Fig.2 (b) shows the diversity of the number of files a publisher published in different categories. A dot in the figure indicates there is a

publisher who has issued the number of files (in row) in the category (in column). In Fig.2(c)(d) we present the relationship between publishers and the number of files or categories. The curves in Fig.2(c) are sorted by the number of files (or categories) in decreasing order and Fig.2(d) shows the distribution of the number of categories (blue line) or files (red line) on the number of users. The curves in (d) tell us how many users published given number of files or categories. The solid lines in Fig. 2(c) and (d) are drawn from the real data and the dotted ones from curve fitting. The blue curves associated with categories are exponentially fitted and the red curves associated with files are fitted linearly in log-log base. Therefore, we can conclude that the number of files follows power-law distribution while the number of categories is exponential distributed. It can be observed the flat head phenomenon if one looks at the distribution of the number of files (curve in red solid line on Fig. 2(d)) carefully. We will try to give it a reasonable explanation in next subsection.

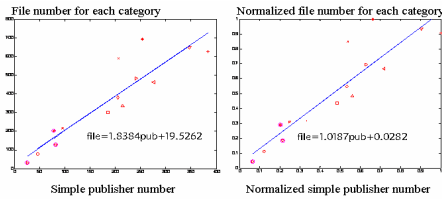


Fig. 3. Relationship between number of files and number of simple publishers

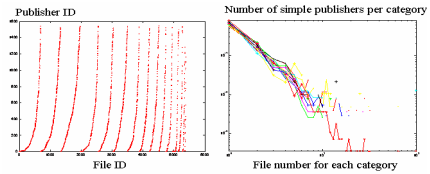


Fig. 4. Publishing behavior of simple publishers for each and all categories

4.2 Indirect Selection in Publishing Behaviors

Almost all literatures published so far on one user’s selection behavior assume the user selects objects directly. We think this model isn’t fit for BT users in publishing their files. The reason why a user is willing to publish a file for sharing is mainly its content but is not its size. Put it in other words, the file size a user publishes is the result of his indirect selections. We propose the indirect selection principle (ISP) because it can provide a good explanation for flat head phenomenon in file publishing behaviors which we mentioned in last subsection.

Following discussions above, it tells us from Fig.2 the distribution of the number of categories published can be more easily and precisely modeled than that of files published. So we will introduce ISP which starts with the selection of categories to describe the file publication process.

The idea of ISP goes as follows: although each publisher issues different number of files in different categories, we think it is reasonable to assume that the distributions of published file number are almost the same for each category. That is, a publisher needs only to care he will publish files in how many categories instead of in which categories. Therefore, we propose a new concept of simple publisher who is only responsible for determining the number of files published in one category according to the same distribution function. In other words, a simple publisher is an abstract of the representation of (publisher, category) pair, the number of files published by simple

publisher quantitatively equal to the number of files a publisher issued in the category. For example, a publisher named *Alice* has issued 3 files in “talk shows” and 5 files in “sports” can be modeled with two simple publishers, one is the 3-file (*Alice*, “talk shows”) and the other is the 5-file (*Alice*, “sports”). In our datasets, there are in total 1543 publishers who issued files in 15 categories, which results in 2772 (publisher, category) pairs or simple publishers.

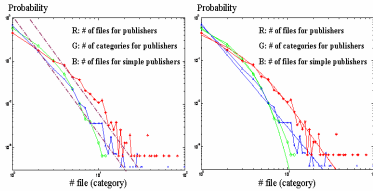


Fig. 5. Indirect Selection Principle (ISP) for number of files issued by publishers

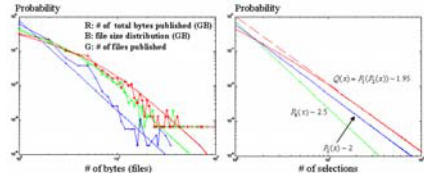


Fig. 6. Other applications of ISP. The left is for number of bytes published, and the right is for two exact power-laws.

We will use Fig.3 and 4 to argue the feasibility of the simple publisher assumption. Fig. 3 shows the file number published in different categories. It reveals that the number of files in each category is proportional to the number of simple publishers of that category. From Fig. 4(left) it can be observed how a publisher chooses categories and files to publish. Each trace in the figure stands for one category and all of these 15 traces have similar shape, which means that publishers follow similar rules to choose files in different categories. The above statement can also be proved by Fig. 4(right). In the figure, the distributions of the number of files on the number of simple publishers in each category are drawn separately. There are 16 traces in total and one of them is the distribution of the number of files on the number of all (publisher, category) pairs. All these traces largely overlap, which indicates that the fraction of simple publishers who published given number of files is independent of the category.

We redraw Fig.2 (d) in Fig. 5(left) and plot the distribution of the number of files on the number of all (publisher, category) pairs as well. It can be observed these three curves are all power-law like, but the head of former two are much flatter than that of the latter. As we discussed above, the probability of the number of categories one publisher chooses follows exponential while that of files issued by a simple publisher follows power-law. We will adopt ISP to explain why an exponential selection followed by power-law selection will result in a flat-head power-law like distribution. For the convenience of our description, we denote them as following:

$$\begin{aligned} \text{Probability of categories a publisher selects: } P_c(n) &\sim e^{-\alpha} \\ \text{Probability of files a simple publisher selects: } P_f(n) &\sim n^{-\beta} \end{aligned}$$

Because every file can be published only once, FAMO cannot explain the flat head which appears in the distribution of the number of files a publisher issues. Therefore, we raise the principle of indirect selection (ISP) here to explain the phenomenon and the analytic expression can be deduced. In ISP model, the publisher firstly decides how many categories will be issued by the simple publisher according to $P_c(n)$, and the simple publisher then independently selects the number of files according to $P_f(n)$.

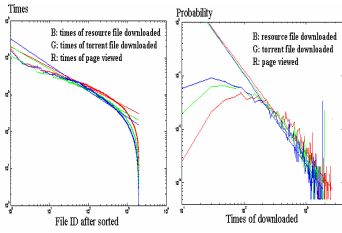


Fig. 7. FAMO phenomenon in BT users' fetching behaviors

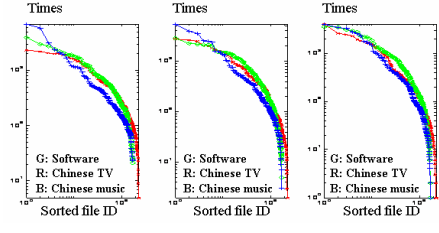


Fig. 8. Fetching characteristics for three representative categories

Finally the publisher will publish all files which have been chosen by all the simple publishers. Thus $P_F(n)$, the probability that a publisher issues n files is consequences of above two selections. To get a tight expression for probability $P_F(n)$, the generation function is a simple tool. We define the generation function for a given probability $P(n)$ is $Q(x) = \sum_n P(n)x^n$. We will use $Q_c(x)$, $Q_f(x)$ and $Q_F(x)$ to denote the generation function of $P_c(n)$, $P_f(n)$ and $P_F(n)$ respectively. Then the generation function of $P_F(n)$ can be simply written as:

$$Q_F(x) = \sum_n P_c(n)(Q_f(x))^n = Q_c(Q_f(x)) \tag{1}$$

Both the distribution predicted by Eq.(1) and the real measured distribution are drawn in Fig. 5(right). They fitted well beyond our expectation. In the calculation, we take $\alpha = 0.79$ and $\beta = 2.4$ which are the fitting results of the other two distribution curves based on real measurement. We believe ISP is a proper model to describe the publishing behavior and to explain the flat head phenomenon.

As another application of ISP, we consider the total bytes published by publishers. Fig. 6(left) reveals the distribution of the total bytes by publishers is also a power-law like curve with a flatter head. ISP can model this process more natural. Publishers concern more about the content of files they publish and take less notice to the size of those files. And studies have shown that the size of a file is distributed according to power-law $P_S(n) \sim n^{-\gamma}$. Then the bytes a publisher issues will be the consequence of the file selection rule and the distribution of the file size. Therefore, the generation function of total bytes a publisher issued can be expressed as

$$Q_B(x) = Q_F(Q_S(x)) \tag{2}$$

$Q_B(x)$, $Q_F(x)$ and $Q_S(x)$ are the generation functions of the distribution of total bytes issued, file number issued and the file size respectively. In fig. 6(left) we draw these three real measured distributions and their fitting curves as well. We take $\gamma = 2.6$ here which is the fitting result of file size distribution. Previously calculated $Q_F(x)$ by (1) is also used to calculate the generation function $Q_B(x)$ in (2). The predicted curve by ISP also fits well with the real measured bytes distribution curve in this case.

A more general example is provided in Fig. 6(right). It shows how flat the head can be resulted by ISP with consecutive two exact power-law selections. Even though this example is not fit for our measured data, it may be helpful to other applications or fields

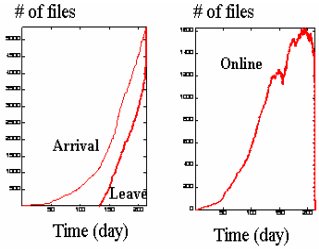


Fig. 9. Arrival, leave and online processes for torrent file

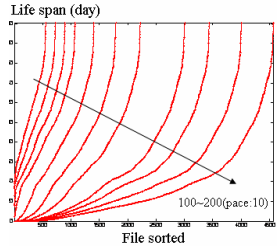


Fig. 10. The life span for arrivals in different intervals

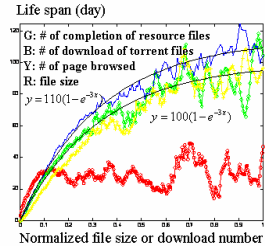


Fig. 12. Relationships between four factors and life span of the files

since many studies have shown that power-law is a universal rule in nature and in human behaviors as well.

4.3 Fetch-At-Most-Once (FAMO) Behavior

It has become one of the “modern classical” conclusions of the network theory that “everything on web is power-law”. However, the paper which first challenges the conclusion is [6]. Based on the analysis and modeling of the measurements of KaZaa network, it was revealed that the fetched times of the files in KaZaa obviously deviate from power-law, especially for popular files. And the paper presented the explanation of FAMO. Unlike web pages, the files involved in P2P network are almost large files and invariable. It’s enough to download only once for each file. So for popular files the number of the downloaded times will obviously smaller than that of power-law.

We present the FAMO phenomenon observed in our datasets in Fig. 7. It is shown that there exist obvious FAMO no matter what type of operations is considered: the times of page browsed, the torrent file downloaded or the resource file completed. As for each category, we choose three typical categories from all 15 categories: “Chinese TV”, “software” and “Chinese music”, which can represent different scale of average downloaded times. The user’s fetching behaviors in these three categories are plotted in Fig. 8. And we can observe that the FAMO phenomenon looks more apparent for the category with more downloaded times, which accords with the result of [7].

5 The Life-Span of Torrent Files

The publishers issue new torrent file occasionally on the website, and those who are interested in these files will begin downloading until the death of the torrent. So we define the life span of one torrent file as the period from the publishing of it to the time when the last downloader downloads this torrent file. Here we consider the accumulated number of torrent files published before given observation time as the arrival process. Similarly, the leave process is defined to be the accumulated number of torrent file which leaves the BT system. We also define the online process which is the difference between the above two processes. In fact the online process indicates the number of files alive at that time. We plot these processes in Fig. 9. It can be observed that the website developed very fast in first 100 days, and reached steady state after 150

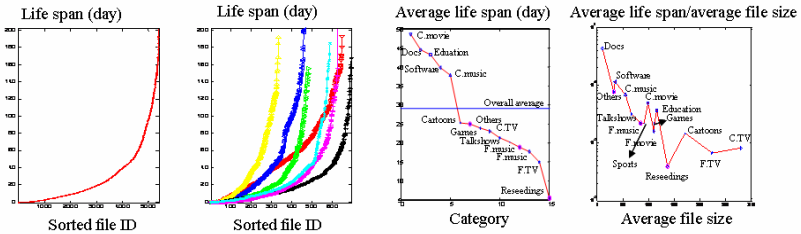


Fig. 11. The life span characteristics for torrent files are plotted. We denote them as (a)~(d) from left to right. It is depicted in (a) the overall life span distribution, whereas the remaining three graphs reveal the characteristics between life span and each category.

days. During the stable period, the arrival and leave processes can be approximately modeled as Poisson processes. The reason for the final decline of the curve is the edge effect of our datasets.

It’s hard to calculate the exact life span of a torrent file because there are many torrent files which in fact haven’t ended their lives when our data collection is ended. We will call it the edge effect of the measurement. Of course we can only consider the files published before some predetermined date to relieve this effect, but it’s hard to decide which date to choose. As a test, we consider the life span for files published between 100 and 200 days after the establishment of the website, and we group files at the pace of 10 days and calculate the life span respectively. In Fig. 10 we show the result. Surprisingly, it seems that the earlier the publishing of the files, the higher the ratio of the short-lived ones. Therefore, we adopt the assumption that all files end up with the finish of our data collection in the paper.

We plot the distribution of the life span for all and for each category in Fig. 11 (a) and (b) respectively. The distribution characteristics are almost the same no matter for overall or for each category. Next we plot the average life span for each category in Fig. 11(c). It’s obvious that all categories can be divided into two regimes according to the average life span for all files. We cannot give feasible reasons why there exist these two regimes except for the obvious explanation for the most short-lived “reseedings” category. It is plotted the relationship between the average life span and the average file size for each category in Fig. 11(d). The rough tendency is the file lives longer if the size of it is smaller. It seems to be a reasonable explanation that users aren’t eager to delete the files with small sizes from their hard disks because the relatively small spaces these files occupy.

Next we consider some other factors which may correlate more with the life span of a torrent file. These factors include: total times of the associated web page browsed, total times the torrent file downloaded, total users who downloaded the associated resource file and the size of the resource file.

Since the life spans fluctuate sharply, the averaged life span in certain windows will reveal more meaningful results than full information will do. Therefore, firstly we will partition torrent files into different groups. Taking the associated web page as an example, we group these web pages browsed by similar number of users together and calculate the average life span and average number of times browsed respectively. The same method is also adopted to process the other three factors. Finally a simple low

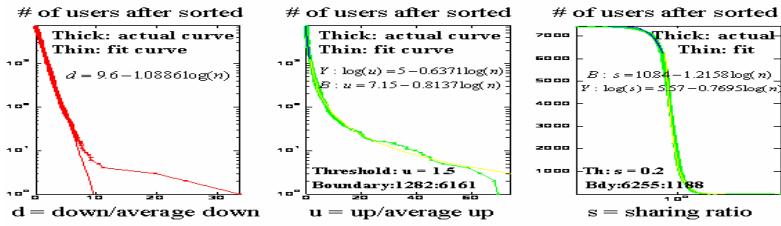


Fig. 13. Uploaded/Downloaded amount and sharing ratio distribution

pass filter is used to smooth these curves. Except for the resource file size, we group the other three factors which are related with times by a granularity of 10. i.e., The k th group includes all the objects which are browsed or downloaded by the number of users between $10(k-1)+1$ and $10k$. Similarly, for resource file size, we take 30 Mega bytes as the granularity. The coefficient for the low pass filter we used here is 0.9. All the results are depicted in Fig. 12. It can be observed that the average life span of the torrent files is increased with the increase of its popularity to users. However, it seems there doesn't exist obvious correlation between the life span and the size of resource file, except when the file size is rather small.

In conclusion, the relation between the life span of files and the normalized amount of browsing or downloading can be expressed by a very simple formula. Denoting the life span as x and the normalize amount fetched as y , we get the following expression:

$$y = K (1 - e^{-3x}) \tag{3}$$

To our surprise, the expression coincides with results in [3] perfectly. K here refers to the maximal life span for all files. In the figure, K is 100 days for page browsed and the resource file downloaded, and 110 days for the torrent file downloaded. For we care more about the life span of the files, we rewrite (3) and get the inverse function:

$$x = 0.33 \ln(1 - y/K) \tag{4}$$

6 Sharing Ratio and Uploaded/Downloaded Amount

The downloaded amount by each user can be regarded as the profit which the user makes by means of BT file-sharing system, whereas the uploaded amount is the contribution which the user makes to the system. The natural restriction is that one's downloaded amount comes from others' uploaded amount, and vice versa. Therefore, the total uploaded amount by all users must be equal to the total downloaded amount. Another important metric that one BT system is concerned about is the sharing ratio of each user, which is defined as the ratio of total uploaded amount to downloaded amount for each user. Because most users in BT are selfish and are likely to leave the system once they finish downloading. Though the tit-for-tat algorithm guarantees the sharing ratio to some extent, it works weakly in forcing users to reach an ideal sharing ratio. In the paper we will present some results that can be drawn through the analysis of our datasets.

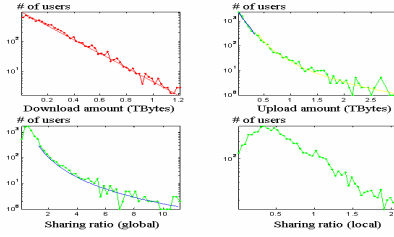


Fig. 14. The distribution of U/D amount and sharing ratio

Table 2. The U/D percentage for different sharing ratios

Sharing ratio	Down%	Up%	Users%
$r > 1$	29.14%	65.1%	24.3%
$1 \geq r > 0.5$	36.43%	24.33%	29.61%
$0.5 \geq r > 0.2$	28.91%	9.92%	31.48%
$0.2 \geq r > 0$	5.52%	0.65%	14.68%

We plot the uploaded/downloaded and sharing ratio distribution in Fig. 13. It can be observed that the downloaded amount follows exponential distribution. While the uploaded amount and the sharing ratio are only exponential at the interval of small value and power-law at the other intervals, which is shown by the fitted curve colored blue and yellow respectively. The same conclusions can be made from Fig. 14. Note there appears a maximum at 0.34 and a flat interval between 0.2 and 0.5 from the PDF of sharing ratio. From CDF curve of the sharing ratio in Fig. 15, we can observe that almost one third of users fall into the flat interval between 0.2 and 0.5. We guess the interval is the limit that tit-for-tat mechanism can guarantee to reach because it take only several minutes for campus users to finish downloading, and most users quit the system once finished. We haven't definite proof of it yet. A possible explanation is that the net downloading ratio (downloading ratio - uploading ratio) for interval between 0.2 and 0.5 is accidentally 20% which is just the quantity reported by many papers that the BT system can endure for free riders. And also from Table 2, we can approximately look on 0.5 to be the threshold for judging if a user is a free-rider.

Most recent works on P2P information sharing adopt the theory of public goods in economics [9]. And the Cobb-Douglas Utility function (CDU) is the utility function most authors applied. If we denote the user's downloaded amount as sd_i and uploaded amount as su_i , A Cobb-Douglas Like Utility (CDLU) function can be written as:

$$U_i = \alpha_i \ln sd_i + \beta_i \ln su_i \tag{5}$$

We call it CDLU function since we do not force $\alpha_i + \beta_i = 1$ as the CDU required. The economic behavior of BT users can be represented to maximize U_i . We plot the users' CDLU characteristics in Fig. 16. All (uploaded, downloaded) pairs are concentrated around a straight line in log-log base. This line can be expressed as:

$$\ln su_i = 1.6 \ln sd_i - 15.6 \tag{6}$$

Furthermore, most users' (uploaded, downloaded) pairs fall within the area bounded by following two lines:

$$\ln su_i = 1.6 \ln sd_i - 15.6 \pm 2.5 \tag{7}$$

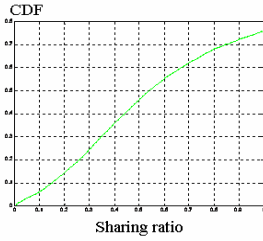


Fig. 15. CDF of sharing ratio

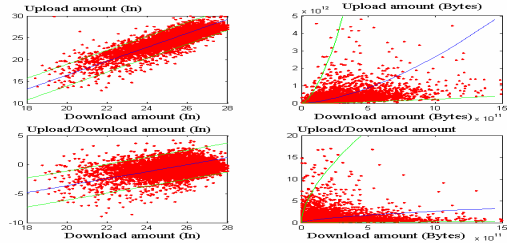


Fig. 16. CDU properties of U/D amount and sharing ratio

Since the sharing ratio is defined as $sr_i = su_i / sd_i$, we get following expression about the sharing ratio vs. downloaded amount immediately

$$\ln sr_i = 0.6 \ln sd_i - 15.6 \pm 2.5 \quad (8)$$

7 Discussions and Conclusions

In the paper we analyzed the log file of one BT website on campus and revealed many characteristics on information sharing behaviors. We discussed the following three aspects: the publishing and fetching behavior of users, the life span of torrent files, and the uploaded/downloaded and sharing ratio properties. Our main conclusions are:

- There appears FAMO phenomenon in the downloading behavior of users.
- The publishing behavior can be modeled by indirect selection principle (ISP). It is proved by both our dataset and simulation that the ISP can accurately describe the flat head phenomenon in measured curve, which is similar to that resulted from FAMO.
- The life span of the torrent file increases with the increase of the fetching activities related files, and it can be modeled by a simple one order expression. But there isn't obvious correlation observed between the life span and the size of resource file.
- The uploaded amount differs greatly from the downloaded amount, and the former follows exponential distribution while the latter follows power-law. And the U/D characteristics are distinct significantly for different users whose sharing ratios fall in different intervals specified by 0.2, 0.5 and 1. Finally, we considered CDU of U/D amount, which can be expressed by a set of simple linear relations.

References

1. Cohen, B.: Incentives Build Robustness in BitTorrent. In Workshop on Economics of Peer-to-Peer System. Berkeley, USA, May 2003. <http://www.bittorrent.com/>
2. Qiu, D., Srikant, R.: Modeling and Performance Analysis of BitTorrent-Like Peer-to-peer Networks. In Proc. ACM Sigcomm04', Aug. 2004.
3. Guo, L et al.: Measurements, Analysis, and Modeling of BitTorrent-Like Systems. In Proc. of Internet Measurement Conference, Oct 2005.
4. Izal, M et al.: Dissecting BitTorrent: Five Months in a Torrent's Lifetime. In Proc. 5th annual Passive & Active Measurement Workshop, Apr. 2004,

5. Pouwelse, J.A., Garbacki, P et al.: The BitTorrent P2P File-sharing System: Measurements and Analysis. In Proc. 4th International Workshop on Peer-to-Peer Systems, Feb. 2005
6. Gummadi, K.P., Dunn R.J., Saroiu, S., Gribble, S. D. et al: Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload. In SOSP'03.
7. Liu, Z. and Chen, C.: Modeling Fetch-at-Most-Once Behavior in Peer-to-Peer File-Sharing Systems. Lecture Notes in Computer Science, Vol 3842. Springer-Verlag, Berlin Heidelberg (2006) 717-724.
8. Conrad, B., Mitzenmacher, M.: Power Laws for Monkeys Typing Randomly: The Case of Unequal Probabilities. IEEE Transactions on Information Theory, Vol. 50, Jul, 2004
9. Gu, B., Jarvenpaa, S: Are Contributions to P2P Technical Forums Private or Public Goods?-An Empirical Investigation. In WEPPS,2003
10. Andrade, N., Mowbray, M.: Influences on Cooperation in BitTorrent Communities. In ACM Sigcomm P2P ECON Workshop 2005, Aug, 2005
11. Jun, S., Ahamad, M.: Incentives in BitTorrent Induce Free Riding. In ACM Sigcomm P2P ECON Workshop 2005, Aug, 2005
12. Bharambe, A.R., Herley, C.: Analyzing and Improving a BitTorrent Network's Performance Mechanisms. In IEEE Infocom 2006. Apr. 2006.
13. Liu, Z. and Chen, C.: Modeling BitTorrent-like Peer-to-Peer Systems. To appear in the IEEE Communications Letters, 2006.

Financial Distress Prediction Based on Similarity Weighted Voting CBR*

Jie Sun and Xiao-Feng Hui

School of Management, Harbin Institute of Technology, Harbin 15 00 01
HeiLongJiang Province, China
sjhit@sina.com
xfhui@hit.edu.cn

Abstract. Financial distress prediction is an important research topic in both academic and practical world. This paper proposed a financial distress prediction model based on similarity weighted voting case-based reasoning (CBR), which consists of case representation, similar case retrieval and combination of target class. An empirical study was designed and carried out by using Chinese listed companies' three-year data before special treatment (ST) and adopting leave-one-out and grid-search technique to find the model's good parameters. The experiment result of this model was compared with multi discriminant analysis (MDA), Logit, neural networks (NNs) and support vector machine (SVM), and it was concluded that similarity weighted voting CBR model has very good predictive ability for enterprises which will probably run into financial distress in less than two years, and it is more suitable for short-term financial distress prediction.

1 Introduction

With the globalization of world economy and the variety of customer demand, enterprises which fail to struggle in the competitive business environment might bankrupt. Bankruptcy not only brings much individual loss to interest parts such as stockholders, creditors, managers, employees, etc., but also too much bankruptcy will greatly shock the whole country's economic development. Generally, most enterprises that ran into bankruptcy had experienced financial distress, which usually have some symptoms indicated by financial ratios. Hence, it is both significant and possible to explore effective financial distress prediction models with various classification and prediction techniques, so that enterprise managers can foresee the financial distress and take effective actions to avoid deterioration of financial state and bankruptcy.

There exist abundant literatures with research methodology typically ranging from statistical ones such as univariate analysis, multi discriminate analysis (MDA), and Logit, to machine learning methods such as neural networks (NNs) and support vector machine (SVM). Beaver (1966) investigated the predictability of 14 financial ratios

* This paper is supported by the National Natural Science Foundation of China (No. 70573030) and the National Center of Technology, Policy, and Management, Harbin Institute of Technology.

using 79 pairs of failed and non-failed sample firms [1]. Atman (1968) used MDA to identify companies into known categories and concluded that bankruptcy could be explained quite completely by a combination of five (selected from an original list of 22) financial ratios [2]. Ohlson (1980) was the first to apply Logit model to predicting financial distress [3]. NNs is the most widely used machine learning method in this field, and Odom and Sharda (1990) made an early attempt to use it for financial distress prediction [4]. SVM, which is a relatively new machine learning technique, were applied to bankruptcy prediction respectively by Shin and Min (2005) with Korean data [5][6], followed by Hui and Sun (2006) with data of Chinese listed companies [7].

Case-based reasoning (CBR) is an important artificial intelligence reasoning methodology, which was first put forward by Prof. Schank and his colleagues at Yale University [8]. The basic model of CBR cycle consists of case retrieval, case reuse, case revision, and case learning. CBR is widely used to solve new problems by learning from past analogous cases, especially in a decision environment where rule-based domain knowledge is hard to acquire, for example medical diagnosis, traffic accident disposal and legal inference [9]. However, there are only a few studies which have applied CBR to classification and prediction problems. Hansen et al. (1992) applied CBR to the auditor litigation problem [10]. Kim and Noh (1997) predicted interest rate with CBR and NNs [11]. Jo et al. (1997) used a case-based forecasting system, NNs and MDA to predict bankruptcy, and concluded that NNs performed best [12].

The contribution of this paper is to propose a new financial distress prediction model based on similarity weighted voting CBR, which integrates the theory of CBR and the combination thought of weighted voting. It not only broadened the application of CBR but also enriched the methodology system of financial distress prediction. The empirical experiment with data of Chinese listed companies further indicated the effectiveness of this new model for short-term financial distress prediction. The rest of the paper is divided into three sections. Section 2 describes the theory of financial distress prediction model based on similarity weighted voting CBR in detail. Section 3 is the empirical experiment which consists of four parts, i.e. experiment design, data collection and preprocessing, grid-search and empirical range of parameters, and experiment results and analysis. Section 4 makes conclusion.

2 Financial Distress Prediction Model Based on Similarity Weighted Voting CBR

2.1 Case Representation and Case Base

For financial distress prediction, cases are in the form of enterprises with healthy or distressed financial state, which is denoted by u . Many cases constitute an initial enterprise case base, denoted by $U = \{u_i\} (i = 1, 2, \dots, m')$. Each case is described with a set of qualitative or quantitative features, which are categorized into three types:

- (1) Qualitative factor features, denoted by $G = \{g_j\} (j = 1, 2, \dots, n')$. For enterprise cases, they are often in the form of enterprise name, economic category, industry, and so on.

- (2) Quantitative factor features, denoted by $V = \{v_j\} (j = 1, 2, \dots, n)$. In the case of financial distress prediction, they should be various financial ratios.
- (3) Class feature, denoted by $D = \{d\}$. If enterprises are categorized into q classes according to their financial states, then the class feature d will have q possible values, i.e. class set $C = \{c_1, c_2, \dots, c_q\} = \{c_l\} (l = 1, 2, \dots, q)$. For example, if enterprises are categorized into two classes according to their financial state, the possible values of feature d are $\{Normal, Distressed\}$.

Then the case model expressed through characteristic features is as follows:

$$u = \text{Case}(G, V, D) = \text{Case}(g_1, g_2, \dots, g_n, v_1, v_2, \dots, v_n, d) \quad (1)$$

By collecting information of financially distressed enterprises and corresponding financially healthy enterprises with the same industry and matching asset scale, case base for financial distress prediction can be built up by the technique of relational database. Through relational database management system, case management and maintenance operations such as adding, deleting and altering can be carried out.

2.2 Similar Case Retrieval for Financial Distress Prediction

Case selection and retrieval is usually regarded as the most important step in the whole CBR cycle, directly influencing the validity and efficiency of the CBR system. Since one basic assumption of CBR is that similar experiences in the past can act as guidance for future reasoning, problem solving and learning, most case retrieval strategies are based on the concept of similarity, which can be defined as the measure of analogy or similarity between target case and stored cases. In the case of financial distress prediction, this paper adopts a case retrieval method integrating knowledge guided strategy and k -nearest neighbor principle and a similarity computation method based on the combination of grey system theory, fuzzy set theory and concept of distance.

Knowledge Guided Primary Case Retrieval. According to the domain knowledge of financial distress prediction, primary case retrieval can be carried out in terms of values of certain qualitative factor features. Because financial distress prediction case base is built by relational database technique, SQL query language can be applied to primarily retrieve useful cases [13]. For instance, in the light of industry feature (assumed to be represented as g_3), in order to primarily retrieve cases which belong to the same industry as target case, retrieval sentence is as follows.

*SELECT * FROM case base WHERE g_3 =industry of target case.*

Advanced k -Nearest Neighbor Case Retrieval Based on Similarity. Suppose the case set primarily retrieved from the initial case base U' is represented as $U = \{u_i\} (i = 1, 2, \dots, m)$, and the quantitative factor feature set composed of various financial ratios is denoted by $V = \{v_j\} (j = 1, 2, \dots, n)$, then the values of all cases' quantitative factor features constitute a case-feature matrix, represented as F' .

$$F' = \begin{bmatrix} f'_{11} & f'_{12} & \cdots & f'_{1n} \\ f'_{21} & f'_{22} & & f'_{2n} \\ \vdots & & \ddots & \\ f'_{m1} & f'_{m2} & & f'_{mn} \end{bmatrix} = (f'_{ij})_{m \times n} \tag{2}$$

In which, f'_{ij} is the j -th quantitative factor feature value of case i .

To eliminate the effect of different dimensions of quantitative factor features and make them comparable, the initial case-feature matrix F' should be standardized. Here the concept of relative fuzzy membership is adopted, and each element of matrix F' is divided by the maximum value of the column to which the element belongs. Thus the case-feature relative membership matrix F is obtained.

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & & f_{2n} \\ \vdots & & \ddots & \\ f_{m1} & f_{m2} & & f_{mn} \end{bmatrix} = (f_{ij})_{m \times n} \tag{3}$$

$$f_{ij} = \frac{f'_{ij}}{\max_j f'_{ij}} \tag{4}$$

In which, $\max_j f'_{ij}$ denotes the maximum value of column j in F' .

Suppose u_0 denotes the target enterprise case whose financial state is to be predicted, and f_{0j} represents the standardized relative fuzzy membership of u_0 's j -th quantitative factor feature. According to grey system theory, the grey correlation degree between target enterprise case u_0 and stored case u_i at quantitative factor feature v_j is defined as γ_{ij} .

$$\gamma_{ij} = \frac{\inf_j |f_{0j} - f_{ij}| + b \sup_j |f_{0j} - f_{ij}|}{|f_{0j} - f_{ij}| + b \sup_j |f_{0j} - f_{ij}|} \tag{5}$$

($i = 1, 2, \dots, m; j = 1, 2, \dots, n$)

In which, $\inf_j |f_{0j} - f_{ij}|$ and $\sup_j |f_{0j} - f_{ij}|$ respectively represent the minimum and maximum distance of relative membership at quantitative factor feature v_j between target enterprise case u_0 and all stored cases in case set U . $|f_{0j} - f_{ij}|$ denotes the distance of relative membership at quantitative factor feature v_j between target enterprise case u_0 and stored case u_i . $b \in [0, 1]$ is the environmental parameter. So it is clear that $\gamma_{ij} \in [0, 1]$, and $m \times n$ γ_{ij} constitute the grey correlation degree matrix R for target enterprise case's financial distress prediction [14].

$$R = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1n} \\ \gamma_{21} & \gamma_{22} & & \gamma_{2n} \\ \vdots & & \ddots & \\ \gamma_{m1} & \gamma_{m2} & & \gamma_{mn} \end{bmatrix} = (\gamma_{ij})_{m \times n} \quad (6)$$

According to the similarity concept based on weighted Euclidean distance, the similarity measure between target enterprise case u_0 and stored case u_i is defined as sim_{0i} .

$$sim_{0i} = 1 - \left\{ \sum_{j=1}^n [w_j (1 - \gamma_{ij})]^2 \right\}^{\frac{1}{2}} \quad (7)$$

The bigger sim_{0i} is, the more similar the target enterprise case u_0 and stored case u_i are. According to k -nearest neighbor principle, given the similarity threshold T , then stored cases whose similarity value with target enterprise case u_0 are bigger than or equal to T constitute a k -nearest neighbor case set of target enterprise case u_0 , denoted by U^* .

$$U^* = \{u_{i^*} \mid sim_{0i^*} \geq T\} (i^* = 1, 2, \dots, k) \quad (8)$$

In practice, the similarity threshold T is often set as a certain percentage (denoted by p) of the maximum similarity between target enterprise case u_0 and all stored cases.

$$T = p \max(sim_{0i}) \quad (9)$$

2.3 Combination of Target Class Based on Similarity Weighted Voting

Suppose u_{i^*} denotes one of the k -nearest neighbor cases selected by advanced retrieval. Its value of class feature d is represented as d_{i^*} , and its similarity with target enterprise case u_0 is represented as sim_{0i^*} . The similarity weighted voting probability of target enterprise case u_0 belonging to financial state class c_l is defined as $prob(c_l)$.

$$prob(c_l) = \sum_{d_{i^*=c_l} sim_{0i^*}} / \sum_{i^*=1}^k sim_{0i^*} \quad (l = 1, 2, \dots, q) \quad (10)$$

In which, $\sum_{i^*=1}^k sim_{0i^*}$ is the sum of similarity between target enterprise case u_0 and all k -nearest neighbor cases in U^* . $\sum_{d_{i^*=c_l} sim_{0i^*}}$ is the sum of similarity between target enterprise case u_0 and those k -nearest neighbor cases in U^* whose class feature value equal to c_l . Assume the financial state class feature of target enterprise case u_0 is denoted by d_0 , according to the combination principle of similarity weighted voting, the

final value of d_0 should be the class c^* , whose similarity weighted voting probability is the biggest among q classes in class set C .

$$d_0 = c^* \quad \text{if } \text{prob}(c^*) = \max_{l=1}^q (\text{prob}(c_l)) \quad (11)$$

3 Empirical Experiment

3.1 Experiment Design

- (1) Collect sample data from Chinese listed companies, and enterprise financial state is categorized into two classes, i.e. *normal* and *distressed*, according to the criteria whether the listed company is specially treated (ST)¹ by China Securities Supervision and Management Committee (CSSMC).
- (2) Suppose the year when a company is specially treated as the benchmark year ($t-0$), then ($t-1$), ($t-2$) and ($t-3$) respectively represent one year before ST, two years before ST and three years before ST. Use three years' data before ST to form three case sets, i.e. U_{t-1} , U_{t-2} , U_{t-3} , and three times of experiments are respectively carried out on them.
- (3) Restricted by sample number, knowledge guided primary case retrieval on the base of qualitative factor feature is skipped over, and the experiment directly starts from quantitative advanced retrieval.
- (4) Leave-one-out strategy is applied to validate the financial distress prediction model based on similarity weighted voting CBR: When doing experiment with data of year ($t-x$) ($x=1, 2, 3$), each time one sample is taken out from case set U_{t-x} ($x=1, 2, 3$) to act as target enterprise case and the rest samples act as stored cases. Then after N_{t-x} (N_{t-x} denotes the total sample number of case set U_{t-x}) times of prediction, each case's predicted financial state class is compared with its true financial state class and the leave-one-out accuracy of the similarity weighted voting CBR model for year ($t-x$) can be figured out.
- (5) Grid-search technique is used to obtain the optimal values for environmental parameter b and similarity threshold percentage p . By fixing one parameter value and changing the other parameter value, further experiment is carried out, trying to get the empirical value range of good model parameters.
- (6) For comparative study, the leave-one-out accuracy of similarity weighted voting CBR model is compared with those of MDA, Logit, NNs and SVM, which are the research results of the authors' former study based on the same sample data, so that validity of the model proposed in this paper can be better analyzed.
- (7) MATLAB 6.5 software and its program language are used to realize the whole experiment process.

¹ The most common reason that China listed companies are specially treated by CSSMC is that they have had negative net profit in continuous two years. Of course they will also be specially treated if they purposely publish financial statements with serious false and misstatement, but the ST samples chosen in this study are all companies that have been specially treated because of negative net profit in continuous two years.

3.2 Data Collection and Preprocessing

Collection of Initial Data. The data used in this research was obtained from China Stock Market & Accounting Research Database. ST companies are considered as companies in financial distress and those never specially treated are regarded as healthy ones. According to the data between 2000 and 2005, 135 pairs of companies listed in Shenzhen Stock Exchange and Shanghai Stock Exchange are selected as initial dataset. 35 financial ratios covering profitability, activity ability, debt ability and growth ability are selected as initial quantitative factor features.

Data Preprocessing. The preprocessing operation to eliminate missing and outlier data is carried out for the three years' data: (1) Sample companies in case of missing at least one financial ratio data were eliminated. (2) Sample companies with financial ratios deviating from the mean value as much as three times of standard deviation are excluded. After eliminating companies with missing and outlier data, the final numbers of sample companies are 75 pairs at year ($t-1$), 108 pairs at year ($t-2$) and 91 pairs at year ($t-3$).

Choice of Quantitative Factor Features. Companies at the different stages before financial distress usually have different symptoms which are indicated by different financial ratios. So aiming at improving the predictive ability, each year's quantitative factor features are selected from 35 original financial ratios by the statistical method of stepwise discriminant analysis. According to the sample data, the final quantitative factor features for year ($t-1$), ($t-2$) and ($t-3$) are listed in Table 1.

Table 1. Quantitative factor features of the three years before ST

Year	Quantitative factor features	
$(t-1)$	Total asset turnover	Asset-liability ratio
	Earning per share	Total asset growth rate
$(t-2)$	Account payable turnover	Current asset turnover
	Fixed asset turnover	Asset-liability ratio
	Return on total asset	Return on current asset
$(t-3)$	Return on equity	
	Current asset turnover	Fixed asset turnover
	The ratio of cash to current liability	Asset-liability ratio
	The proportion of current liability	

Standardization of Case-Feature Matrix. Three case-feature matrixes are respectively formed according to the final quantitative factor features of the three years before ST. They are standardized by the principle that each element should be divided by the maximum value of the column to which the element belongs, and three corresponding case-feature relative membership matrixes are obtained.

3.3 Grid-Search and Empirical Range of Parameters

Financial distress prediction model based on similarity weighted voting CBR mainly have two parameters to determine, i.e. the environmental parameter b for computing

grey correlation degree and the similarity threshold percentage p for determining k -nearest neighbor case set. So aiming at searching for proper values for these parameters, this study follows the grid-search technique which is used by Chih-Wei Hsu et al. to find good SVM parameters [15]. Trying growing sequences of b and p ($b=[0.1:0.05:1]$; $p=[0.6:0.01:1]$), leave-one-out accuracy for each possible pair of parameter values is iteratively calculated so as to find good pairs of parameter values which make the leave-one-out accuracy highest. For each year, a scatter diagram is drawn according to the pairs of parameter values which correspond to the two highest accuracies. As shown in Fig. 1, Fig. 2 and Fig. 3, year ($t-1$) has two optimal pairs of parameters (i.e. [0.2, 0.79] and [0.25, 0.82]); year ($t-2$) has five optimal pairs of parameters (i.e. [0.3, 0.94], [0.65, 0.95], [0.75, 0.95], [0.75, 0.96] and [0.9, 0.96]); year ($t-3$) also has two optimal pairs of parameters (i.e. [0.2, 0.84] and [0.45, 0.92]). In order to make clearer the empirical value range of good model parameters, by fixing the value of parameter b and changing the value of parameter p in the range of [0.5, 1], three curve graphs are drawn, taking parameter p as abscissa and leave-one-out accuracy as ordinate, as Fig. 4, Fig. 5 and Fig. 6. By analyzing Fig. 1 to Fig. 6, it can be seen that optimal values of parameter b are dispersed between the range of [0, 1], while optimal values of parameter p are much more centralized and its empirical value range should be concluded as [0.7, 1].

3.4 Experiment Results and Analysis

Experiment results of financial distress prediction model based on similarity weighted voting CBR are listed in Table 2. For comparative study, leave-one-out accuracy of MDA, Logit, NNs, SVM, which are the research results of the authors' former study based on the same sample data (see reference [7] for detailed information of model parameters and experiment tools used in former research), are also listed in Table 2.

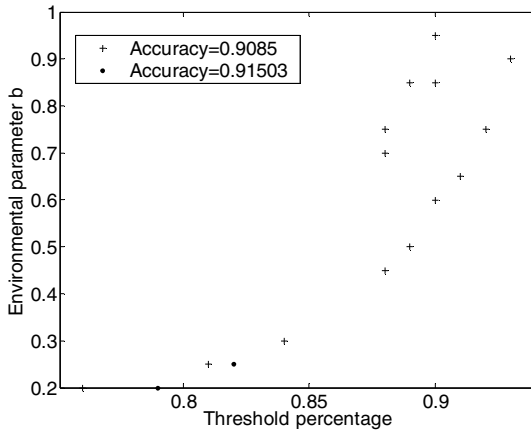


Fig. 1. Good pairs of parameter values obtained by grid-search at year ($t-1$)

As Table 2 shows, the predictive ability of similarity weighted voting CBR model declines from year ($t-1$) to year ($t-3$), just the same as the other four models, which indicates that the nearer to the time when financial distress breaks out, the more information content the financial ratios contain, so that the more strong predictive ability each model has. According to leave-one-out accuracy at year ($t-1$) and ($t-2$), similarity weighted voting CBR model prominently outperforms all the other four models in their application to financial distress prediction. However, at year ($t-3$), CBR model's leave-one-out accuracy is not as high as MDA, Logit and SVM. This may indicates that the financial distress prediction model based on similarity weighted

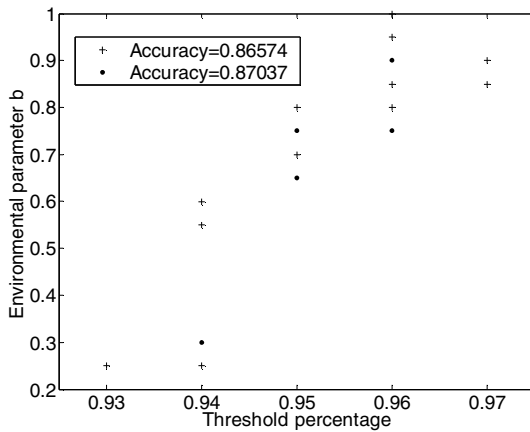


Fig. 2. Good pairs of parameter values obtained by grid-search at year ($t-2$)

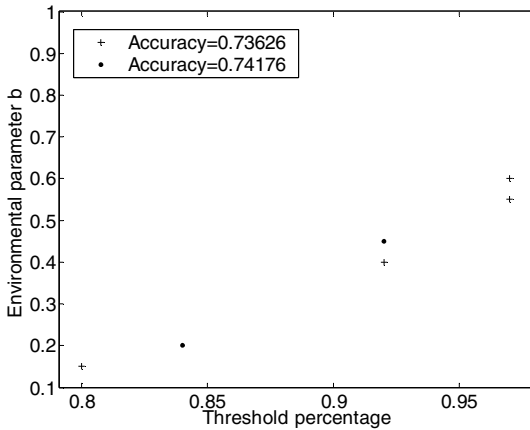


Fig. 3. Good pairs of parameter values obtained by grid-search at year ($t-3$)

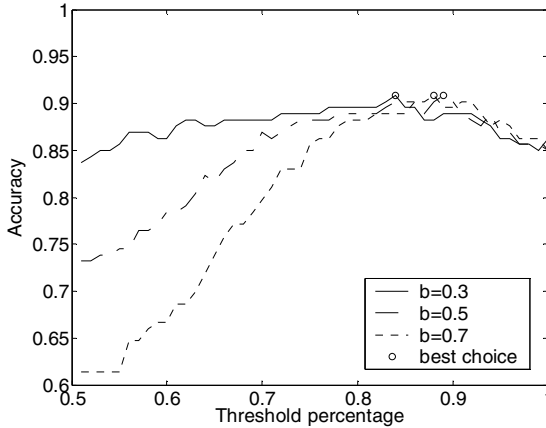


Fig. 4. Curve relationship between p and accuracy with b fixed at certain values at year ($t-1$)

Table 2. The leave-one-out accuracy of different models at different years

Year	Models	MDA (%)	Logit (%)	NNs (%)	SVM (%)	CBR (%)
($t-1$)		88.2	86.9	88.2	88.9	91.5
($t-2$)		85.2	84.7	84.3	85.6	87.0
($t-3$)		75.3	77.5	74.2	78.6	74.2

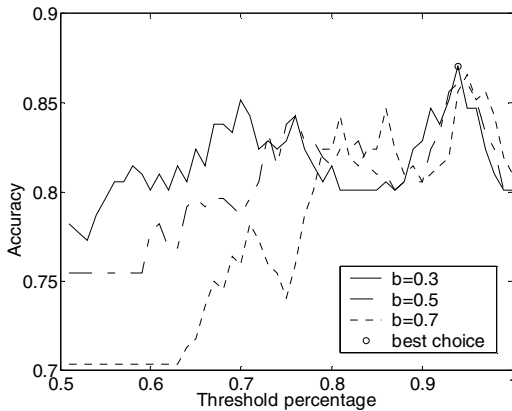


Fig. 5. Curve relationship between p and accuracy with b fixed at certain values at year ($t-2$)

voting CBR proposed in this paper has very good predictive ability for enterprises which might probably run into financial distress in less than two years, and it is more suitable to predict enterprises' financial distress in short term. For enterprises which

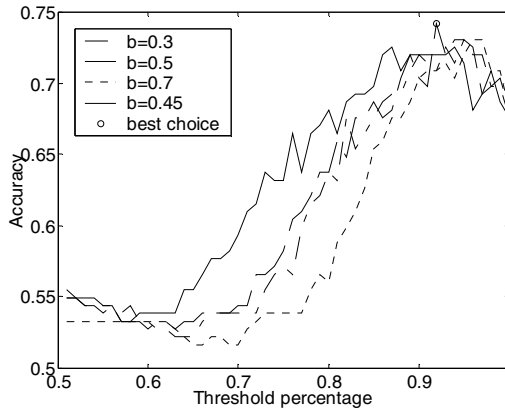


Fig. 6. Curve relationship between p and accuracy with b fixed at certain values at year ($t-3$)

will probably run into financial distress in more than two years, because CBR model has worse predictive ability than MDA, Logit and SVM, when determining the final predicted financial state class of an enterprise, its prediction result can only act as a reference instead of a decisive factor.

4 Conclusion

By integrating the thought of CBR and weighted voting combination, this paper proposed a financial distress prediction model based on similarity weighted voting CBR, which mainly include three parts, i.e. representation of enterprise case in financial distress prediction, retrieval of similar cases of target enterprise case, and combination of target class based on similarity weighted voting. In the empirical experiment, three years' data of 135 pairs of Chinese listed companies were selected as initial sample data; stepwise discriminant analysis method was used to select quantitative factor features; and grid-search technique was utilized to find good model parameters and their empirical value ranges. By comparing the experiment results of similarity weighted voting CBR model with those of MDA, Logit, NNs and SVM, it is concluded that this model has very good predictive ability for enterprises which might probably run into financial distress in less than two years, and it is suitable for enterprises' short-term financial distress prediction. So the proposal of financial distress prediction model based on similarity weighted voting CBR and its experiment conclusion can further widen the methodology system and enrich the theory system for financial distress prediction.

References

1. Beaver W.: Financial Ratios as Predictors of Failure. *Journal of Accounting Research*, Vol. 4. (1966) 71-111
2. Altman E. I.: Financial Ratios Discriminant Analysis and the Prediction of Corporate Bankruptcy. *Journal of Finance*, Vol. 23. (1968) 589-609

3. Ohlson J. A.: Financial Ratios and Probabilistic Prediction of Bankruptcy. *Journal of Accounting Research*, Vol. 18. (1980) 109-131
4. Odom M., Sharda R.: A Neural Networks Model for Bankruptcy Prediction. *Proceedings of the IEEE International Conference on Neural Network* (1990) 163-168
5. Shin K.-S., Lee T. S., Kim H.-J.: An Application of Support Vector Machines in Bankruptcy Prediction Model. *Expert Systems with Applications*, Vol. 28. (2005) 127-135
6. Min J. H., Lee Y.-C.: Bankruptcy Prediction Using Support Vector Machine with Optimal Choice of Kernel Function Parameters. *Expert Systems with Applications*, Vol. 28. (2005) 128-134
7. Hui X.-F., Sun J.: An Application of Support Vector Machine to Companies' Financial Distress Prediction. *Lecture Notes in Computer Science*, Vol. 3885. Springer-Verlag, Berlin (2006) 274-282
8. Schank, R.: *Dynamic Memory: A Theory of Learning in Computers and People*. Cambridge University Press, New York (1982)
9. Sankar K. P., Simon C. K. S.: *Foundations of Soft Cased-Based Reasoning*. Wiley, New Jersey (2004)
10. Hansen J., McDonald J., Stice, J.: Artificial Intelligence and Generalized Qualitative Response Models: An Empirical Test on Two Audit Decision Making Domains. *Decision Science*, Vol. 23. (1992) 708-723
11. Kim S. H., Noh H. J.: Predictability of Interest Rates Using Data Mining Tools: A Comparative Analysis of Korea and the US. *Expert Systems with Applications*, Vol. 13. (1997) 85-95
12. Jo H., Han I.: Bankruptcy Prediction Using Case-Based Reasoning, Neural Networks, and Discriminant Analysis. *Expert Systems with Applications*, Vol. 13. (1997) 97-108
13. Xu X.- Z., Gao G.-A.: Research and Implement of Case - based Reasoning in an Multi - criteria Evaluation IDSS. *Computer Integrated Manufacture System (in Chinese)*, Vol. 7. (2001) 16-18
14. Ai W.-G., Li H., Sun J.: Case Based Reasoning Model in a Multi-Criteria Group Intelligent Decision Support System. *Journal of Harbin Institute of Technology (in Chinese)*, Vol. 36. (2004) 805-807
15. Hsu C.-W., Chang C.-C., Lin C.-J.: A Practical Guide to Support Vector Classification. Technical Report. <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

Customer Churn Prediction by Hybrid Model

Jae Sik Lee and Jin Chun Lee

Dept. of Business Administration, Graduate School, Ajou University
San 5, Wonchun-Dong, Youngtong-Gu, Suwon 443-749, Korea
leejsk@ajou.ac.kr, giny777@empal.com

Abstract. In order to improve the performance of a data mining model, many researchers have employed a hybrid model approach in solving a problem. There are two types of approach to build a hybrid model, i.e., the whole data approach and the segmented data approach. In this research, we present a new structure of the latter type of hybrid model, which we shall call SePI. In the SePI, input data is segmented using the performance information of the models tried in the training phase. We applied the SePI to a real customer churn problem of a Korean company that provides streaming digital music services through Internet. The result shows that the SePI outperformed any model that employed only one data mining technique such as artificial neural network, decision tree and logistic regression.

1 Introduction

When the target domain is quite complex, we often face with the difficulty of building a good single model that shows high hit ratio while overcoming overfitting problem. Therefore, many researchers have paid attention to hybrid models.

There are two types of approaches to build a hybrid model, i.e., the whole data approach and the segmented data approach. In the first approach, several single models are built using the whole data, and then the results obtained from single models are used by each other or combined to produce a solution. In the second approach, the whole data is divided into several segments, and then a single model using each segmented data is built. In the first approach, all built models might have learned similar patterns. In other words, some specific data patterns might not be captured by any built models. In the second approach, it can be difficult to devise a criterion for data segmentation. In this research, we suggest new structure of hybrid model that alleviates the above-mentioned problems and deficiencies. Using a real data acquired from a Korean streaming digital music service company, we build a customer churn prediction model.

This paper consists of six sections. In section 2, we describe the types of hybrid models we arranged by reviewing the previous researches and introduce our new structure of hybrid model. In section 3, the problem of customer churn prediction for digital music service is described. The single models for our problem are constructed and their performances are compared. In section 4, our hybrid model is completed and its performance is compared with the single models' performances and evaluated. The final section provides concluding remarks and directions for further research.

2 Types of Hybrid Model

2.1 Whole Data Approach

There are four types of hybrid model that employs the whole data approach as shown in Figure 1. In W1 type, model A functions as a preprocessor for model B [6], [9], [14], [16]. For example, feature selection is performed by a Decision Tree (DT) model, and then an Artificial Neural Network (ANN) model is built using those selected features. While two models are performed sequentially in W1 type, model A is included in model B in W2 type [11]. For example, genetic algorithm is used for building an ANN model. The chromosome in genetic algorithm contains such information as whether certain features are to be included in the input layer, the number of nodes in hidden layer and so forth. In W3 type, the result obtained from model A is used as input for model B. For example, the result of DT model can be used as a part of input data for ANN model. Finally, in W4 type, model A, model B and model C produce their own results using the given data as a whole. Then the final solution is obtained by combining these results in some appropriate way [2], [3], [4], [5], [7], [8], [10], [12], [13], [17].

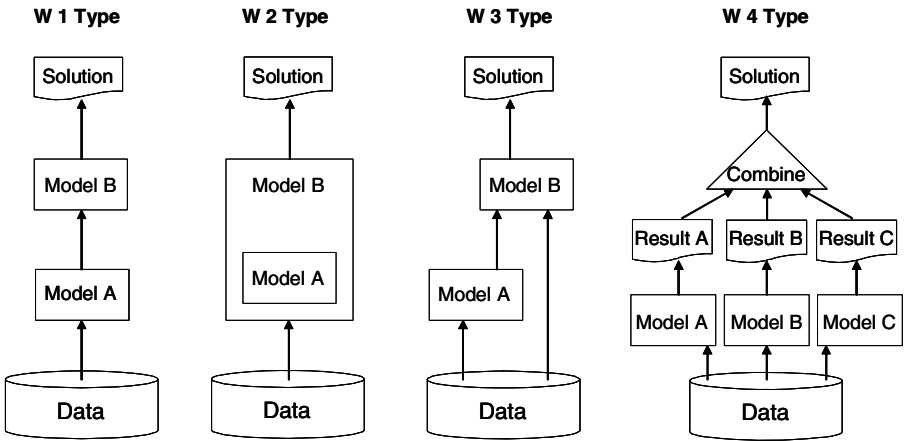


Fig. 1. W Types of Hybrid Model: Whole Data Approach

2.2 Segmented Data Approach

In segmented data approach, the given data is divided into several segments, and a different model is built for each of these segments. This kind of hybrid models are categorized into two types as shown in Figure 2 according to whether the results of models are combined or not.

Data can be segmented either according to the characteristics of certain features or by some modeling techniques such as clustering [15]. When the data is segmented according to certain features, we shall call this S type of hybrid model 'S1_F' or 'S2_F' type. When the data is segmented by some modeling techniques, we shall call this S

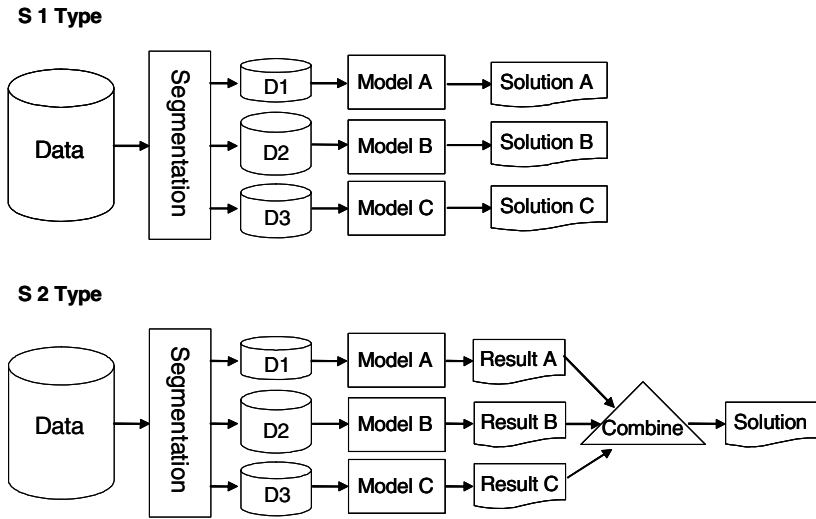


Fig. 2. S Types of Hybrid Model: Segmented Data Approach

type of hybrid model 'S1_M' or 'S2_M' type. An example of S1_F type is as follows: Suppose the data we will analyze contains existing customers and new customers. The records of new customers may have more missing values than those of existing customers. If we are to build a single model using this data, we have to put much effort to handle missing values. Therefore, for this data, it is more effective and efficient to build different models for the existing customers and the new customers. When using the built model, depending on the status of a customer, only the corresponding model will be used. In other words, we do not need to worry about combining the results obtained from those different models. An example of S2_F type can be found in Bay [1]. He presented an algorithm called Multiple Feature Subsets (MFS), a combining algorithm designed to improve the accuracy of the nearest neighbor (NN) classifier. MFS combines multiple NN classifiers each using only one random subset of features.

2.3 Hybrid Model of this Research: SePI

The hybrid model we propose in this research, shown in Figure 3, belongs to the S1_M type. Discrimination Model is developed using the performance information of

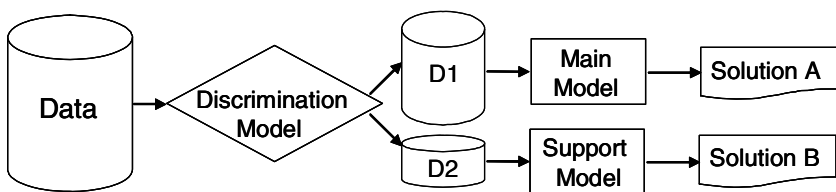


Fig. 3. SePI Hybrid Model

Main Model on the training data set. We shall call this hybrid model 'SePI' (Segmentation by Performance Information) hybrid model.

The SePI hybrid model is constructed as follows:

- Step 1: Build Main Model: Using the training data set, build various models using various techniques. After comparing the performances of these models, select the best model and set it 'Main Model'.
- Step 2: Build Discrimination Model: Append a feature, which contains the information whether Main Model predicts correctly or not on each of the data, to the data set. By setting this feature as the target feature, build a model that discriminates the data on which Main Model will show good performance.
- Step 3: Build Support Model: Using the data on which Main Model does not predict correctly, build various models in the same way as in Step 1. Select the best model and set it 'Support Model'.
- Step 4: Complete the SePI Hybrid Model: By connecting Main Model, Discrimination Model and Support Model as shown in Figure 3, complete constructing the SePI hybrid model.

3 Customer Churn Prediction by Single Models

The data used in this research is a real data obtained from a Korean company that provides digital music service through Internet. In Korea, the size of digital music service market exceeded the size of traditional music market in 2003, and has increased by 30% every year. From January 2005, transaction of music on Internet is charged by legislation in Korea. Since then, competition for occupying the market has been intensified severely. Under this competition, a company's primary concern is to increase the ratio of its regular subscribers, and it naturally follows that the company should prevent the existing subscribers from churning.

In our research, we define a churn-customer as "an existing customer who does not subscribe to any service in the next month." According to our definition, about 31% of 11,587 subscribers are classified as churn-customers.

The raw data consists of customer registration data, customer transaction data, customer interest data, service history log data and music sources. The data set for our research consists of 11,587 subscribers' demographic and transaction data. After feature selection, a total of 24 features consisting of 7 categorical features and 17 numerical features were selected. We shall call this data set 'original data set'.

The original data set is divided into three data sets, i.e., training data set, validation data set and test data set. For ANN model, we divided the data set into three at the ratio of 5, 3 and 2 for training, validation and test, respectively. For C5.0 DT model and Logistic Regression (LR) model, we set the parameter values by prior data analysis. Therefore validation data set is not required. For these models, we divided the data set into two at the ratio of 8 and 2 for training and test, respectively. In order to prove the stability of our model, we performed 10-fold cross validation.

The churn prediction hit ratio of the three component models, i.e., ANN, C5.0 and LR models on the test data set are presented in Table 1.

Table 1. Hit Ratio of Single Models on Test Data Sets

	Fold										Unit: %	
	01	02	03	04	05	06	07	08	09	10	Mean	Var.
ANN	82.5	82.4	82.7	82.3	83.4	82.8	80.3	81.8	82.3	82.3	83.4	0.539
C5.0	83.5	82.9	83.4	83.2	83.4	83.5	81.6	82.7	83.8	83.7	83.2	0.394
LR	82.9	83.2	82.9	82.4	82.4	82.8	80.9	81.5	82.4	82.9	82.3	0.641

As shown in Table 1, C5.0 model with hit ratio 83.2% performs better than ANN model with 82.4% and LR model with 82.3%. Moreover, variance of C5.0 model is smaller than those of other models, which stands for C5.0 model is more stable.

We performed the paired t-test for statistically verifying the difference of hit ratios of two models. For significance level of 1%, it is proved that C5.0 model outperforms other two models, and that there is no difference between the hit ratios of ANN model and LR model.

4 SePI Hybrid Model

4.1 Main Model

As described in section 2.3, the best model among single models is selected as Main Model. By the performance comparison described in section 3, we select C5.0 model as Main Model.

4.2 Discrimination Model and Support Model

The role of Discrimination Model is, given a new data, to determine whether Main Model is appropriate for the new data. Discrimination Model is built in the following four steps:

- Step 1 : Add a feature called 'S/F', which represents whether Main Model predicts correctly or not on each of the data, to the original data set.
- Step 2 : Perform feature selection and construct model data sets.
- Step 3 : Set the feature 'S/F' as target feature, build various models using various modeling techniques.
- Step 4 : After comparing the performances of these models, select the best model and set it 'Discrimination Model'.

Table 2 shows the hit ratios of each candidate for Discrimination Model.

On the validation data set, the ANN candidate model, D_ANN with hit ratio 86.05% performs better than the C5.0 candidate model, D_C5.0 with hit ratio 83.16%,

Table 2. Average Hit Ratios of the Candidates for Discrimination Model

	Training Data Set (%)			Validation Data Set (%)		
	D_ANN	D_C5.0	D_LR	D_ANN	D_C5.0	D_LR
Mean	86.77	83.51	83.43	86.05	83.16	83.01
Variance	2.53	0.03	0.03	2.62	0.39	0.39

and the LR candidate model, D_LR with hit ratio 83.01%. The results of paired t-test prove that, for significance level of 1%, D_ANN outperforms D_C5.0 and D_LR.

Support Model is the model that performs prediction on the data for which Main Model is judged inappropriate by Discrimination Model. Therefore, Support Model is built using the data on which Main Model predicts incorrectly. Support Model is built in the following four steps:

- Step 1 : Extract the data on which Main Model predicts incorrectly from the original data set.
- Step 2 : Perform feature selection and construct model data sets.
- Step 3 : Build various models using various modeling techniques.
- Step 4 : After comparing the performances of these models, select the best model and set it 'Support Model'.

We employed ANN to build Support Model. The structure and the parameter settings of ANN are the same as those used in developing ANN single model in section 3. Needless to say, any modeling techniques can be employed in this step. As shown in Table 3, the hit ratio on the validation data set is 90.7% on the average.

Table 3. Average Hit Ratios of ANN for Support Model

	Training Data Set (%)	Validation Data Set (%)
Mean	95.1	90.7
Variance	4.91	4.87

4.3 Completion and Evaluation of SePI Hybrid Model

The structure of the completed SePI hybrid model for churn prediction of digital music services is depicted in Figure 4.

Table 4 shows the performance comparison between the SePI and the single models. As shown in Table 4, the hit ratio of the SePI hybrid model is 86.5%, and it is greater than those of single models.

The result of the paired t-test proved that the SePI hybrid model outperforms all single models at the 1% significance level.

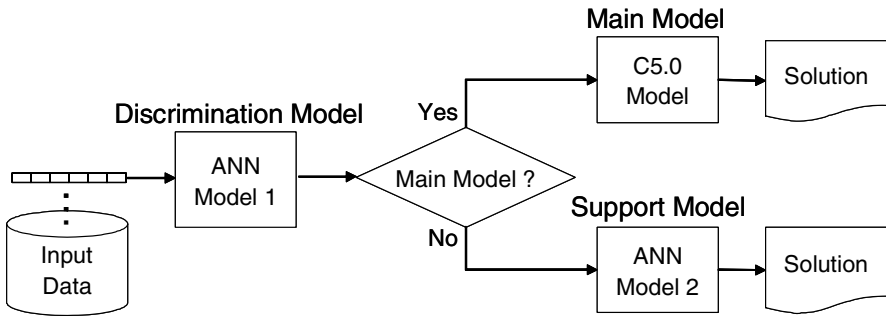


Fig. 4. Structure of the SePI Hybrid Model for Customer Churn Prediction

Table 4. Performance Comparison between the SePI and Single Models

	SePI	ANN	C5.0	LR
Average Hit Ratio	86.5 %	82.4 %	83.2 %	82.3 %

5 Conclusion

In this research, we applied our hybrid model to a real customer churn problem of a Korean company that provides streaming digital music service through Internet. The result shows that our hybrid model outperformed any other single models. The hit ratio of our hybrid model was 86.5% and it was 3.3% point higher than C5.0 decision tree model that showed best performance among single models.

We do not assert that our hybrid model is appropriate for all general data mining problems. If Main Model can handle most of the data set, we may not have a good supply of data for constructing Support Model and Discrimination Model. However, for problems with complex data patterns, our hybrid model can provide the data miners with possibility of improving the performance.

Acknowledgment

This research is supported by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Information and Communication (MIC), 21st Century Frontier R&D Program in Korea.

References

1. Bay, S.: Nearest Neighbor Classification from Multiple Feature Subsets. *Intelligent Data Analysis*, Vol.3. (1999) 191-209
2. Breiman, L.: Bagging Predictors. *Machine Learning*, Vol.24. (1996) 123-140

3. Cho, S.B.: Pattern Recognition with Neural Networks Combined by Genetic Algorithm. *Fuzzy Sets and Systems*, Vol.103. (1999) 339-347
4. Daskalaki, S., Kopanas, I., Goudara, M., Avouris, N.: Data Mining for Decision Support on Customer Insolvency in Telecommunications Business. *European Journal of Operational Research*, Vol.145. (2003) 239-255
5. Hothorn, T., Lausen, B.: Bagging Tree Classifiers for Laser Scanning Images: A Data- and Simulation-Based Strategy. *Artificial Intelligence in Medicine*, Vol.27. (2003) 65-79
6. Hsieh, N.C.: Hybrid Mining Approach in the Design of Credit Scoring Models. *Expert Systems with Applications*, Vol.28. (2005) 655-665
7. Jiang, Y., Zhou, Z.H., Chen, Z.Q.: Rule Learning based on Neural Network Ensemble. In: *Proceedings of the International Joint Conference on Neural Networks*, Honolulu HI (2002) 1416-1420
8. Kim, Y.S., Street, W.N., Menczer, F.: Optimal Ensemble Construction via Meta-evolutionary Ensembles. *Expert Systems with Applications*, Vol.30. (2006) 705-714
9. Last, L., Kandel, A., Maimon, O.: Information-Theoretic Algorithm for Feature Selection. *Pattern Recognition Letters*, Vol.22. (2001) 799-811
10. Opitz, D., Maclin, R.: Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, Vol.11. (1999) 169-198
11. Sexton, R.S., Sikander, N.A.: Data Mining Using a Genetic Algorithm Trained Neural Network. *International Journal of Intelligent Systems in Accounting, Finance & Management*, Vol.10. (2001) 201-210
12. Tax, D. M.J., Breukelen, M.V., Duin, R. P.W., Kittler, J.: Combining Multiple Classifiers by Averaging or by Multiplying. *Pattern Recognition*, Vol.33. (2000) 1475-1485
13. Wang, X., Wang, H.: Classification by Evolutionary Ensembles. *Pattern Recognition*, Vol. 39. (2006) 595-607
14. Yang J., Honavar, V.: Feature Subset Selection Using a Genetic Algorithm. *IEEE Intelligent Systems and their Applications*, Vol.13. (1998) 44-49
15. Yeo, A.C., Smith, K.A., Willis, R.J., Brooks, M.: Clustering Technique for Risk Classification and Prediction of Claim Costs in the Automobile Insurance Industry. *International Journal of Intelligent Systems in Accounting, Finance & Management*, Vol.10. (2001) 39-50
16. Zhang, P., Verma, B., Kumar, K.: Neural vs. Statistical Classifier in Conjunction with Genetic Algorithm based Feature Selection. *Pattern Recognition Letters*, Vol.26. (2005) 909-919
17. Zhou, Z.H., Wu, J.X., Jiang, Y., Chen, S.F.: Genetic Algorithm based Selective Neural Network Ensemble. In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, Seattle WA Vol.2. (2001) 797-802

Base Vector Selection for Kernel Matching Pursuit

Qing Li and Licheng Jiao

Institute of Intelligent Information Processing, Xidian University, P.O. 224, Xi'an, 710071,
P.R. China
kingdomyangfan@hotmail.com, lchjiao@xidian.edu.cn

Abstract. Kernel matching pursuit (KMP) is a relatively new learning algorithm to produce non-linear version of conventional supervised and unsupervised learning algorithm. However, it also contains some defects such as storage problem (in training process) and sparsity problem. In this paper, a new method is proposed to pre-select the base vectors from the original data according to vector correlation principle, which could greatly reduce the scale of the optimization problem and improve the sparsity of the solution. The method could capture the structure of the data space by approximating a basis of the subspace of the data; therefore, the statistical information of the training samples is preserved. In the paper, the deduction of mathematical process is given in details and the number of simulation results on artificial data and practical data has been done to validate the performance of base vector selection (BVS) algorithm. The experimental results show the combination of such algorithm with KMP can make great progress while leave the performance almost unchanged.

1 Introduction

Kernel matching pursuit (KMP) is a new and promising classification technique proposed by Pascal Vincent and Yoshua Bengio. [1]. The basic idea of KMP originates from Matching Pursuit, a greedy constructive algorithm that approximates a given function by a linear combination of basis functions choosing from a redundant basis function dictionary, and can be seen as a form of boosting [2,3]. The performance of the KMP shows comparable to that of Support Vector Machine (SVM), while typically requiring far fewer support points [4,5,6]. In the following, the development of the KMP will be overviewed.

1.1 The Development of KMP

In the development of the kernel matching pursuit, two milestones have to be mentioned. The first milestone is the construction of matching pursuit algorithm, proposed by Mallat and Zhang, which can be seen as basic building blocks of the KMP's production [7]. Matching pursuit was introduced in the signal-processing community as an algorithm "that decomposes any signal into a linear expansion of waveforms that are selected from a redundant dictionary of function". It is a general, greedy sparse-approximation scheme. Given l noisy observations $\{y_1, \dots, y_l\}$ and a finite dictionary D of functions in Hilbert space H , the aim of the algorithm to find sparse

approximations of $\{y_1, \dots, y_l\}$ that is the expansion of the form $\bar{f}_N = \sum_{n=1}^N \beta_n \bar{g}_n$, where N means the maximum of the basis functions. A preferable alternation of the matching pursuit is back-fitting matching pursuit. In basic algorithm, when appending $\beta_{i+1} \bar{g}_{i+1}$ in i th iterative, the expansion may not be optimal, so doing back-fitting is to recompute the optimal set of coefficients $\beta_{1, \dots, i+1}$ at each step instead of only the last β_{i+1} to approximate the objective function more accurately. While this can be quite time-consuming, one trade-off method is to do back-fitting algorithm in every few steps instead of every step.

The second milestone is the successful use of kernel method in the problem of machine learning. Kernel method were first used in Support Vector Machines (SVM) [4,5], and applied to a wide class of problems such as classification and regression. The kernel methods map the input space of the data into a high dimension space called feature space, in which the problem becomes linearly separable.

Kernel matching pursuit (KMP) is simply the idea of applying the Matching Pursuit (MP) family of algorithms to problem in machine learning, using a kernel-based dictionary [1]. KMP uses kernels to map the data in input space to a high dimensional feature space in which we can process a problem in linear form. Given a kernel function $K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, we construct the basis dictionary of MP by the kernel centered on the training data: $D = \{d_i = K(*, x_i) | i = 1 \sim l\}$. Method of kernels is enlightened in great part to the success of the Support Vector Machine (SVM), and there exist a lot of commonly used kernels, such as polynomial kernel with the form of $K(x, x_i) = [(x, x_i) + 1]^d$ and RBF kernel with the form of $K(x, x_i) = \exp(-\|x - x_i\|/2p)$. Executing kernel matching pursuit algorithm to get the approximation function in regression [10]

$$f_N(\mathbf{x}) = \sum_{k=1}^N \beta_k g_k(\mathbf{x}) = \sum_{k=1}^N \beta_k K(\mathbf{x}, \mathbf{x}_k) \tag{1}$$

or the decision function in classification

$$f_N(\mathbf{x}) = \text{sgn} \left(\sum_{k=1}^N \beta_k g_k(\mathbf{x}) \right) = \text{sgn} \left(\sum_{k=1}^N \beta_k K(\mathbf{x}, \mathbf{x}_k) \right) \tag{2}$$

where $\{\mathbf{x}_k | k = 1 \sim N\} \in \{\mathbf{x}_1, \dots, \mathbf{x}_l\}$ are support points. There is two ways to stop the algorithm. One is that the basis functions reach the maximum N , the other is that the error goes below a predefined given threshold. More details about KMP can be seen in [1].

1.2 Proposed Approach

The KMP uses kernel methods to map the data in input space to a high-dimensional feature space in which the problem becomes linearly separable. However, there still exist some drawbacks in KMP. First, training a KMP needs to handle with a very large and fully dense kernel matrix. For large-scale problems, perhaps it cannot be saved in the main memory at all. Thus, traditional optimization algorithms like Newton or Quasi-Newton cannot be directly used. Second, KMP is a sparse algorithm in theory,

but the sparsity of the solution is not as good as what we expect, so many problems are presented such as the reduction of the time when test a new samples.

It has been shown that the capacity of generalization depends on the geometrical characteristics of the training data and not on their dimensionality [11]. Therefore, if these characteristics are well chosen, the expected generalization error could be small even if the feature space has a huge dimensionality[12,13]. In this paper, we proposed a new insight to pre-select a subset of the training samples according to vector correlation principle. Base vectors are far fewer than the original data in many circumstances, so training SVM by these samples could greatly reduce the scale of the optimization problem and improve the sparsity of the support vector solution.

Given a nonlinear mapping function ϕ , it maps the training data $(\mathbf{x}_i, y_i)_{1 \leq i \leq l}$ in input space into a high-dimensional feature space $(\phi(\mathbf{x}_i), y_i)_{1 \leq i \leq l}$. According to the linear theory, these feature vector $\{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_l)\}$ maybe not linear independent. Suppose $\{\phi(\tilde{\mathbf{x}}_1), \phi(\tilde{\mathbf{x}}_2), \dots, \phi(\tilde{\mathbf{x}}_M)\}$ (often M is far less than l) are the base vectors of the $\{\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_l)\}$, then any vector $\phi(\mathbf{x}_i)$ can be expressed in the linear form of the base vector $\sum \beta_j \phi(\tilde{\mathbf{x}}_j)$. Therefore, training on the original data will be equal to training on the base vector samples as well as we exactly know the corresponding coefficients matrix $\beta = [\beta_1, \beta_2, \dots, \beta_M]^T$, because

$$\begin{bmatrix} \phi(\mathbf{x}_1) \\ \phi(\mathbf{x}_2) \\ \vdots \\ \phi(\mathbf{x}_l) \end{bmatrix} = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1M} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{l1} & \beta_{l2} & \cdots & \beta_{lM} \end{bmatrix} \begin{bmatrix} \phi(\tilde{\mathbf{x}}_1) \\ \phi(\tilde{\mathbf{x}}_2) \\ \dots \\ \phi(\tilde{\mathbf{x}}_M) \end{bmatrix} \tag{3}$$

where $[\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_l)]^T$ is the actual training samples in feature space. This paper aims to pre-select the base vectors $X_B = \{x_{B_1}, x_{B_2}, \dots, x_{B_M}\}$ ($1 \leq B_i \leq M$) according to the vector correlation principle and the kernel function, and then training SVM with the base vectors to reduce the scale of the optimization problem and improve the sparsity of the solution.

1.3 Organization of the Paper

This paper is organized as follows. We first deduce the mathematical process of the base vector selection (BVS) in section 2, and then (in section 3) provide the simulation results when applying BVS to KMP in the regressive and classification problem. Finally, we give our concluding remarks.

2 Base Vector Selection

Let $(\mathbf{x}_i, y_i)_{1 \leq i \leq l}$ ($\mathbf{x}_i \in R^N$, $y_i \in R$) be the training samples. A nonlinear mapping function ϕ maps the input space of the training data into a feature Hilbert space H :

$$\begin{aligned} \Phi: R^N &\rightarrow H \\ \mathbf{x} &\rightarrow \phi(\mathbf{x}) \end{aligned} \tag{4}$$

Therefore, the mapping training set in feature space are $(\phi(\mathbf{x}_i), y_i)_{1 \leq i \leq l}$ ($\phi(\mathbf{x}_i) \in H$, $y_i \in R$), which lies in a subspace H_s of the H with the dimension up to l . In practice, the dimension of this subspace is far lower than l and equal to the number of its base vector. As we have shown in the introduction, training on the original samples will be equal to training on the base vector samples as well as we exactly know the corresponding coefficients matrix. So we propose a preprocessing method to select base vectors of the original data and use these base vectors as the training samples to train SVM so that to reduce the scale of the optimization problem.

In the following, the method of select base vector will be introduced. For notation simplification: for each \mathbf{x}_i the mapping is noted $\phi(\mathbf{x}_i) = \phi_i$ for $1 \leq i \leq l$ and the selected base vectors are noted by \mathbf{x}_{B_j} and $\phi(\mathbf{x}_{B_j}) = \phi_{B_j}$ for $1 \leq j \leq M$ (M is the number of the base vectors). For a given base vectors $X_B = \{x_{B_1}, x_{B_2}, \dots, x_{B_M}\}$, the mapping of any vector x_i can be expressed as a linear combination of X_s with the form

$$\hat{\phi}_i = \beta_i^T \Phi_B \tag{5}$$

where $\Phi_B = (\phi_{B_1}, \dots, \phi_{B_M})^T$ is the matrix of the mapping base vectors and $\beta_i = (\beta_{i1}, \dots, \beta_{iM})^T$ is the corresponding coefficient vector.

Given $(\mathbf{x}_i, y_i)_{1 \leq i \leq l}$, the goal is to find the base vectors $X_B = \{x_{B_1}, x_{B_2}, \dots, x_{B_M}\}$ such that for arbitrary mapping ϕ_i , the estimated mapping $\hat{\phi}_i$ is as close as possible to ϕ_i . For this purpose, we minimize the following form to select the base vectors:

$$\min_{X_B} \left(\sum_{\mathbf{x}_i \in X} (\|\phi_i - \hat{\phi}_i\|^2) \right) \tag{6}$$

Let $\rho_i = \|\phi_i - \hat{\phi}_i\|^2$, we can get:

$$\rho_i = \|\phi_i - \beta_i^T \Phi_B\|^2 = (\phi_i - \beta_i^T \Phi_B)^T (\phi_i - \beta_i^T \Phi_B) = \phi_i^T \phi_i - 2\Phi_B^T \phi_i \beta_i + \beta_i^T \Phi_B^T \Phi_B \beta_i \tag{7}$$

Putting the derivative of ρ_i to zero gives the coefficient vector β_i :

$$\begin{aligned} \frac{\partial \rho_i}{\partial \beta_i} &= 2(\Phi_B^T \Phi_B) \beta_i - 2\Phi_B^T \phi_i, \\ \frac{\partial \rho_i}{\partial \beta_i} = 0 &\Rightarrow \beta_i = (\Phi_B^T \Phi_B)^{-1} \Phi_B^T \phi_i \end{aligned} \tag{8}$$

$(\Phi_B^T \Phi_B)^{-1}$ exists if the mapping base vectors are linear independent. Substituting (7) and (8) to (6), we get

$$\min_{X_B} \left(\sum_{\mathbf{x}_i \in X} (\phi_i^T \phi_i - \phi_i^T \Phi_B (\Phi_B^T \Phi_B)^{-1} \Phi_B^T \phi_i) \right) \tag{9}$$

By Mercer’s theorem, we can replace the inner product between feature space vectors by a positive defined kernel function over pairs of vectors in input space. In other word, we use the substitution $\phi(\mathbf{x})^T \phi(\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = k(\mathbf{x}, \mathbf{y})$ to (9) and get

$$\min_{X_B} \left(\sum_{x_i \in X} (k_{ii} - \bar{K}_{Bi}^T K_{BB}^{-1} \bar{K}_{Bi}) \right) \tag{10}$$

We defined the base set fitness F_B and the arbitrary vector fitness F_{B_i} corresponding to a given base vector set X_B by

$$F_B = \frac{1}{l} \sum_{x_i \in X} F_{B_i} \tag{11}$$

where

$$F_{B_i} = k_{ii} - \bar{K}_{Bi}^T K_{BB}^{-1} \bar{K}_{Bi} \tag{12}$$

Now, (10) is equivalent to:

$$\min_{X_B} (F_B) \tag{13}$$

The process of base vector selection is a greedy iterative algorithm. When selecting the first base vector, we look for the samples that gives the minimum F_B . In each iteration, (11) is used to estimate the performance of the current base set and (12) is used to select the next best candidate base vector, as the one having the maximal fitness F_{B_i} for the current base set, which means the collinearity between such vector and the current base set is the worst (or in other word, such vector could hardly be expressed as the linear combination of the current base vectors). A pre-defined base set fitness and the expected maximal number of the base vector could be used to stop the iterative process. When the current base set fitness reaches the pre-defined fitness (which means that X_B is a good approximation of the basis for the original dataset in H) or the number of the base vectors reaches the expected value, the algorithm stops. Another important stop criterion must be noted that the algorithm should be stopped if the matrix of $K_{BB} = (\Phi_B^T \Phi_B)$ is not invertible anymore, which means the current base set X_B is the real basis in feature space H .

3 Validation of Effectiveness

In the paper, we compare the BVS KMP with the standard KMP and adopting RBF kernel both for KMP and base vector selection with the form $K(x, x_i) = \exp(-\|x - x_i\|^2 / 2p)$. We adopt the parameters' notation of BVS as follows: maxN—maximum of the base vectors; minFit—BVS stopping criterion (predefined accuracy). In the test of regression, we adopt the approximation error as $e_{ss} = \sqrt{(\sum_{i=1}^l (y_i - f_i)^2) / l}$. For avoiding the weak problem, each experiment has been performed 30 independent runs, and all experiments were carried on a Pentium IV 2.6Ghz with 512 MB RAM using Matlab 7.01 compiler.

3.1 Regression Experiments

3.1.1 Approximation of Single-Variable Function

In this experiment, we approximate the 1-dimension function, $y = \sin(x)/x$, which is also named Sinc function. We uniformly sample 1200 points over the domain $[-10,10]$, 400 of which are taken as the training examples and others as testing examples.

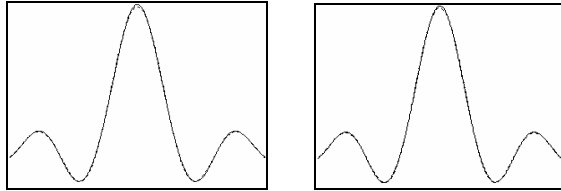


Fig. 1. Regression results by KMP(left with 107 SP/ 400 total¹) and BVS KMP (right with 48 SP/ 100 BSV/ 400 total). The solid line represents approximation curve and dotted line represents original function.

The kernel both for KMP and base vector selection adopt RBF kernel. For training KMP, RBF kernel parameter $\sigma=1.52$ and $N = 200$, $stops = 0.001$, $fitN = 5$; for BVS, $\max N = 100$, $\min Fit = 0.05$. Table 1 lists the approximation errors for BVS KMP and standard KMP algorithm. We also give the approximation drawings in Fig 1 with kernel parameter $\rho = 0.1$ for BVS.

Table 1. Approximation Results of 1-D Sinc Function

Algorithm	Para. ²	#.bv/ #.tr ³	#.sp	Test time (s)	Error
BVS KMP	$\rho = 0.1$	100/400	48	2.1094	0.0067
	$\rho = 0.2$	77/400	43	1.8906	0.0069
	$\rho = 0.3$	51/400	32	1.4063	0.0070
	$\rho = 0.4$	39/400	26	1.125	0.0071
Standard KMP	None	None/400	107	5.5	0.0065

3.1.2 Approximation of Two-Variable Function

This experiment is to approximate a two-dimension Sinc function with the form $y = \sin(\pi\sqrt{x_1^2 + x_2^2}) / (\pi\sqrt{x_1^2 + x_2^2})$ over the domain $[-10,10] \times [-10,10]$. We uniformly sample 10000 points, 400 of which are taken as the training examples and others as testing examples.

The kernel both for KMP and base vector selection adopt RBF kernel. For training KMP, RBF kernel parameter $\sigma=1.42$ and $N = 300$, $stops = 0.005$, $fitN = 5$; for BVS,

¹ 124 SV/ 400 total means 400 training samples in which contains 124 support vectors.

² In this paper, ‘Para.’ means the selected Kernel parameter for BVS.

³ We adopt note ‘#’ to present ‘the number of’ and ‘bv’, ‘tr’, ‘sp’ means the base vectors, training samples, support points, respectively.

maxN=200, minFit=0.05. Table 2 lists the approximation errors for BVS KMP and standard KMP algorithm.

Table 2. Approximation Results of 2-D Sinc Function

Algorithm	Para.	#.bv/#.tr	#.sp	Test time (s)	Error
BVS KMP	$p = 0.3$	200/400	63	25.906	0.070658
	$p = 0.416$	157/400	58	24.625	0.071138
	$p = 0.532$	122/400	44	18.625	0.071286
	$p = 0.821$	79/400	41	17.453	0.071568
Standard KMP	None	None/400	82	34.984	0.070374

3.1.3 Boston Housing Data Approximation

A well-known dataset, Boston housing dataset, choosing from the UCI machine learning database⁴, has been tested in this experiment. The input space of Boston housing dataset is 13 dimensions. It contains 506 samples, and 400 of them are taken as the training examples while others as testing examples. For training KMP, RBF kernel parameter $\sigma=23$ and $N = 300$, $stops = 0.005$, $fitN = 5$; for BVS, maxN=250, minFit=0.05. Table 3 lists the approximation errors for BVS KMP and standard KMP algorithm.

Table 3. Approximation Results of Boston Housing Data

Algorithm	Para.	#.bv / #.tr	#.sp	Test time (s)	Error
BVS KMP	$p = 15$	250/400	115	0.5365	7.2371
	$p = 18$	241/400	110	0.5136	7.2489
	$p = 22$	196/400	109	0.5001	7.2794
	$p = 25$	173/400	102	0.48438	7.2924
Standard KMP	None	None	125	0.5867	7.2357

3.2 Experiments of Pattern Recognition

3.2.1 Two Spirals' Problem

Learning to tell two spirals apart is important both for purely academic reasons and for industrial application [14]. In the research of pattern recognition, it is a well-known problem for its difficulty. The parametric equation of the two spirals can be presented as follows:

$$\begin{aligned}
 \text{spiral-1:} \quad & x_1 = (k_1\theta + e_1) \cos(\theta) \\
 & y_1 = (k_1\theta + e_1) \sin(\theta) \\
 \text{spiral-2:} \quad & x_2 = (k_2\theta + e_2) \cos(\theta) \\
 & y_2 = (k_2\theta + e_2) \sin(\theta)
 \end{aligned} \tag{14}$$

⁴ URL:<http://www.ics.uci.edu/mlearn>.

where k_1, k_2, e_1 and e_2 are parameters. In our experiment, we choose $k_1 = k_2 = 4, e_1 = 1, e_2 = 10$. We uniformly generate 2000 samples, and randomly choose 400 of them as training data, others as test data.

Before the training, we add noise to data—randomly choosing 80 training samples, changing its class attributes. The kernel both for KMP and base vector selection adopt RBF kernel. For training KMP, RBF kernel parameter $\sigma = 0.35$ and $N = 200, stops = 0.01, fitN = 5$; for BVS, $\max N = 200, \min Fit = 0.05$. Table 4 lists the classification result by BVS KMP and standard KMP algorithm, respectively.

Table 4. Recognition Results of Two Spirals

Algorithm	Para.	#.bv/ #.tr	#. sp	Test time (s)	Accuracy
BVS KMP	$p = 0.5$	200/400	102	7.0656	94.25%
	$p = 0.87$	176/400	93	6.8406	94.11%
	$p = 1.05$	153/400	90	6.7329	94.00%
	$p = 1.24$	133/400	86	6.5183	93.78%
Standard KMP	None	None	180	12.734	94.63%

3.2.2 Artificial Data Experiment

We generate artificial dataset with the parametric equation of the data as

$$\begin{cases} x = \rho \sin \varphi \cos \theta \\ y = \rho \sin \varphi \sin \theta \\ z = \rho \cos \varphi \end{cases} \quad \theta \in U[0, 2\pi] \quad \varphi \in U[0, \pi].$$

Parameter ρ of the first class is of continuous

uniform distribution $U[0, 50]$, and ρ in the second class is of $U[50, 100]$. We randomly generate 8000 samples, and randomly choose 600 samples as training data, others as test data. The kernel both for KMP and base vector selection adopt RBF kernel. For training KMP, RBF kernel parameter $\sigma = 4.75$ and $N = 400, stops = 0.01, fitN = 5$; for BVS, $\max N = 200, \min Fit = 0.05$. Table 5 lists the classification result by BVS KMP and standard KMP algorithm, respectively.

Table 5. Recognition Results of Artificial Dataset

Algorithm	Para.	#. bv	#.sp	Test time (s)	Accuracy
BVS KMP	$p = 1.0$	300	62	78.781	90.98%
	$p = 1.25$	204	49	53.75	90.92%
	$p = 1.5$	147	38	38.875	90.87%
	$p = 1.75$	109	30	28.961	90.01%
Standard KMP	None	None	465	522.19	91.08%

3.2.3 Pima Indian Diabetes Dataset

We did a further experiment on a well-known dataset, pima Indians diabetes datasets dataset, coming from UCI Benchmark Repository. Pima Indians diabetes dataset is a binary problem with 8 characteristic attributes and one class attribute. It has 768 samples and we randomly select 450 samples as training data and others as testing data.

The kernel both for KMP and base vector selection adopt RBF kernel. For training KMP, RBF kernel parameter $\sigma=15$ and $N=300$, $stops=0.01$, $fitN=5$; for BVS, $maxN=200$, $minFit=0.05$. Table 6 lists the classification result by BVS KMP and standard KMP algorithm, respectively.

Table 6. Recognition Results of Pima Indian diabetes

Algorithm	Para.	#.bv / #.tr	#.sp	Test time (s)	Accuracy
BVS KMP	$p=28$	200/450	149	2.0562	74.64%
	$p=30$	194/450	138	2.0483	74.41%
	$p=32$	177/450	134	1.9500	74.28%
	$p=34$	160/450	129	1.7281	73.98%
Standard KMP	None	None/450	203	2.8594	75.33%

4 Concluding Remarks

KMP uses a device called kernel mapping to map the data in input space to a high-dimensional feature space in which the problem becomes linear. It has made a great progress on the fields of machine learning. However, it also contains some defects such as storage problem (in training process) and sparsity problem, etc. In this paper, a method of base vectors selection is introduced to improve these disadvantages of KMP. The method could capture the structure of the data by approximating a basis of the subspace of the data; therefore, the statistical information of the training samples is preserved. From the method, three advantages will be obtained: 1. the storage problem of saving kernel matrix dictionary is greatly reduced; 2. the solution of the KMP becomes much sparser; 3. shorten the time to test a new sample (due to the less support points). We have tested many simulations on regression and classification problems. The data cited in the paper include not only artificial data but also the practical data. The experimental results show that combination of such algorithm with KMP can make great progress while can't sacrifice the performance of the KMP. Therefore, it is valid and feasible to implement such algorithm for embedded applications.

References

1. Pascal Vincent, Yoshua Bengio. Kernel matching pursuit. *Machine Learning*, 48:165--187, 2002.
2. Mallat S. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674-693, July 1989.
3. Davis G., Mallat S., Z. Zhang. Adaptive time-frequency decompositions. *Optical Engineering* 33(7), 2183-2191.
4. Vapnik, V.N. An overview of statistical learning theory. *IEEE Trans. Neural Network*, vol. 10, no.5, pp.988-999, 1999.
5. V. Vapnik. Three remarks on support vector machine. In: S. A. Solla, T. K. Leen, K. R. Müller (Eds.), *Advances in Neural Comput.* 10 (1998), pp.1299-1319.

6. Burges C. J. C. Geometry and invariance in kernel based method. In *Advance in Kernel Method-Support Vector Learning*. Cambridge, MA: MIT Press, 1999, pp.86-116.
7. Mallat S., Z. Zhang (1993, Dec.). Matching pursuit with time-frequency dictionaries. *IEEE Trans. Signal Proc.* 41 (12), 3397-3415.
8. C. J. C. Burges. Geometry and invariance in kernel based method. In *Advance in Kernel Method-Support Vector Learning*. Cambridge, MA: MIT Press, 1999, pp.86-116.
9. G. Baudat, F. Anouar. Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation* 12(10): 2385-2404 (2000).
10. Engel, Y., Mannor, S., Meir, R. The kernel recursive least-squares algorithm. *IEEE Trans. Signal Processing*, vol. 52, Issue: 8, pp. 2275-2285, August 2004.
11. Graepel, T., R. Herbrich, J. Shawe-Taylor (2000). Generalization Error Bounds for Sparse Linear Classifiers. In *Thirteenth Annual Conference on Computational Learning Theory*, 2000, pp. in press. Morgan Kaufmann.
12. G. Baudat, F. Anouar. Kernel based methods and function approximation. *International Joint Conference on Neural Network, IJCNN 2001*, Washington, DC, pp.1244-1249.
13. G. Baudat, F. Anouar. Feature vector selection and projection using kernels. *Neurocomputing*, 55(1-2): 21-28, 2003.
14. Lang K.J., Witbrock M.J. Learning to tell two spirals apart. In *Proc. 1989 Connectionist Models Summer School*, 1989, pp.52-61.

WaveSim Transform for Multi-channel Signal Data Mining Through Linear Regression PCA

R. Pradeep Kumar and P. Nagabhushan

Department of Studies in Computer Science,
University of Mysore, Karnataka, India – 570 006
rpradeep25@lycos.com, pnagabhushan@hotmail.com

Abstract. Temporal data mining is concerned with the analysis of temporal data and finding temporal patterns, regularities, trends, clusters in sets of temporal data. In this paper we extract regression features from the coefficients obtained by applying WaveSim Transform on Multi-Channel signals. WaveSim Transform is a reverse approach for generating Wavelet Transform like coefficients by using a conventional similarity measure between the function $f(t)$ and the wavelet. WaveSim transform provides a means to analyze a temporal data at multiple resolutions. We propose a method for computing principal components when the feature is of linear regression type i.e. a line. The resultant principal component features are also lines. So through PCA we achieve dimensionality reduction and thus we show that from the first few principal component regression lines we can achieve a good classification of the objects or samples. The techniques have been tested on an EEG dataset recorded through 64 channels and the results are very encouraging.

1 Introduction

Temporal Data Mining is a step in the process of Knowledge Discovery in Temporal Databases that enumerates structures (temporal patterns or models) over the temporal data. Any algorithm that enumerates temporal patterns from, or fits models to temporal data is a Temporal Data Mining Algorithm [1][2][3]. A multi-channel database is a repository where the information of an entity or object is recorded through multiple channels. The information may be recorded in the form of a numerical value or a signal or an image etc. If the information recorded is a signal then it is normally a temporal database. This paper deals with a dataset where the information of 'm' number of objects is recorded by N channels. So each object is associated with N signals. Such kind of datasets is very common in medical applications. In most cases N is a very large number i.e. the number of channels are more. This implies each sample is represented by N dimensional features where each feature is a signal. In this paper we propose a novel technique based on WaveSim transform and principal component regression lines for mining knowledge through a dimensionality reduction induced multi-resolution analysis in such multi-channel dataset. The method is a foundational idea which can be used in many applications dealing with huge number of channels and sensors. Section 2 will brief the concept of WaveSim transform and the procedure for extracting WaveSim regression features

from each signal. At the end of this process the dataset is transformed into a form where each sample/object is represented by ‘N’ number of regression line features corresponding to each wavelet scale. Section 3 would elaborate on computing the principal component regression lines and thus the dimensionality reduction that can be achieved by considering only the first few regression line features. Section 4 discusses the experimentation procedure and results.

2 WaveSim Transform

Wavelets or wavelet analysis or the wavelet transform refers to the representation of a signal in terms of a finite length or fast decaying oscillating waveform known as the mother wavelet. This waveform is scaled and translated to match the input signal. The wavelet transform coefficients has been used as an index of similarity between a function $f(t)$ and the corresponding wavelet, in the fields of pattern recognition and knowledge discovery. In these fields, the coefficients are generated to acquire a set of features. WaveSim transform [3] explores the possibility of a reverse approach for generating wavelet coefficients by using a conventional similarity measure between the function $f(t)$ and the wavelet. It is a reverse approach from the point that the wavelet coefficients are indices of similarity, and the proposed method is an alternate method to generate a normalized set of similarity indices, whose characteristics are similar to that of wavelet coefficients. More details of the novel interpretation of wavelet transform in the evolution of WaveSim transform is provided in [3].

2.1 Wavesim Transform

The WaveSim(WS) transform of $f(t)$ with respect to a wavelet $\psi_{a,b}(t)$ is defined as

$$WS(a,b) = Sim(f(t), \psi_{a,b}(t)) \tag{1}$$

Where $Sim(..)$ is the similarity measure computed as illustrated in figure 1 with a Haar Wavelet [3]. The amplitude of ψ is $\max(f(t))$.

A1 = Area Covered by the positive half of the Haar Wavelet

A2 = Area Covered by the negative half of the Haar Wavelet

A3 = Positive area spanned by $f(t)$ within the positive half of the wavelet.

A4 = Negative area spanned by $f(t)$ within the negative half of the wavelet

A5 = Negative area spanned by $f(t)$ within the postive scale range of the wavelet

A6 = Positive area spanned by $f(t)$ within the negative scale range of the wavelet

The WaveSim coefficient is computed as follows

$$Sim(f(t), \psi(t)) = \frac{1}{2}((A3 - A5)/A1 + (A4 - A6)/A2) \tag{2}$$

where $(A3/A1)$ is the similarity measure between the positive half of the wavelet and $f(t)$ in the positive scale range, $(A5/A1)$ is the dissimilarity measure between the positive half of the wavelet and $f(t)$ in the positive scale range. Likewise $(A4/A2)$ is the similarity measure between the negative half of the wavelet and $f(t)$ in the negative scale range, $(A6/A2)$ is the dissimilarity measure between the negative half of the wavelet and $f(t)$ in the negative scale range. $\frac{1}{2}$ helps in normalizing the coefficient values to lie in the interval $[0,1]$.

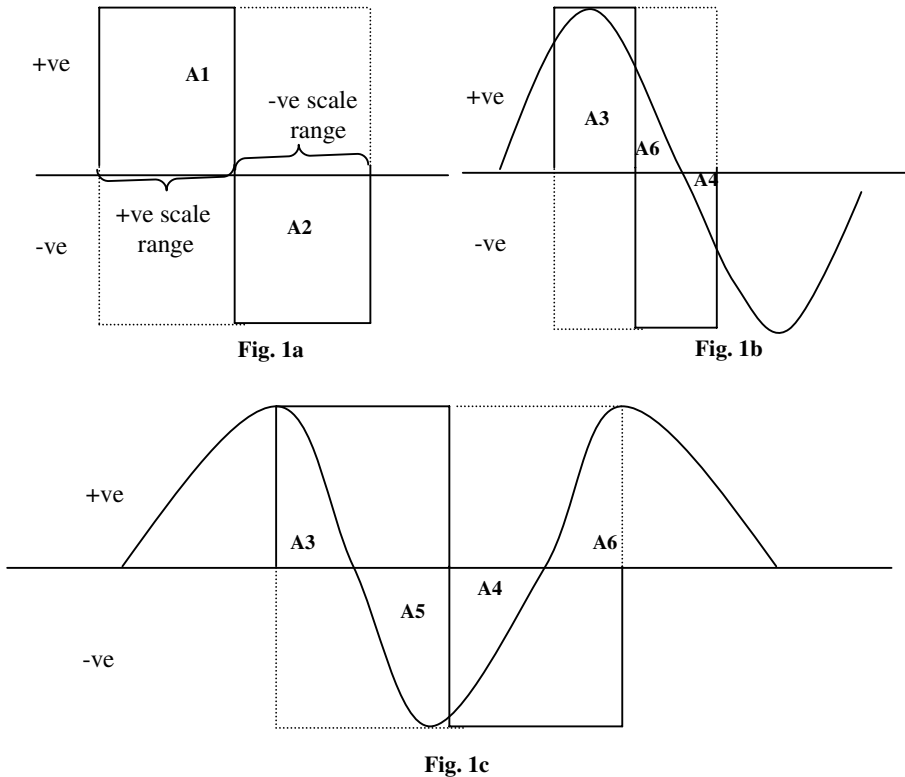


Fig. 1. Illustration of the Areas considered for computing WaveSim *Transform*

WaveSim Transform results in the generation of a normalized set of coefficients with values from 0 to 1. The coefficients are generated at different scales as computed in the case of continuous wavelet transform with a varying translation parameter 'b'. This leads to many interesting inferences. The coefficients generated by WaveSim Transform is a similarity coefficient between the template/wavelet and the function $f(t)$. It is based on the area occupied/spanned by the function $f(t)$ over a template/wavelet. It is not just another kind in the interpretation of Wavelet Transform. This Transform could open up a new arena for research in transforms for pattern recognition applications because, in most transforms the resulting coefficients that are generated are a measure of correlation between the function and the basis function spanning it, resulting out of their inner product. In case of pattern recognition correlation is one of the similarity measures and not the only one. Thus the correlation measure can be replaced by any similarity measure depending on the application. Flexibility to choose the similarity between the function $f(t)$ and the basis function can prove handy for different kinds of applications. The area based similarity measure proposed in this approach can be replaced by any other similarity measure.

2.2 WaveSim Features for Multi-channel Signal Data Mining

Given a source of temporal data, such as the stock market or the monitors in an intensive care unit, there is high utility in determining whether there are qualitatively different regimes in the data and in characterizing those regimes. For example, one might like to know whether the various indicators of a patient's health measured over time are being produced by a patient who is likely to live or one that is likely to die. In this case, there is a priori knowledge of the number of regimes that exist in the data (two), and the regime to which any given time series belongs can be determined post hoc (by simply noting whether the patient lived or died) [4]. However, these two pieces of information are not always present.

Let MD denote a multi-channel data set with the information about an object/sample O_i recorded over N channels. The information of each channel is a signal recorded for 'p' time steps. Given a data set MD consisting of 'm' objects/samples with its temporal signals recorded over N channels and with each signal sampled at 'p' instants we want to obtain, in an unsupervised manner, a partition of these temporally characterized objects/samples into subsets such that, each subset corresponds to a qualitatively different regime. In this section we propose a method for extracting WaveSim features from a multi-channel signal data for clustering. At the end, each signal in the multi-channel data will be characterized by a set of WaveSim regression lines generated at multiple scales. Then clustering can be performed at multiple resolutions with these WaveSim regression features. The coefficients obtained at the dyadic scales (scales at 2, 4, 8, 16, 32, 64 ...) are good enough to represent the unique clusters hidden in the huge data.

Algorithm for WaveSim feature extraction

Step 1: Consider a multi channel data set MD with the information about an object/sample O_i recorded over N channels. The information of each channel is a signal sampled at 'p' time instants.

Step 2: Quantize the time line of the 'p' time instants into k intervals with each interval spanning a time interval of I time units.

Step 3: For each temporal signal S apply WaveSim Transform with a Haar Wavelet (Any wavelet can be used) at Scale Sc_X where X is a dyadic scale value.

Step 4: Sort the WaveSim Coefficients in descending order and capture the corresponding indices of the sorted array.

Step 5: Collect the first P (where $40 < P < 100$) number of indices and determine the interval into which each index fall. Now generate a Histogram which gives the energy sum of the coefficients corresponding to the indices values falling into the k intervals. So each temporal Signal is now represented by a distribution or a histogram with k number of bins.

Step 6: Compute a cumulative histogram from the k -bin histogram.

Step 7: Normalize the cumulative histogram and fit a regression line to characterize the normalized cumulative histogram

Step 8: Repeat Step3 to Step 7 at different dyadic scales starting from $Sc_X = 2$ until $Sc_X < n/2$ where n is the length of the temporal signal.

Thus each object is now represented by 'N' number of regression lines corresponding to each scale i.e. a multi-channel signal data set is transformed into a multi-resolution multi-channel regression line data set.

3 Principal Component Regression Lines

In conventional data analysis, the objects are numerical vectors. The length of such vectors depends upon the number of features recorded, leading to the creation of multidimensional feature space. Symbolic objects are extensions of classical data type. In conventional data sets, the objects are 'individualized', whereas in symbolic objects they are 'unified' by means of relationship. Features characterizing a symbolic object may take more than one value or interval of values or may be qualitative. In real life, quite often we come across features of interval/duration/spread/span/distribution [5].

Recently Pradeep Kumar and Nagabhushan introduced regression line as one of symbolic data object which can reduce the computational complexity of histogram related models drastically [8]. While dealing with SDA data tables has been considered as a challenge there has also been a lot of work carried out for dimensionality reduction of such data tables [9]. All problems become harder as the dimensionality increases. Larger the dimensionality severe is the problems of storage and analysis. The curse of dimensionality refers to the exponential growth of hyper volumes as a function of dimensionality. Hence a lot of importance has been attributed to the process of dimensionality reduction.

Principal components analysis (PCA) is a quantitatively rigorous method for achieving this reduction. In this section we propose a method for computing principal components when the feature is of regression line type. The resultant principal component features are also regression lines. We show that from the first few principal component regression lines we can achieve a good classification of the objects/samples and thus through principal component regression lines we achieve dimensionality reduction.

Problem Statement: There are m samples in n -dimensional space. Each feature f_i of sample j is of symbolic regression line type ie $f_{ij} = L_{ij}$ where $1 \leq j \leq m$ and $1 \leq i \leq n$. It is required to transform the given n -d regression line features f_{ij} to n -d regression line features $F_{ij} = L^*_{ij}$ where $F = T(f)$. Here T represents a feature transformation function which generates the principal components.

Dimensionality Reduction of regression line features: In this section we introduce the basic concept of principal component regression lines. The complete regression line definitions and arithmetic can be referred in [7] for analysis of regression line PCA.

For describing the computational details of principal component method on regression line data set let us consider an original space of 2-d data set D .

$$\text{Let } D = \begin{array}{cc} & \begin{array}{cc} f_1 & f_2 \end{array} \\ \begin{array}{c} S_1 \\ S_2 \end{array} & \begin{array}{cc} L_{11} & L_{12} \\ L_{21} & L_{22} \end{array} \end{array}$$

And let the resultant principal component data set scores be given as

$$\text{PCA Scores} = \begin{array}{cc} & \begin{array}{cc} F_1 & F_2 \end{array} \\ \begin{array}{c} S_1 \\ S_2 \end{array} & \begin{array}{cc} L^*_{11} & L^*_{12} \\ L^*_{21} & L^*_{22} \end{array} \end{array}$$

The variance and covariance of the 2d data set are computed. Let matrix A, be the variance – covariance matrix. The covariance function is defined as

$$A = \text{VarCov}(D) = \begin{bmatrix} E[(L_{11} - \mu_1) (L_{21} - \mu_1)] & E[(L_{21} - \mu_1) (L_{12}-\mu_2)] \\ E[(L_{21} - \mu_1) (L_{12} - \mu_2)] & E[(L_{12} - \mu_2) (L_{22}-\mu_2)] \end{bmatrix} \tag{3}$$

where E is the expectation , μ_1 and μ_2 are mean regression lines of f1 and f2 respectively.

$$\mu_1 = (1/2) * [L_{11} + L_{21}] \text{ and } \mu_2 = (1/2) * [L_{12} + L_{22}] \tag{4}$$

Let K be a column regression line matrix of eigenvectors

$$K = \begin{bmatrix} k1 \\ k2 \end{bmatrix}$$

and λ be its corresponding regression line vector of eigen values.

$$[A] [K] = \lambda [K] \tag{5}$$

Equation (1) can be rewritten as

$$[A - \lambda I] [K] = 0 \tag{6}$$

where I is an identity regression line matrix of the size that of A. The solution of equation (2) can be obtained as follows:

$$\text{DET}(A - \lambda I) = 0 \tag{7}$$

Since we are considering a 2-d data set, i.e. A is 2 x 2 matrix, equation 3 gives 2 quadratic equations (at X(1) and X(2)) in λ . Let the roots of these quadratic equations be λ_{1i} and λ_{2i} corresponding to X(i). Thus we obtain 2 sets of (λ_1, λ_2) . Now corresponding to X(i) substituting λ_{1i} in equation 2 and solving it gives eigen vectors of the form $V1 = a_{11}^i k1(i) + a_{12}^i k2(i)$ where a_{11}^i, a_{12}^i are coefficients of the eigen vector of dimension 1 (F1) corresponding to X(i).

Similarly corresponding to X(i) substituting λ_{2i} in equation 2 and solving it gives eigen vectors of the form $V2 = a_{21}^i k1(i) + a_{22}^i k2(i)$ where a_{21}^i, a_{22}^i are coefficients of the eigen vector of dimension 2 (F2) corresponding to X(i).

Now corresponding to X(i) if the first coefficient of the eigen vector V1 is multiplied with the feature values of dimension 1, second coefficient with the feature values of dimension 2 and combination of the two gives the feature values in dimension 1 of the rotated co-ordinate system.

$$L_{11}^* = a_{11}^i L_{11}(i) + a_{12}^i L_{12}(i) \tag{8}$$

$$L_{12}^* = a_{21}^i L_{11}(i) + a_{22}^i L_{12}(i) \tag{9}$$

Similarly corresponding to X(i), if the first coefficient of the eigen vector V2 is multiplied with the feature values of dimension 1, second coefficient with the feature values of dimension 2 and combination of the two gives the feature values in dimension 2 of the rotated co-ordinate system.

$$L_{21}^* = a_{11}^i L_{21}(i) + a_{12}^i L_{22}(i) \tag{10}$$

$$L_{22}^* = a_{21}^i L_{21}(i) + a_{22}^i L_{22}(i) \tag{11}$$

For n dimensional data the first principal component regression lines represent the large percentage of the total scene variance, succeeding components (pc-2, pc-3,... pc- n) contains a decreasing percentage of the scene variance. Furthermore, because successive components are chosen to be orthogonal to all previous ones, the data are uncorrelated. Most of the time the first few principal components are good enough for classification, thus resulting in dimensionality reduction. In the following section it can be observed that the first principal component regression lines can itself classify the objects into their classes successfully which implies that the entire variance of the 'N' channel signal features are accumulated in one regression line.

4 Experimental Results and Discussions

We have considered multiple electrode time series EEG recordings of control and alcoholic subjects for proving our technique. This data arises from a large study to examine EEG correlates of genetic predisposition to alcoholism (Acknowledgments to Henri Begleiter at the Neurodynamics Laboratory at the State University of New York Health Center at Brooklyn). It contains measurements from 64 electrodes placed on subject's scalps which were sampled at 256 Hz (3.9-msec epoch) for 1 second. There were two groups of subjects: alcoholic and control. Each subject was exposed to either a single stimulus (S1) or to two stimuli (S1 and S2) which were pictures of objects chosen from the 1980 Snodgrass and Vanderwart picture set. When two stimuli were shown, they were presented in either a matched condition where S1 was identical to S2 or in a non-matched condition where S1 differed from S2.

Shown in fig.2a and fig 2b are example plots of a control and alcoholic subjects respectively. Each plot shows the signals recorded through 64 channels the second plot showing clear indications of an alcoholic. We have considered 60 such sets of data with the first 30 belonging to the alcoholic and the second 30 belonging to the non-alcoholic. The results obtained with the first principal component regression lines at wavelet scales 4 and 8 are presented in figure 3.

Discussions. Figure 3 clearly depicts the two regimes in the dataset. The samples marked as 'C' is the cluster of control group and the group marked as 'A' is the cluster of alcoholic group. It also shows under control condition the EEG readings will almost be uniform which is implied by the strong cluster and under alcoholic conditions the EEG readings undergoes different variations for different objects which is implied by the larger span taken for clustering. And the most important is, the above regimes are obtained by the first principal component regression lines. So the information of an object/sample recorded through 64 channels have been accumulated in a single regression line. Mining in multiple resolutions or scales lets us to view different trends of associations between the intra cluster objects and inter-cluster objects. In the EEG data considered, the variation between the control group and the alcoholic group is only in terms of energy of the signal. But in some cases the variation may be due to the behavioral change in the signal for different regimes. So multiple resolution mining helps us in mining the knowledge that is embedded due to both behavioral variation as well as energy variation.

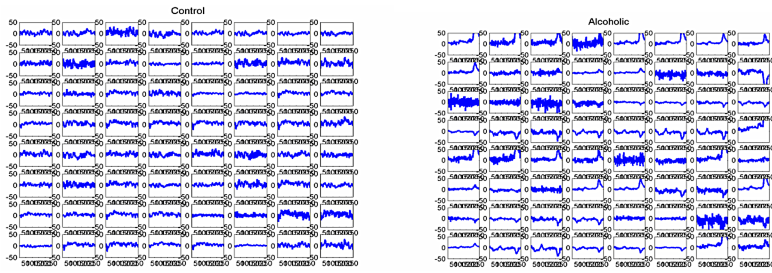


Fig. 2. Plots of signals from 64 channels of a (a) Control object(C) (b) Alcoholic (A)

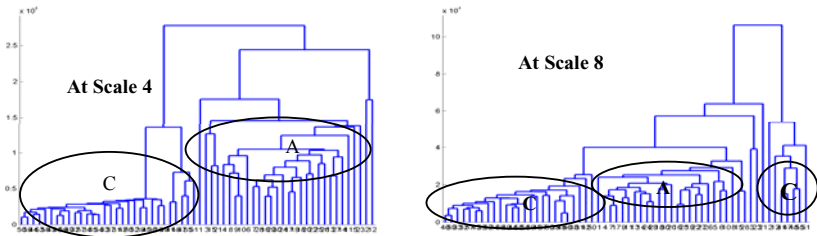


Fig. 3. Clusters depicted through dendrograms obtained from first principal component regression lines

References

1. X.Wang, C. Bettini, A. Brodsky, and S. Jajodia. Logical design for temporal databases with multiple granularities. *ACM Transactions of Database Systems*, 22(2):115–170, June 1997.
2. J.Roddick and K. Hornsby: Temporal, Spatial, and Spatio-Temporal Data Mining. In First Int'l workshop on Temporal, Spatial, and Spatio-Temporal Data Mining 2000
3. R.Pradeep Kumar, P.Nagabhushan. *WaveSim Transform – A New Perspective of Wavelet Transform for Temporal Data Clustering*. IEEE International Conference on Granular Computing, 2006.
4. Tim Oates, Laura Firoiu and Paul R.Cohen, Clustering Time Series with Hidden Markov Models and Dynamic Time Warping, International workshop on Times Series Analysis, 1999.
5. Gowda K.C., Diday E. (1992) "Symbolic clustering using a new similarity measure". IEEE Trans. Syst. Man and Cybernet. 22 (2), 368-378
6. Edwin Diday, An Introduction to Symbolic Data Analysis and Sodas Software, The Electronic Journal of Symbolic Data Analysis,2002, Vol 0.0
7. P.Nagabhushan and R. Pradeep Kumar, Curse of Symbolic Dimensions-Overcoming through Histogram PCA and Regression Line PCA, Communicated to Journal of Symbolic Data Analysis.
8. Nagabhushan, Gowda, Diday, Dimensionality reduction of symbolic data, vol -16. Pattern Recognition Letters, (1995) 219-223.

Research on Query-by-Committee Method of Active Learning and Application

Yue Zhao, Ciwen Xu, and Yongcun Cao

School of Mathematics and Computer Science,
Central University for Nationalities, 100081 Beijing, China
zhaoyueso@sina.com

Abstract. Active learning aims at reducing the number of training examples to be labeled by automatically processing the unlabeled examples, then selecting the most informative ones with respect to a given cost function for a human to label. The major problem is to find the best selection strategy function to quickly reach high classification accuracy. Query-by-Committee (QBC) method of active learning is less computation than other active learning approaches, but its classification accuracy can not achieve the same high as passive learning. In this paper, a new selection strategy for the QBC method is presented by combining Vote Entropy with Kullback-Leibler divergence. Experimental results show that the proposed algorithm is better than previous QBC approach in classification accuracy. It can reach the same accuracy as passive learning with few labeled training examples.

1 Introduction

Obtaining labeled training examples for some classification tasks is often expensive, such as text classification, mail filtering, credit classification and et al., while gathering large quantities of unlabeled examples is usually very cheap. For example, the available cases with actual classes are not enough for building credit classification model in practice, especially for the newly established system in which old customers' data do not exist. In this situation, one solution is the manual classification. The credit customers are evaluated by experts and classified into different risk levels. It is time-consuming and costly. Thus, active learning can reduce annotation cost by sample selection. QBC is a kind of sample selection method of active learning. It is less computation than the one based on Error Reduction Sampling, but existing QBC approaches do not reach the same classification accuracy as passive learning [1-3]. In this paper, the present selection strategy for QBC attempts to provide a solution for this problem.

We consider two selection functions for measuring the disagreement among committee members in QBC. One uses Kullback-Leibler divergence (KL-d) to the mean for capturing the information [4]. One disadvantage of KL divergence is that it misses some examples on which committee members disagree, but these examples are exactly needed by QBC. The other selection function measures the disagreement by Vote Entropy (VE) [5]. The disadvantage of Vote Entropy is that it does not consider the committee members' class distributions, $P_m(C | e_i)$. Each committee member

m produces a posterior class distribution, $P_m(C | e_i)$, where C is a random variable over classes and e_i is an input unlabeled example. So, Vote Entropy also misses some informative unlabeled examples to label. Because both of them above do not select enough useful examples, they can not achieve the same classification accuracy as passive learning.

We propose a new select strategy for QBC by combining the Vote Entropy with Kullback-Leibler divergence to improve the classification accuracy with much fewer training data than passive learning.

2 The Query-by-Committee Method of Active Learning

The Query-by-Committee method of active learning examines unlabeled examples and selects only those that are most informative for labeling. This avoids redundant labeling examples that contribute little new information. Our research follows theoretical work on sample selection in the Query-by-Committee paradigm [5]. In this committee-based selection scheme, the learning receives a stream of unlabeled examples as input and decides for each of them whether to ask for its label or not. To that end, the learner constructs a ‘committee’ of (two or more) classifiers based on the statistics of the current training set. Each committee member then classifies the candidate example and the learner measures the degree of disagreement among the committee members. The example is selected for labeling depending on this degree of disagreement according to some selection protocol. Its algorithm is available in [5].

There are two selection functions for QBC to measure the disagreement. One uses Kullback-Leibler divergence to the mean for capturing the information of disagreement. KL-d measures the strength of the certainty of disagreement by calculating differences in the committee members’ class distributions. KL-d to the mean is an average of the KL divergence between each distribution and the mean of all distribution:

$$\frac{1}{K} \sum_{m=1}^K D(P_m(C | e_i) \| P_{avg}(C | e_i)), \tag{1}$$

where $P_{avg}(C | e_i)$ is the class distribution mean over all committee members, m :

$$P_{avg}(C | e_i) = (\sum_n P_n(C | e_i)) / K. \tag{2}$$

KL divergence, $D(\bullet \| \bullet)$, is an information-theoretic measure of the difference between two distributions. It is:

$$D(P_1(C) \| P_2(C)) = \sum_{j=1}^{|C|} P_1(c_j) \log \left(\frac{P_1(c_j)}{P_2(c_j)} \right). \tag{3}$$

The other selection function measures disagreement by the entropy of the distribution of classification ‘voted for’ by the committee members. This Vote Entropy is

natural measure for quantifying the uniformity of classes assigned to an example by the different committee member. It is:

$$-\frac{1}{\log \min(K, |C|)} \sum_c \frac{V(c, e_i)}{K} \log \frac{V(c, e_i)}{K}, \tag{4}$$

where $V(c, e_i)$ denotes the number of committee members assigning a class c for e_i and K is the number of committee members.

One disadvantage of KL-d is that it misses some examples on which committee members disagree, but these examples are exactly needed by QBC. An illustrative experiment of learning the binary classification task is presented in Table 1 and 2. Supposed the 2-member (two classification model) committee is generated and 3 examples need to be decided whether to ask for its label or not. If we compare the Vote Entropy (VE) of e_1 with e_2 in Table 2, we see that they are both selected for labeling (when the degree of disagreement is most, VE is 1 for binary classification. If committee is unanimous for an example, VE is zero.). But their KL-divergence is quite different and only e_2 is selected. It shows that KL-d misses the examples which are very informative.

Table 1. The results of committee members' class vote for unlabeled examples

Model	e_1	e_2	e_3
1	0.52(c_1)	0.72(c_2)	0.60(c_2)
2	0.58(c_2)	0.60(c_1)	0.70(c_2)

Table 2. The Kullback-Leibler divergence and Vote Entropy of examples

Example	VE	KL-d
e_1	1	0.005(miss)
e_2	1	0.052
e_3	0	0.006

However, one disadvantage of Vote Entropy is that it does not consider the committee members' classifications distributions, $P_m(C | e_i)$, according the formula (4). It can also miss informative examples.

Because both of them above do not select enough useful examples, they can not achieve the same classification accuracy as passive learning.

3 A New Algorithm for QBC

Through discussing the disadvantages of exist selection functions of QBC in section 2, we consider to combine the Vote Entropy with Kullback-Leibler divergence by selecting some examples which committee agree on and have high uncertainty for a member of committee. The degree of disagreement among the committee members is

measured by Vote Entropy and each example’s uncertainty of classification is measured by KL-d. So we redefine KL-d. Let it measure the difference between $P_m(C | e_i)$ and $P_{m_avg}(C | e_i)$ which are class probability distributions in the most uncertainty. The minimum of KL-d among committee members for each unlabeled example is taken as an example’s the most uncertainty from committee members, denoted by $KL-d_{min}$. If $KL-d_{min}$ of an example satisfies the term of some threshold α (α is near to zero.), this example is selected for labeling and being added into training data set. These selected examples are informative and can contribute to improving classification accuracy [6]. $KL-d_{min}$ is:

$$KL-d_{min}(e_i) = \underset{m=1}{MIN}^K (\sum_{j=1}^{lcl} P_m(c_j | e_i) \log(\frac{P_m(c_j | e_i)}{P_{m_avg}(C | e_i)})), \tag{5}$$

where

$$P_{m_avg}(C | e_i) = (\sum_{j=1}^{lcl} P_m(c_j | e_i)) / l, \tag{6}$$

$$l = \begin{cases} 2, \max(P_m(c_j | e_i)) = 1 \\ (lcl - num_zero), \max(P_m(c_j | e_i)) \neq 1 \end{cases}, \tag{7}$$

num_zero is the number of the class probabilities which are zeros.

The other experiment is presented in Table 3 and 4. It shows that the example of e_1 is missed by VE, but it has the same high classification uncertainty as e_2 and e_3 by $KL-d_{min}$. It is helpful for building classification model by QBC.

Table 3. The results of committee members’ class vote for unlabeled examples

Model	e_1	e_2	e_3	e_4
1	0.55(c_1)	0.55(c_2)	0.52(c_1)	0.80(c_2)
2	0.55(c_1)	0.55(c_2)	0.75(c_2)	0.90(c_2)
3	0.60(c_1)	0.55(c_1)	0.85(c_2)	0.75(c_2)
4	0.60(c_1)	0.55(c_1)	0.95(c_2)	0.85(c_2)

We combine the Vote Entropy with $KL-d_{min}$ to propose our algorithm for QBC. When an example is agreed by committee members, it is measured by $KL-d_{min}$ further. If its $KL-d_{min}$ satisfies some threshold α , it is selected for labeling and added into training data set. The new algorithm is described as follows.

Table 4. The $KL-d_{min}$ and Vote Entropy of examples

Example	VE	KL-d
e_1	0.0(miss)	0.005
e_2	1.0	0.005
e_3	0.81	0
e_4	0.0	0.1308

A new algorithm for QBC

Input: Classification algorithm: A
 The number of committee members: K
 Few labeled examples: L
 Unlabeled examples: UL
 The condition of stopping: ζ
 The threshold of Vote Entropy: θ
 The threshold of $KL-d_{\min}$: α

1. Learn K classifiers $\{M_h\}$ from L using A ;
2. While not ζ
 - { 1) $\forall e_i \in UL$, for $h=1, \dots, K$:
 Classify e_i using M_h to get class label C_h ;
 - 2) using formula (4) to Compute $VE(e_i)$;
 - 3) If $VE(e_i) > \theta$,
 Select e_i from UL , get true label and add e_i to L , learn K classifiers from L using A again;
 - Else
 Compute $KL-d_{\min}(e_i)$;
 If $KL-d_{\min}(e_i)$ satisfies α , select e_i from UL , get true label and add e_i to L , learn K classifiers from L using A again;
 - End
 - 4) Check the condition of stopping ζ ;
3. Learn classifier M from L using A ;

Output: Classifier M .

4 Experimental Results

We now discuss the results of our experiments on Nursery database and Tic-Tac-Toe Endgame database from the UCI Machine Learning Repository.

We randomly sample 4171 instances from Nursery database. The data set is randomly partitioned into labeled examples set, unlabeled examples set and independent test set. Each instance contains 9 attributes.

Tic-Tac-Toe Endgame database consist of 958 instances. Each instance contains 10 attributes. The data set is also randomly partitioned into labeled examples set, unlabeled examples set and independent test set. The class attribute has 2 values. About 65.3% are positive of class distribution.

We choose the TAN classifier [7] as classification algorithm. For QBC, we use a committee size of two. The condition of stopping ζ is that classification accuracy reaches the expected value or unlabeled data set is empty.

Fig.1 plots the learning curves obtained from the 3 learning methods—VE of QBC, VE& $KL-d_{\min}$ of QBC and passive learning—on Nursery database. It is clearly

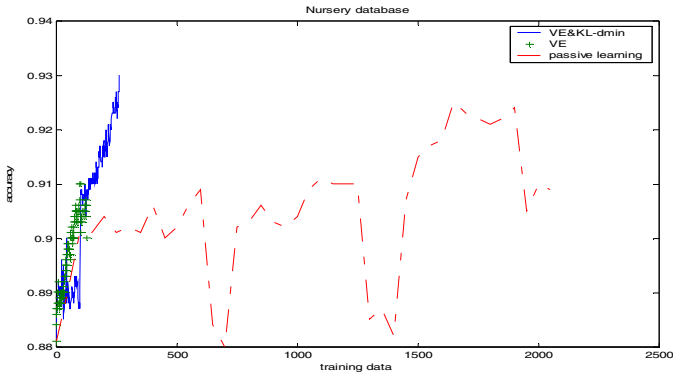


Fig. 1. The classification accuracy comparison of VE, VE&KL-d_{min} and passive learning on Nursery database

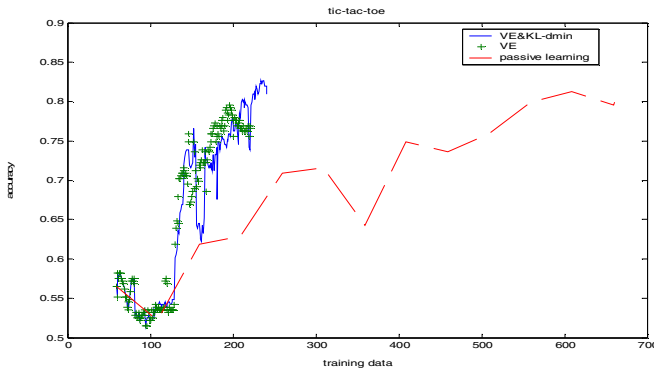


Fig. 2. The classification accuracy comparison of VE, VE&KL-d_{min} and passive learning on Tic-Tac-Toe Endgame database

seen from these graphs that the VE of QBC achieves the accuracy of 91% after selecting 128 unlabeled examples, VE&KL-d_{min} achieves 93% after 264 unlabeled examples and passive learning achieves 92.51% after all unlabeled data.

Fig.2 plots the learning curves on Tic-Tac-Toe Endgame database. It is clearly seen from these graphs that the VE achieves the accuracy of 79% after selecting 138 unlabeled examples, VE&KL-d_{min} achieves 83% after 176 unlabeled examples and passive learning achieves 81% after all unlabeled data.

A Chinese credit rating data set for telecom clients was collected from Jan. to May, 2001. It consists of 33512 instances. Each instance contains 15 attributes and one credit class attribute which has 4 values of credit risk levels. Fig.3 plots the learning curves on the credit scoring data set. It is clearly seen from these graphs that the VE achieves the accuracy of 81% after selecting 278 unlabeled examples, VE&KL-d_{min} achieves 84% after 921 unlabeled examples and passive learning achieves 84% after all unlabeled data.

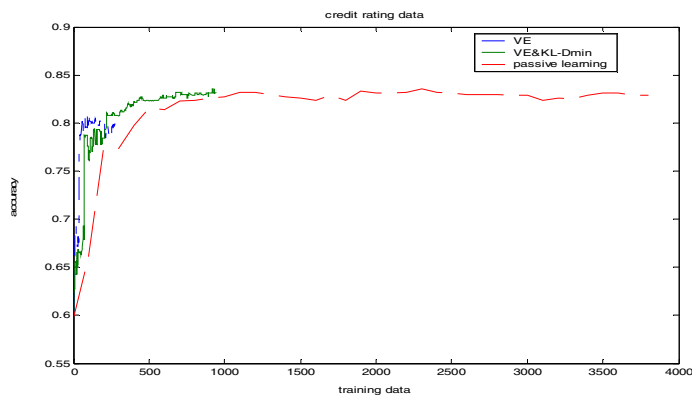


Fig. 3. The classification accuracy comparison of VE, VE&KL- d_{\min} and passive learning on Chinese credit scoring data

The results show that the VE&KL- d_{\min} is better than previous VE of QBC. It can reach the same accuracy as passive learning with few labeled examples. The selected newly examples contribute to improve classifier's accuracy.

5 Summary

In this paper, we present a new selection strategy for QBC to improve the accuracy using Vote Entropy and Kullback-Leibler divergence. The experimental results indicate that the proposed algorithm is better than previous QBC in classification accuracy with much fewer labeled examples than passive learning.

References

1. Freund, Y., Seung, H.S., Samir, E., Tishby, N.: Selective Sampling Using the Query by Committee Algorithm. *Machine Learning*, 28(1997)133-168
2. Gong, X.J., Shun, J.P., Shi, Z.Z.: An Active Bayesian Network Classifier. *Computer research and development*, 39 (2002)574-579
3. Riccardi, G., Hakkani-Tür, D.: Active Learning: Theory and Applications to Automatic Speech Recognition. *IEEE Transaction on Speech and Audio Processing*, 13 (2005)504-511
4. McCallum, A. K., Nigam, K.: Employing EM and Pool-based Active Learning for Text Classification. In: *Proceeding of the 15th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco Madison (1998) 350—358
5. Argamon-Engleson, S., Dagan, I.: Committee-based Sample Selection for Probabilistic Classifiers. *Journal of Artificial Intelligence Research*, 11 (1999)335-460
6. Lewis, D.D., Gale, W.A.: A Sequential Algorithm for Training Text Classifiers. In: *Proceedings of {SIGIR}-94, 17th {ACM} International Conference on Research and Development in Information Retrieval*, Springer-Verlag, Berlin Heidelberg Dublin (1994)3-12
7. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers. *Machine Learning*, 29(1997)131-161

Traffic Management Genetic Algorithm Supporting Data Mining and QoS in Sensor Networks

Yantao Pan, Wei Peng, and Xicheng Lu

School of Computer, National University of Defense Technology, Changsha, P.R. China
pytmail@126.com

Abstract. Sensor networks are expected to be used for spatial cognition in harsh environments. When an event happens, there will be several sensors detect it and send their reports to a sink, but these data are neither integrated nor reliable. Therefore, it is reasonable to make use of data mining on intermediate nodes to acquire deeper knowledge on an event and cut down the total traffic at the same time. Furthermore, the QoS requests should be considered too. In this paper, we propose a centralized algorithm to achieve optimal traffic management on sensor networks with considering QoS and data fusion. Its efficiency is shown by experiments.

1 Introduction

Consider a number of wireless static sensor nodes randomly distributed in a region. Each node has a limited battery energy supply which is mainly used for receiving and sending data, and the data throughput of a node is also limited. When an event happens in the sensing field of a node, a report will be generated to describe the kind of the event and the node's confidence level. These reports are gathered from network and sent to corresponding sinks. Because the nodes are supposed to be densely distributed, several nodes maybe detect the same event. Reports are generated to describe the different aspects of it and their confidence levels are presented. This confidence level depends on the distance from a node to the event and other factors. However, these raw data are hard to understand and utilize. Since these data are spatially and temporally correlated, an intermediate node could apply data mining to achieve deeper knowledge about the events. In addition, QoS requirements are necessary for many applications. For example, an upper bound should be set to the hop-count, which a report could mostly take before it arrives the sinks. At the same time, throughput constraints should be considered to prevent traffics from blocking critical nodes.

In this scenario, limited energy is still the scarcest resource, so it is a key challenge to save energy and prolong the network lifetime. Wireless communication consumes the most energy. So it is efficient to make use of traffic management to save energy. If the nodes' communication powers remain the same, we can find an optimal traffic management in polynomial time [1]. However, if radios can use different power level to communicate with different neighbors, the traffic management optimization problem becomes very hard, which is reported as NP-complete in [2].

The traffic management optimization problem has been addressed before. The literatures [3-5] investigate the upper bounds or expected value of the maximum lifetime. In [6] and [1], Chang et al. provide a heuristic algorithm. In the literature [7-9], other heuristic approaches are presented. However, these works do not take energy consumed by receiving data, QoS, and data fusion into account. Actually, the energy consumed by receiving data should not be ignored in most cases. In literature [10-12], the powers of RX and TX units are reported as in the same order of magnitude. QoS requirements are also important for many applications such as disaster detection. Furthermore, the affection of data fusion should be considered in sensor networks with data mining.

In this paper, we investigate the lifetime optimization problem with considering energy consumption of data transmission, QoS, and data fusion in the process of data mining. Genetic computation is chosen to solve this complex optimization problem.

The rest of this paper is organized as follows. In section 2, the problem is formulated. In section 3, our approach is described. In section 4, experiments results are given. Finally in section 5, some concluding remarks are made.

2 Problem Formulation

A sensor network in consideration is modeled as $N^s = (G, p, q, w, o, h, X, Y)$. $G(V, A)$ is a connected directed graph, where V is the set of nodes and A is the set of directed links. Each node u has initial energy E_u . Let $p(u)$ be the residual energy of node u . Let $q(u, v)_s$ and $q(u, v)_r$ denote the energies required by node u to send and receive an information unit to and from v . According to the probability distribution of events, each node has a data-generating rate w . Let $o(u)$ denote the limited data throughput at node u . Let $h(u)$ be the upper bound of hop-count that a report from node u can take. In addition, we denote the source and sink sets as X and Y .

A virtual traffic management scheme is defined to be $\delta_a^x : (X, A) \mapsto \overline{R^-}$, if

$$\sum_{v \in V} \delta_{(x,v)}^x - \sum_{v \in V} \delta_{(v,x)}^x = w(x) \text{ for } \forall x \in X ; \tag{1}$$

$$\sum_{y \in Y \setminus \{x\}} \left(\sum_{v \in V} \delta_{(v,y)}^x - \sum_{v \in V} \delta_{(y,v)}^x \right) = w(x) \text{ for } \forall x \in X ; \tag{2}$$

$$\sum_{u \in V} \delta_{(v,u)}^x = \sum_{u \in V} \delta_{(u,v)}^x \text{ for } \forall x \in X, \forall v \in V \setminus (X \cup Y) ; \tag{3}$$

$$\sum_{v \in V} \max_{x \in X} \delta_{(v,u)}^x + \sum_{v \in V} \max_{x \in X} \delta_{(u,v)}^x \leq o(u) \text{ for } \forall u \in V ; \tag{4}$$

$$\text{the total hops from } x \text{ to its sink} \leq h(x) \text{ for } \forall x \in X . \tag{5}$$

Formula (6) defines the lifetime of N^s under the virtual traffic management scheme δ_a^x . If there is no $\delta_a^{x'}$ and its corresponding lifetime T' so as to $T' > T$, δ_a^x is an

optimal virtual traffic management scheme and T is the maximum lifetime. The problem is stated as follows. Given a sensor network $N^s = (G, p, q, w, o, h, X, Y)$, ask for an optimal virtual traffic management scheme $\delta_a^{x^*}$ and the maximum lifetime T^* .

$$T \triangleq \min_{u \in V} \left(p(u) / \left(\sum_{v \in V} q(v, u)_r \cdot \max_{x \in X} \delta_{(v, u)}^x + \sum_{v \in V} q(u, v)_s \cdot \max_{x \in X} \delta_{(u, v)}^x \right) \right). \tag{6}$$

3 Optimal Traffic Management

In this section, a genetic algorithm is proposed to solve the lifetime optimization problem. Both data fusion and QoS are considered.

Consider a sensor network with maximum lifetime T^* . In the period of lifetime, node u generates $w(u) \cdot T^*$ packets, and could take $w(u) \cdot T^*$ paths at most to balance its original data flow. An individual is defined to be a set of chromosomes. The number of chromosomes depends on the number of sources. A chromosome contains a set of unit paths as genes. Each takes a unit virtual flow velocity from a source to its sink. Furthermore, we define a parameter PREC to control the number of paths that a source can take. The number is determined by $\lfloor w(u) \cdot \text{PREC} \rfloor$.

The performance of a genetic algorithm greatly depends on the fitness of initial population. We generate initial population by taking random unit paths as genes. The randomness of this method is in favor of achieving high community diversity. However, the fitness of an initial population might be very low. This disadvantage brings more bad effects on the performance while PREC increases. Since the number of genes increases with the PREC, the population size contrastively becomes smaller. Therefore, a population size that offers sufficient community diversity cannot work well under a PREC much higher than before. Of course, we might use a large population size to improve the community diversity. But it is not effective. In fact, the algorithm obtains little improvement with increased population sizes, especially when PREC is high. Another method is to generate initial population with high fitness to improve the performance. We use the final population under a lower PREC to generate an initial population under a higher PREC. At the beginning, a few paths are taken by each node and a small population size is enough to offer sufficient community diversity. While PREC increases, a node might redistribute its original flow velocity to more paths basing on the earlier assignments. In this way, we get an initial population with high fitness and a small population size works well even under a high PREC.

The crossover operation includes two steps. The first step is to choose individuals according to parameter CP and individuals' fitness. We use CP to control the total number of individuals that will take part in crossover operation. However, the opportunity of each individual depends on its fitness. The second step is to crossover two individuals in each chromosome. Here, we use parameter CN to control the number of points that two chromosomes will take crossover in. The mutation operation is similar. First step is to choose individuals according to MPROB and fitness. The second step is to choose random genes according to MPER and replace them with new generated ones.

In order to evaluate the fitness of an individual, we calculate the lifetime and a punishment function. The network lifetime depends on the most short-lived node. Assume that energy is mainly used for receiving and sending data. Suppose that two events do not happen at the same time. An intermediate node can therefore aggregate input packets into one event report and assign a new confidence level for it. When an event happens, some nodes will detect it and others will not. This depends on the distribution of the nodes and the events. From long views, the nodes have different data-generating rates. The actual flow velocity depends on the maximum virtual traffic on each link. The lifetime is calculated by formula (6).

Additionally, we define a punishment function to support QoS. If an individual violates the throughput constraint, a punishment will be added to the evaluation of its fitness. In the process of initial population generation and the mutation operation, all unit paths are generated following the hop-count constraint. However, we are not worried about whether an individual violates the energy constraint, because we code the flow velocities instead of the flows. The punishment is calculated by formula (7).

$$P \triangleq \sum_{u \in V} \varphi(u), \text{ where } \varphi(u) = \begin{cases} 0, & o(u) \geq \left(\sum_{v \in V} \max_{x \in X} \delta_{(v,u)}^x + \sum_{v \in V} \max_{x \in X} \delta_{(u,v)}^x \right), \\ o(u) - \left(\sum_{v \in V} \max_{x \in X} \delta_{(v,u)}^x + \sum_{v \in V} \max_{x \in X} \delta_{(u,v)}^x \right), & \text{otherwise.} \end{cases} \quad (7)$$

Because the lifetime and the punishment have different units, it is reasonable to standardize them to values between 0 and 1 according to the average lifetime and the average punishment of a population. We denote standardized lifetime and punishment as T' and P' .

The fitness of an individual is calculated by $F = \alpha \cdot T' + (1 - \alpha) \cdot P'$, where α is a parameter that is set to balance the affections of the lifetime and the punishment.

4 Experiment Results

Firstly, we take experiments to evaluate the performance of our approach under a designed network topology. From these experiments, we get a set of parameters as the base of further evaluation. Secondly, we take experiments on random generated networks.

Consider a designed sensor network with 3 sources, 2 relay nodes and a sink. We set the initial energy of each node and the energy consumption of sending or receiving an information unit intentionally. The maximum lifetime of this network is 1.8, which is called ideal solution.

Fig. 1 shows how PS affects on the algorithm performance. All results are average values from 200 times calculations. Fig. 1.1 plots the best solution as a function of PS. Fig. 1.2 shows the generations, which is necessary to achieve the ideal solution in a probability not less than 90%. Fig. 1.3 shows the probability of achieving the ideal solution in a single time calculation. Fig. 1.4 shows the probability, in which one time calculation can achieve the ideal solution in each generation, with PS ranging from 50 to 120. The value on generation 0 denotes the probability in which one time calculation cannot achieve the ideal solution in any generation.

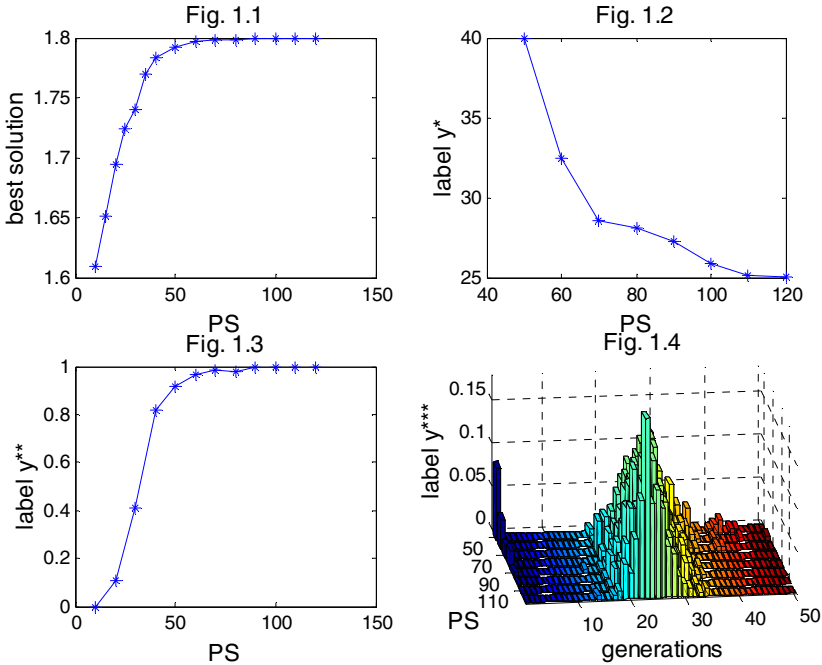


Fig. 1. The performance of our approach as functions of PS. *Generations needed to achieve ideal solution in a probability not less than 90%. **Probability to achieve ideal solution in 200 times computation. ***Probability to achieve ideal solution in each generation.

Fig. 2.1 shows the best solution as a function of MPER. The best solution achieves 1.799571 when MPER takes 0.15, and holds to be a high value when MPER keeps go on. Fig. 2.2 shows the affection of MPROB on the best solution. Two lines are got from difference parameters. The PS, MPER, and CN take 70, 0.25, and 2 in configuration-1, but take 60, 0.15, and 4 in configuration-2. Fig. 2.3 shows the affection of CP on the best solution. When CP takes 0.40, the best solution achieves 1.797857. Fig. 2.4 shows the best solution as a function of CN. The best solution achieves the ideal solution when MPER takes 4.

From these numeric experiences, we get a set of parameter values: MPER = 0.15; MPROB = 0.95; CP = 0.40; CN = 4. In the reset of this paper, all experiments are based on this configuration.

We take further experiments to investigate the affections of hop-count and throughput constraints on the lifetime of sensor networks. We generate a network with 20 nodes (including a sink). They are distributed in a 100×100 square area randomly. An initial energy (chosen from 400 to 800 randomly) and a data-generating rate (chosen from 5 to 10 randomly) are assigned to each node. A receiving price and a sending price are assigned to each link according to its length.

As shown in Fig. 4, traffic management schedules are plotted under different hop-count constraints, where the node in the shape of a diamond is the sink. We could observe that the schedule under a tight constraint is simpler but more short-lived than that under a loose constraint.

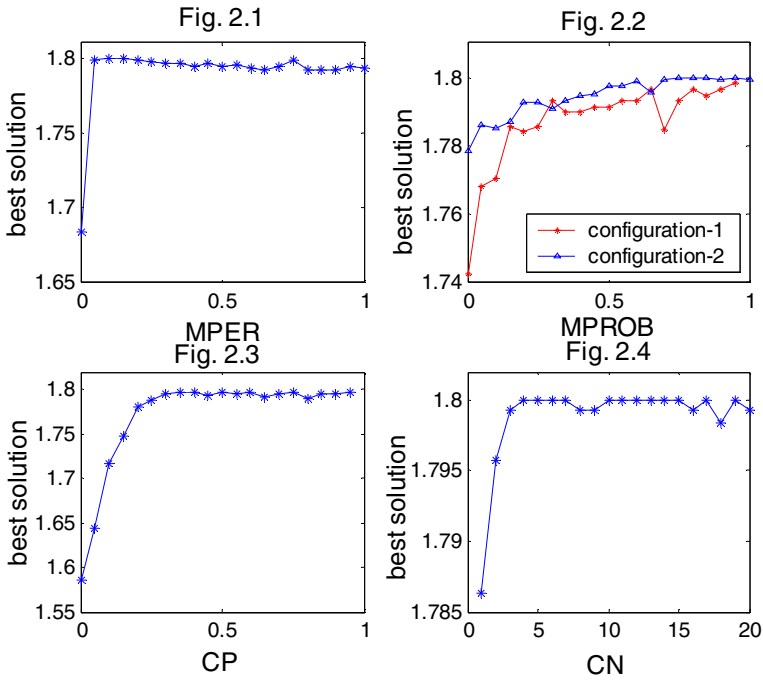


Fig. 2. The best solutions as functions of MPER, MPROB, CP and CN

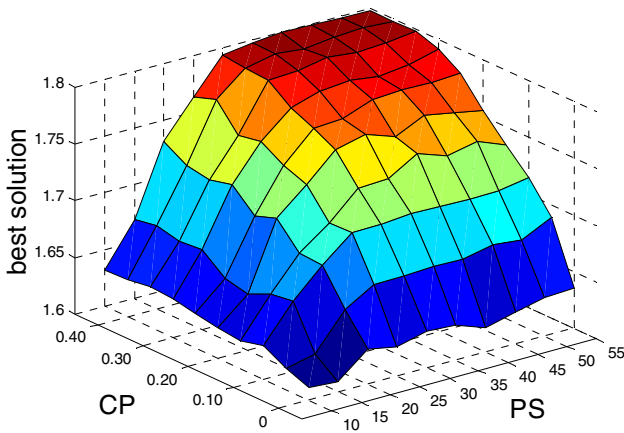


Fig. 3. The best solution as a function of PS and CP

Fig. 5.1 plots a traffic management schedule where the throughput of every node is set to 65. It is obvious that the traffics shown in Fig. 5.1 are distributed equably. Contrastively, the traffics shown in Fig 5.2 are not even, where the throughput constraint is not considered.

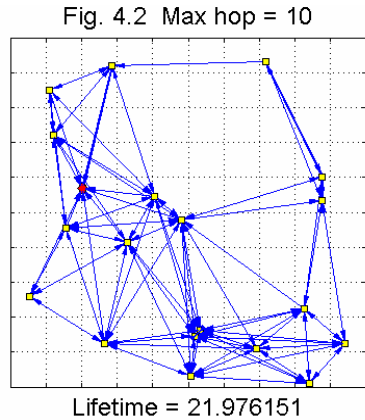
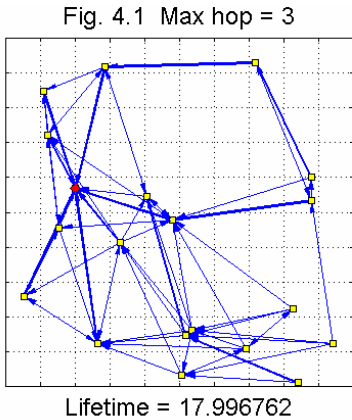


Fig. 4. Lifetimes under different hop-count constraints in a 20 nodes random network

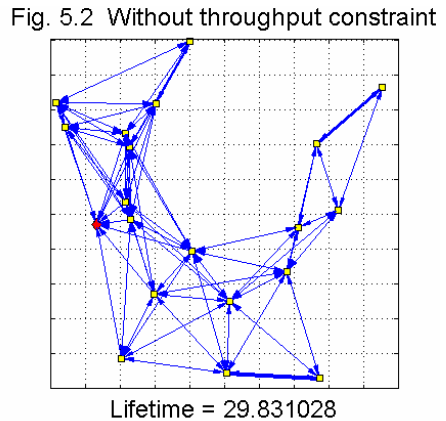
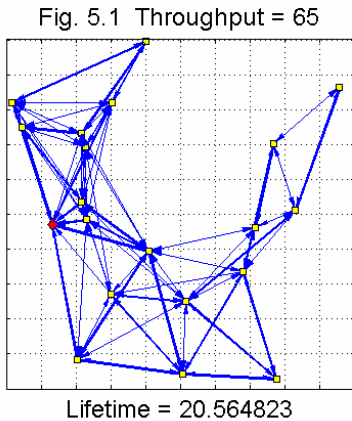


Fig. 5. Lifetimes under different throughput constraints in a 20 nodes random network

5 Conclusion

Wireless sensor networks are energy constrained. One of key challenges of sensor networks is maximizing the lifetime. It is a hard work and many heuristic algorithms are proposed. However, few of them take data fusion and QoS into account. In this paper, we propose a genetic algorithm to solve this problem. It supports data mining and QoS. The complexity of our approach is $RANSMG$, where R is the maximum $PREC$, A is the maximum data-generating rate, N is the population size, S is the source number, M is the arc number, and G is the generation number.

We have assumed that two events would not happen at the same time, so an intermediate node could fusion all input reports into one output by data mining. However, this is not a key assumption. An intermediate node might generate more output packets to describe several simultaneous events and our approach still performs well after a little modified.

References

1. J.-H. Chang, L. Tassiulas: Energy conserving routing in wireless ad-hoc networks. IEEE INFOCOM'2000.
2. S. Singh, M. Woo, C.S. Raghavendra: Power-aware routing in mobile ad hoc networks. Proceedings of Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, Dallas, TX, Oct. 1998.
3. M. Bhardwaj, A. Chandrakasan: Bounding the Lifetime of Sensor Networks Via Optimal Role Assignments. IEEE INFOCOM'2002.
4. E. J. Duarte-Melo, M. Liu, A. Misra: A Modeling Framework for Computing Lifetime and Information Capacity in Wireless Sensor Networks. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, Cambridge, UK, March 2004.
5. Vivek Rai, Rabi N. Mahapatra: Lifetime Modeling of a Sensor Network. Design, Automation and Test in Europe, Munich, Germany, March 2005.
6. J.-H. Chang, L. Tassiulas: Routing for Maximum System Lifetime in Wireless Ad-hoc Networks. 37th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 1999.
7. K. Dasgupta, K. Kalpakis, P. Namjoshi: Efficient Algorithms for Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks. Computer Networks, vol. 42, 2003.
8. R. Madan, S. Lall: Distributed Algorithms for Maximum Lifetime Routing in Wireless Sensor Networks. Global Telecommunications Conference, IEEE, volume 2, Nov 2004.
9. Y. Xue, Y. Cui, K. Nahrstedt: Maximizing Lifetime for Data Aggregation in Wireless Sensor Networks. <http://cairo.cs.uiuc.edu/publications/paper-files/xue-monet.pdf>.
10. D. Estrin, M. Srivastava: Wireless Sensor Networks (Tutorial). Proceedings of ACM MobiCom'02, Atlanta, Georgia, USA, 2002.
11. J.J. Garcia-Luna-Aceves, C.L. Fullmer, E. Madruga: Wireless mobile internetworking. Manuscript.
12. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson: Wireless Sensor Networks for Habitat Monitoring. ACM International Workshop on Wireless Sensor Networks and Applications, 2002.

Comparison of Data Pre-processing in Pattern Recognition of Milk Powder Vis/NIR Spectra

Haiyan Cen, Yidan Bao, Min Huang, and Yong He

College of Biosystems Engineering and Food Science
Zhejiang University, 310029, Hangzhou, China
ydbao@zju.edu.cn, yhe@zju.edu.cn

Abstract. The effect of data pre-processing, including standard normal variate transformation (SNV), Savitzky-Golay first derivative transformation (S. Golay 1st-Der) and wavelet transforms (WT) on the identification of infant milk powder varieties were investigated. The potential of visible and near infrared spectroscopy (Vis/NIRS) for its ability to nondestructively differentiate infant formula milk powder varieties was evaluated. A total of 270 milk powder samples (30 for each variety) were selected for Vis/NIRS on 325-1075 nm using a field spectroradiometer. Partial least squares (PLS) analysis was performed on the processed spectral data. In terms of the total classification results, the model with the wavelet transforms processed data is the best, and its prediction statistical parameters were r^2 of 0.978, SEP of 0.435 and RMSEP of 0.413. This research shows that visible and near infrared reflectance spectroscopy has the potential to be used for discrimination of milk powder varieties, and a suitable pre-processing method should be selected for spectrum data analysis.

1 Introduction

Milk products are among the most sensitive of food industry due to the important status in people's daily life. The milk powder, especially of infant formula milk powder has attracted great attention of the whole society. Its quality is defined by the scientific distributed compositions, such as linoleic acid, protein, carbohydrate, vitamins, and other necessary element for the growth of neonates and infants. Consumers expect manufactures and retailers to provide products with high quality, security and good variety. These factors have underlined the need for reliable techniques that can authenticate the varieties of infant formula milk powder.

Visible and near infrared reflectance spectroscopy (Vis/NIRS) is a well-established technique for constituent analysis of agricultural and food products [1][2]. Nondestructive optical method based on Vis/NIR spectroscopy has been evaluated for nondestructive estimation of moisture, fat, protein, total solids in cheese and elements in milk [3]. Sorensen et al. [4] published their results in assessment of sensory properties of cheese by near infrared spectroscopy. Hermida et al. [5] analyzed moisture, solids-non-fat and fat in butter by NIR spectroscopy. The important problem about the selection of pre-processing methods should be

discussed before analyzing Vis/NIRS data. The spectra were recorded in a broad wavelength range and subject to large baseline shifts. This problem especially appears in the case of solid powdered samples due to the differences of the size and color of individuals [6]. Thus, the selection of a suitable pre-processing method is an important step in the process of building models.

The object of this study was to assess the potential of visible and NIR spectroscopy to distinguish infant formula milk powder varieties. The partial least square (PLS) models with different pre-processing methods were evaluated and three pre-processing methods for the model of classification in the varieties of infant formula milk powder were investigated.

2 Materials and Methods

2.1 Samples and Reflectance Measurement

Nine different varieties of milk powder were selected for visible and NIR spectral characteristics. 270 samples were gained from the local market and each species in 30 samples, which are Neatle Nestogen, Yili, Sanlu, Guoli, Beinmate, Dumex Dulac, Syntra, Youbo Saint and Abbott Premilac. All of the infant formula milk powder was suitable for babies from 6 months to 1 year. The samples were placed in airtight plastic bags, and each sample was labeled.

Milk powder samples were extended upperly throughout the bottom surface of glass sample containers with 95 mm diameter and 10mm height. 30 similar samples of each variety (270 totals) were prepared for visible and NIRS analysis on 325–1075 nm. For each sample, three reflectance spectra were taken for three equidistant rotation positions of approximately 120° around the container centre with a field spectroradiometer (FieldSpec Pro FR (325–1075 nm)/ A110070). The scan number for each spectrum was set 10 at exactly the same position. Considering its 25° field-of-view (FOV), the spectroradiometer was placed at a height of approximately 10 cm above the sample container, and a light source of Lowell pro-lam 14.5V Bulb/128690 tungsten halogen was placed about 300 mm from the center of the container to make the angle between the incident light and the detector optimally about 45° .

2.2 Pre-processing of the Optical Data

Due to the potential system imperfection, obvious scattering noises could be observed at the beginning and end of the spectral data. Thus, the first and last 75 wavelength data were eliminated to improve the measurement accuracy, i.e., all visible and NIR spectroscopy analysis was based on 400–1000 nm. The above spectral data pre-processing was finished in ViewSpec Pro V2.14 (Analytical Spectral Device, Inc.). Absorbance for the scanned was recorded as $\log [1/R]$, and all spectral records were checked visually and averaged.

Then, all absorbance wavebands were pretreated using standard normal variate transformation (SNV), Savitzky-Golay first derivative transformation (S. Golay 1st-

Der) and wavelet transforms (WT). SNV removes the multiplicative interferences of scatter, particle size, and the change of light distance [7].

Another pre-processing method first derivative is used to remove background and to increase spectral resolution. In this trial, Savitzky Golay [8], which is a moving window averaging method, is used as the smoothing method in first derivative, and the number of smoothing is three.

Wavelet transforms (WT) is a very popular kind of operation today due to its application in chemometrics and signal processing [9][10]. Daubechies-4 is as the wavelet basis function in wavelet analysis. The scale of decomposition is 5, which was selected according to the prediction residual sum of squares (PRESS) in the validation set [11]. The process of wavelet transforms was achieved in the software Matlab7.1. The computer program was edited using two wavelet functions *wavedec* and *appcoef* [12]. The whole calibration process was achieved in The Unscrambler V9.2 (CAMO Process AS.), a statistical software for multivariate calibration.

2.3 Partial Least Squares

PLS is a bilinear modeling method where the original independent information (X-variable) is projected onto a small number of latent variables (LV) to simplify the relationship between X and Y for predicting with the smallest number of LVs [13]. The Y-variable is actively used in assessing the latent variables to ensure that the first one is most relevant for predicting the Y-variable. The Y values can be assigned artificially as 1, 2, 3, 4, 5, 6, 7, 8 and 9, presenting the nine milk powder varieties of Neatle Nestogen, Yili, Sanlu, Guoli, Beinmate, Dumex Dulac, Syntra, Youbo Saint and Abbott Premilac, respectively.

In PLS analysis, the optimal number of PLS components that optimizes the predictive ability of the model was determined by cross-validation. Prediction residual sum of squares (PRESS) for the test samples is used as a function of the number of PLS components retained in the regression model that was formed with the training data [14].

3 Results and Discussion

3.1 Features of NIR Reflectance Spectra

The average absorbance spectra from 400 to 1000 nm were showed in Fig. 1 for randomly selected samples of nine varieties of milk powder: (1) Neatle Nestogen, (2) Yili, (3) Sanlu, (4) Guoli, (5) Beinmate, (6) Dumex Dulac, (7)Syntra, (8) Youbo Saint, (9) Abbott Premilac. It could be seen that the spectral profiles were substantially distinguished from each other, especially at the visible wavelength band. It indicates that it is possible to discriminate the nine milk powder varieties. Qualitative clustering is achievable based on these spectral differences. The differences may be caused by the different internal attributes of milk powder, such as the energy, fat, linoleic acid, α - linoleic acid, et al.. The discrimination of different elements maybe causes the variation of the spectra.

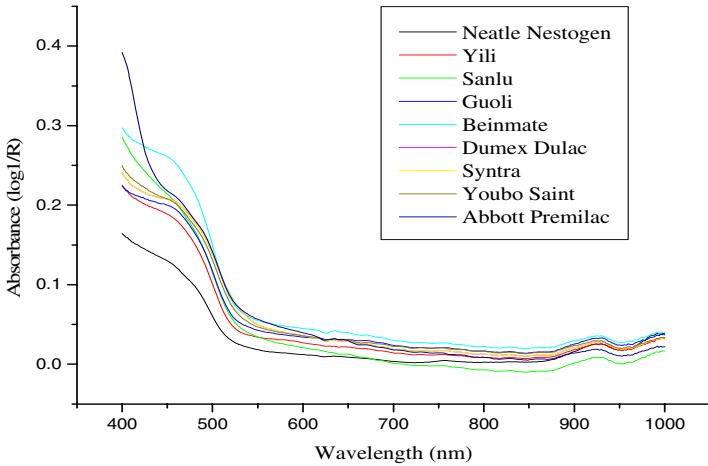


Fig. 1. Average absorbance spectra of wavelength 400-1000 nm for nine infant formula milk powder varieties

Before the spectral analysis, three pre-processing methods were used to eliminate noises. The SNV method reduced the parallel translation of the spectra, and made the peak and valley more obvious. S. Golay 1st-Der caused the great changes in the slopes of the raw spectra and many overlapped peaks could be differentiated. For the spectra with WT comprssion technique, the 601 original variables were compressed into 25 characteristic wavelet components.

For a complex analysis model, especially for the spectral analysis with large data, it is difficult to decide which pretreatment method is the best or optimal. Thus, in this research, the comparison of different pre-processing methods was discussed, and then the better method was selected to bulid the prediction model.

3.2 Comparison of PLS Classification Models Using Different Pre-processing Methods and Wavelength Bands

With different pre-processing methods and wavelength bands, seven PLS calibration models were built. The 225 samples for modeling were split randomly into a calibration and validation set. The quality of the calibration model was quantified by the standard error of calibration (SEC), standard error of cross-validation (SECV) and the coefficient of determination (r_c^2 and r_{cv}^2) [15]. The cross-validation was performed on the calibration samples based on excluding a certain number of samples for the calibration model. For comparison of different sized models of similar constituents, only SEC and SECV were considered, because adding any independent variable to the model would increase r^2 but adding an “unimportant” independent variable can increase either SEC or SECV values [14]. In addition, the root mean square errors of calibration and cross-validation (RMSEC and RMSECV) were used to determine the optimal model without “overfittedness” or “underfittedness”.

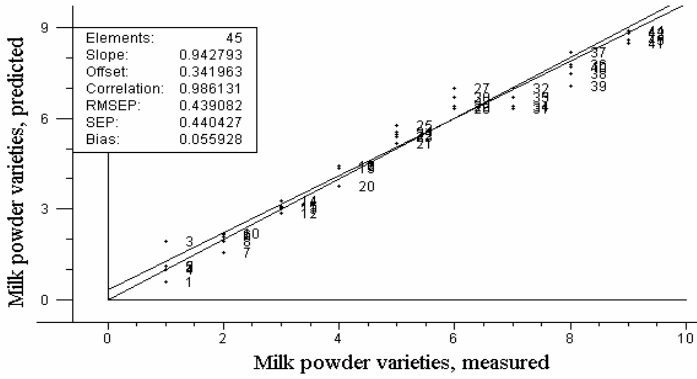
All the parameters of different calibration and validation sets were showed in Table 1. The r_c^2 was above 0.9 in all the models and the r_{cv}^2 also exceeded 0.9 except the model of S. Golay 1st-Der with the wavelength region in 700-1000nm. In the models of SNV (400-700 nm and 400-1000 nm) and WT (400-1000 nm), the SEC, RMSEC, SECV and RMSECV were low enough, and the r^2 of calibration and cross-validation were very high. These three models were considered as optimal models to predict remained samples and evaluate reliability of them further.

Table 1. Parameters of PLS models in different pre-processing methods

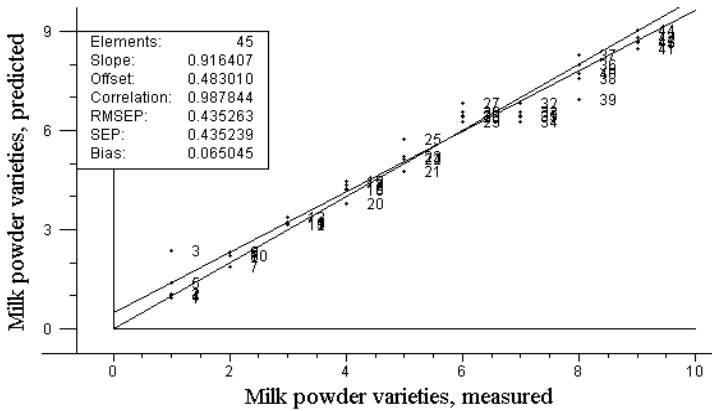
Method	Region (nm)	Calibration			Cross-validation		
		SEC	RMSEC	r_c^2	SECV	RMSECV	r_{cv}^2
SNV	400-700	0.432	0.431	0.972	0.459	0.458	0.969
	700-1000	0.677	0.675	0.932	0.715	0.713	0.924
	400-1000	0.433	0.432	0.972	0.462	0.461	0.968
S. Golay 1 st -Der	400-700	0.576	0.575	0.950	0.648	0.646	0.937
	700-1000	0.644	0.642	0.938	0.834	0.832	0.896
	400-1000	0.463	0.462	0.968	0.557	0.555	0.954
WT	400-1000	0.442	0.441	0.968	0.476	0.475	0.966

3.3 Comparison of Predicted Results of Three PLS Models

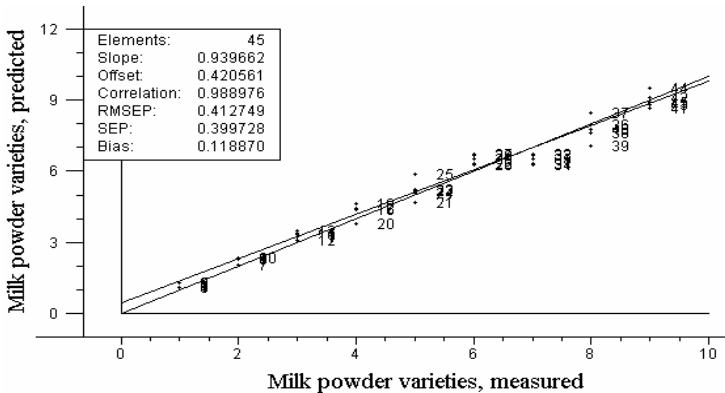
The three PLS models were applied to predict the 45 remaining samples after similar mathematical pretreatment to the calibration ones. The predicted results of nine varieties using three different pre-processing methods were shown in Fig. 2. All of the models developed had a coefficient of determination (r^2) in prediction higher than 0.97 ($r > 0.98$), and the r^2 in PLS with WT model is the highest. The values of standard error of prediction (SEP) and the root mean square error of prediction (RMSEP) were also calculated. From Fig. 2a, Fig. 2b and Fig. 2c, it was found that the same pre-processing method SNV with different wavelength regions had different predicted results. The model with the full wavelength range is better than the model with a certain wavelength range. However, considering the practical application and simplification of the model, the method using less wavelength range is superior if the use of fewer variables obtains the same or better predictive performance. The highest r^2 of 0.978 and lowest SEP of 0.435 and RMSEP of 0.413 in PLS with WT model showed that wavelet transforms was an optimal method to pre-process data of the spectra, and it is desirable to solve the problem of the variety discrimination. Thus, it is very important to select a proper pre-processing method when building the model.



(a) PLS with SNV on 400-700nm ranges predicted results for nine milk powder varieties



(b) PLS with SNV on 400-1000nm ranges predicted results for nine milk powder varieties



(c) PLS with WT on 400-1000nm ranges predicted results for nine milk powder varieties

Fig. 2. Predicted results of three different PLS models for nine milk powder varieties

4 Conclusions

Nine different varieties of infant milk powder showed an excellent difference in their spectra. It demonstrates that it is possible to develop a non-destructive technique for discrimination of milk powder varieties by visible and near infrared spectra technique. The absorbance spectra of milk powder were pre-processed by three methods including SNV, S. Golay 1st-Der and WT. A PLS analysis was applied to build the classification model. The calibration, validation and prediction statistics showed that PLS model was an available alternative for pattern recognition. The best three models including PLS with SNV (400-700nm), SNV (400-1000nm), and WT (400-1000nm) were used to predict the remained samples. The predicted results of PLS analysis showed that these three models can qualify for discrimination of milk powder varieties. Besides, the highest r^2 of 0.978 and lowest SEP of 0.435 and RMSEP of 0.413 in PLS with WT model showed that wavelet transforms was an optimal method to pre-process data of the spectra. For a complex analysis system, it is necessary to select and compare different pre-processing methods when building models.

Acknowledgements

This study was supported by the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of MOE, P. R. C., Natural Science Foundation of China (Project No: 30270773), Specialized Research Fund for the Doctoral Program of Higher Education (Project No: 20040335034), and Science and Technology Department of Zhejiang Province (Project No. 2005C21094, 2005C12029).

References

1. He, Y., Li, X.L., Shao, Y.N.: Quantitative Analysis of the Varieties of Apple Using Near Infrared Spectroscopy by Principal Component Analysis and BP Model. *Lecture Notes in Artificial Intelligence*, Vol. 3809 (2005) 1053-1056
2. He, Y., Feng, S.J., Deng, X.F., Li, X.L.: Study on Lossless Discrimination of Varieties of Yogurt Using the Visible/NIR-spectroscopy. *Food Research International*, Vol. 39. No. 6 (2006)
3. Laporte, M.F., Paquin, P.: Near-infrared Analysis of Fat, Protein and Casein in Cow's Milk. *Journal of Agricultural and Food Chemistry*, Vol. 47. No. 7 (1999) 2600-2605
4. Sorensen, L.K., Jepsen, R.: Assessment of Sensory Properties of Cheese by Near-infrared Spectroscopy. *Int. Dairy Journal*, Vol. 8. No. 10-11 (1998) 863-871
5. Hermida, M., Gonzalez, J.M., Sanchez, M., Rodriguez-Otero, J.L.: Moisture, Solids-non-fat and Fat Analysis in Butter by Near Infrared Spectroscopy. *International Dairy Journal*, Vol. 11. No. 1-2 (2001) 93-98
6. Candolfi, A., Maesschalck, R.D., Jouan-Rimbaud, D., Hailey, P.A., Massart, D.L.: The Influence of Data Pre-processing in the Pattern Recognition of Excipients Near-infrared Spectra. *Journal of Pharmaceutical and Biomedical Analysis*, Vol. 21.No. 1 (1999) 115-132

7. Chu, X.L., Yuan, H.F., Lu, W.Z.: Progress and Application of Spectral Data Pretreatment and Wavelength Selection Methods in NIR Analytical Technique. *Progress in Chemistry*, Vol. 16. No. 4 (2004) 528-542
8. Liang, Y.C., Yi, Z.S.: *The Handbook of Analytical Chemistry: Chemistry Metrology*. Chemical Industry Press, Beijing (2001)
9. Staszewski, W.J.: Wavelet Based Compression and Feature Selection for Vibration Analysis. *Journal of Sound and Vibration*, Vol. 211. No. 5 (1998) 735-760
11. Drai, R., Khelil, M., Benchaala, A.: Time Frequency and Wavelet Transform Applied to Selected Problems in Ultrasonics NDE. *NDT&E International*, Vol. 35. No. 8 (2002) 567-572
12. Wang, F., Chen, D., Shao, X.G.: Application of Wavelet Transform and Partial Least Square in Prediction of Common Chemical Compositions in Tobacco Samples. *Tobacco Science & Technology/ Inspection & standard*, No. 3 (2004) 31-34
13. FECIT Sci-Tech.: *The Theory of Wavelet Analysis and MATLAB 7 Application*. Publishing House of Electronics Industry, Beijing (2005)
14. Xing, J., Linden, V.V., Vanzebroeck, M., Baerdemaeker, J.D.: Bruise Detection on Jonagold Apples by Visible and Near-infrared Spectroscopy. *Food Control*, Vol. 16. No. 4 (2004) 357-361
15. Dagneu, M., Crowe, T.G., Schoenau, J.J.: Sensing of Hog Manure Nutrients with Reflectance Spectroscopy. *CSAE/SCGR-NABEC Meeting*, July 8-11, Guelph, Canada (2001)
16. Naes, T., Isaksson, T., Fearn, T., Davies, A.M.: *A User-friendly Guide to Multivariate Calibration and Classification*. NIR Publications, UK (2002)

Semi-automatic Hot Event Detection*

Tingting He^{1,2}, Guozhong Qu², Siwei Li³, Xinhui Tu², Yong Zhang², and Han Ren²

¹ Software College of Tsinghua University 102201 Beijing, China

² Department of Computer Science, Huazhong Normal University 430079 Wuhan, China

³ School of Mathematics and Statistics, Wuhan University, 430072 Wuhan, China
tthe@mail.ccnu.edu.cn, qu_g_z@mails.ccnu.edu.cn,
lisiwei2000@hotmail.com, tuxinhui@mails.ccnu.edu.cn,
ychang@mails.ccnu.edu.cn, renh@163.net

Abstract. In this paper, we propose a method to detect hot event automatically. We use all the web pages from Jan 1st 2005 to Dec 31st 2005, and detect new events by using incremental TF-IDF model and incremental cluster algorithm. Based on analysis of the attributes of events, we propose a method to measure the activity of events, then filter and sort the event according to the activity of events; finally a hot event list can be derived.

1 Introduction

Hot Events imply the focus people most concern during a period of social revolution and development. For example, “超级女声” (“*chao ji nv sheng*”, A Chinese TV show like *American Idol*) is consider to be an hot event in 2005 China. Correlative mediums promulgate hot event list every year in different domains, such as economic hot events, entertainment hot events, etc. But most of the events are chosen out manually. Individual factor normally takes a decisive effect in judging whether an event is a hot event and also these work costs a lot. It is significant to develop scientific and effective methods to detect hot events automatically.

Contrasting to detecting hot event by manual method, our method has advantages of the following aspects: 1) It can reduce the reliance on manual intervention. Each event is treated equally by the computer, the effect of the human emotion is weaken to a certain extent during the process of judging whether an event is a hot event. 2) It also is domain independent. It can be used to detect economic hot events, entertainment hot events and also hot events in other domains. 3) It has no time limitation to investigate hot events. It can be done whenever users want to investigate new events and hot events. 4) It also can be used in business intelligence extraction.

In the following section, previous work related to event detection and extraction of the attributes of hot event is discussed in more details. Our system design for new event detection is presented in Section 3, followed by a description of hot event detection in Section 4. In section 5 we show the results obtained by our detection systems and the evaluation methodology used. Finally we give a conclusion in Section 6.

* The paper is supported by National Natural Science Foundation of China (NSFC), (Grant No.60496323; Grant No.60375016 Grant No.10071028;); Ministry of education of China, Research Project for Science and technology, (Grant No. 105117).

2 Related Work

The first step to detect hot events is to detect all new events occurs in news reports. New event detection is typically approached by reducing a news report to a set of features, either as a vector [1] or a probability distribution [2]. When a news report arrives, its feature set is compared with those of all past news reports. If there is a sufficient difference, the news report is marked as a new event, otherwise, not. There is evidence that this simple approach to new event detection was not very effective [3]. Another approach to do new event detection is using clustering algorithm. The cluster detection system's job is not just to identify the new topic in the news, but also to cluster new reports on the same topic into bins. When a new bin is needed, i.e., when first new report arrives, it must be created. The creation of bins is an unsupervised task: the system has no knowledge of the number of expected bins [4,5,6]. In our work, we integrate these two methods.

To detecting hot events automatically is a difficult but interesting task. By far, we have not found papers about automatic hot event detection. There is some research done in the field of buzz detection. The research of hot event detection we describe in this paper use the methodology of buzz detection as reference [7].

In the research of buzz word detection, the researchers extract the valid words and phrases, analyze the attributes of every word and make the definition of activity, based on which they use methods to measure the degree of concern for words and phrases, and also according to the tendency curves, sort the words by computer, and finally got the candidate popular words and phrases.

3 New Event Detection

The first step of hot event detection is to detect the event in the news report. We use incremental TF-IDF model and incremental cluster algorithm to detect new event. In this section we will describe the processing steps in new event detection and the models we used in our system specifically.

3.1 New Event Detection Processing

At first, we download about 290,000 news web pages from news web sites. The web pages are converted into text files in the same Chinese code by extracting the content of the web pages, and then we segment the texts and remove the stop words from segmented texts.

According to the subject, all the web pages are divided into 7 groups (international, society, finance, science technology, sports and entertainment) denoted by G_i ($0 < i < 8$). After that, we process G_i by following steps.

Step 1. Divide G_i is into 12 corpuses denoted by C_{ij} ($0 < j < 13$) according to the time of publication, i.e., one corpus per month.

Step 2. For each C_{ij} , we use the method described in 3.2 to build term vectors in document space and use the clustering algorithm described in 3.3 to get a temp event list.

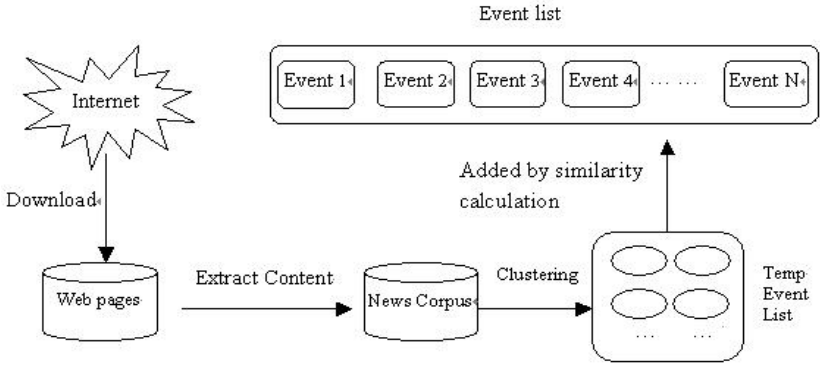


Fig. 1. Outline of new event detection

Step 3. Integrate the events in the temp event list to the final event list by similarity calculation using the formula discussed in 3.4.

Repeat step 2 and step 3 until all corpus of the same subject is done, then we get an event list belong to the subject.

3.2 Building Term Vectors in Document Space

This section describes how to represent the documents in vector space. Our similarity calculations of documents and cluster algorithm are based on an incremental TF-IDF model. In a TF-IDF model, document frequencies are static. In the incremental model, document frequencies $df(w)$ are not static but change in time step t . At time t , a new set of document C_t is added to the model by updating the frequencies [8].

$$df_t(w) = df_{t-1}(w) + df_{C_t}(w) \tag{1}$$

Where $df_{C_t}(w)$ denote the document frequencies in the newly added set of documents.

The document frequencies as described above are used to calculate weights for the terms w in the documents d . At time t , we use the following formula.

$$weight_t(d, w) = \frac{1}{Z_t(d)} f(d, w) \cdot \log \frac{N_t}{df_t(w)} \tag{2}$$

Where N_t is the total number of documents at time t . $Z_t(d)$ is a normalization value such that either the weights sum to 1 (if we use Hellinger distance, KL-divergence, or Clarity-based distance), or their squares sum to 1 (if we use cosine distance)[9,10].

3.3 Clustering Algorithm

Having represented news reports and clusters in vectors, the incremental clustering is straightforward. For each news report, compute the cosine similarity of this news report

and each cluster centroid in the accumulated set. If the similarity score between this news report and the closest cluster is above a threshold (pre-selected), then add this news report to the cluster as a member, and update the prototype vector correspondingly using the method discussed in 3.2. Otherwise, add this story as a new cluster in the set. Repeat the above until the corpus is done.

3.4 Similarity Calculation

The vectors consisting of normalized term weights $weight_t$ are used to calculate the similarity between two events d and q . In this paper, we use the Hellinger distance.

$$sim_t(d, q) = \sum_w \sqrt{weight_t(d, w) \cdot weight_t(q, w)} \quad (3)$$

If the similarity score between this event and the closest event is above a threshold (pre-selected), then add this event to the closest event as a member, and update the vector correspondingly using the method discussed in 3.2. Otherwise, add this event as a new event into the event list.

4 Hot Event Detection

In this section, we propose a method to detect hot events from events list we get in previous section. Firstly, the features of hot event are given, which are based on our previous study on buzz words. Secondly, we discuss the quantification of these features. Finally, we discuss our scoring algorithm.

4.1 Attributes Extraction

Maybe, hot events have many features. Here we just take the following two points into consideration.

1. Attracting explicitly more attention during the research period, which is called “ascending-stage”.
2. Stepping in a steady development after “ascending-stage”, which is called “steady-stage”.

For example: in fig.2, the period from A to B is “ascending-stage”. The period from B to C is “steady-stage”.

Features above are estimated by the following two attributes:

1. ef . ef is the frequency of event reported in a time unit, i.e. a period of time, such as a month. The Higher ef indicates the event attracted more attention during a time unit.

2. cu . cu is the count of consecutive effective time units. If the ef of an event in a time unit reaches some threshold, it means the event is effective in the time unit. The higher cu an event has, the longer the event will last.

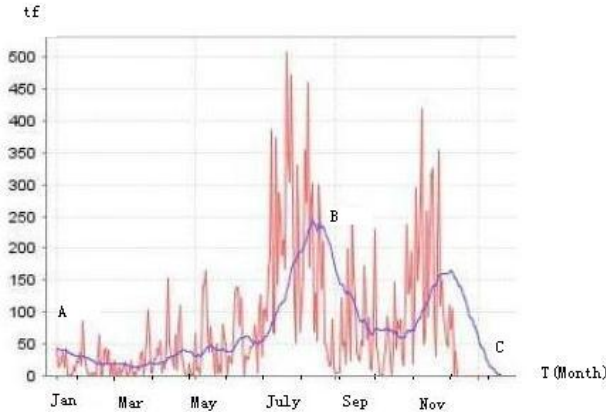


Fig. 2. The activity curve of “ the issue of textile trade” *tf* is the frequency of event reported in a time unit. The toothshaped line denotes the raw data of “ the issue of textile trade”. The other line denotes the smoothed result.

4.2 Ranking

To rank the events, we consider the following two factors to score the activity of the events:

- 1) Relative distribution: the ratio of document frequency of an event in a time unit against the number of all documents in that time unit.
Intuitively, the more frequently an event reported in the time unit, the more important the event tends to be.
- 2) The continuum of event □ The max value of *cu* .
Intuitively, the more consecutively of an event reported, the more important the event tends to be.

With both of the two factors taken into consideration, the score assigned to an event *E* is given by the following formula, which reflects the activity of an event.

$$\sum_{i=1}^n \frac{ef_i}{S_i} \times \sqrt{1 + \frac{\max(cu)}{n}} \tag{4}$$

Where ef_i is the document frequency of an event in time unit *i*, *n* is the sum of time units. S_i is the sum of all documents in time unit *i*. $\sqrt{1 + \frac{\max(cu)}{n}}$ gets 1 when the event does not get consecutive effective time units.

5 Our Works and Results

Most of our works are done by computers automatically. We use all the web pages from Jan 1st 2005 to Dec 31st 2005, and detect new events by using incremental TF-IDF model and incremental cluster algorithm. Based on analysis of the attributes of events, we propose a method to measure the activity of events, then filter and sort the event according to the activity of events, finally got the hot event list.

5.1 Corpus

We download news from the period January to December 2005. It contains approx 290,000 pieces of news. According to the subject, the web pages are divided into 7 groups. Each group is divided into 12 small corpuses, i.e., one corpus per month. Table 1 shows the details of the groups.

Table 1. Details of the groups

Subject	Count
Finance	73,076
International	71,848
Society	16,858
Entertainment	11,142
Sports	103,632
Science and Technology	13,151
Sum	289707

5.2 Processing and Result

The detection procedure is described as following steps:

-
1. Download web pages.
 2. Extract text contents of the web pages to form raw news web pages
 3. Segment the texts and remove stop words.
 4. Detect new events using the method explained in Section 3.
 5. Filter the events as follow:
 - a) Drop the event whose document number of event set is less than 100.
 - b) Drop the event who's df is less than 2.
 6. Score and Rank the events according to the activity of the events.
 7. Re-rank the top 50 events by manual intervention.
-

Table 2 shows the results after step 4 and step 5. After step 5, the number of events is reduced significantly.

Table 3 and Table 5, respectively, show the top 10 international hot events and the top 10 financial hot events found in our work. And we compare our result with the top 10 hot event provided by other organization. Table 4 shows the top 10 international hot events provided by the State Council Information Office of China (<http://news.xinhuanet.com/world/2006-01/05/content4010814.htm>). Table 6 shows the top 10 financial hot events provided by Shanghai Securities News (<http://www.022net.com/2005/12-28/445764383330658.html>).

Table 2. The result after step 4 and step 5

Type	After step 4 (count of event)	After step 5 (count of event)
Finance	7303	762
International	7628	760
Society	4300	719
Entertainment	2125	762
Sports	3019	710
Science and Technology	3504	730
Sum	27879	4443

Table 3. International hot events promulgated by using our method

No	International	No correspond to table 4
1	Hurricane Katrina	6
2	Bomb explode in London	4
3	Paris Rebellion	10
4	South Asia Quake	1
5	EU Constitution	3
6	60th anniversary of Anti-Fascist War	
7	United Nations reformation	9
8	Huygens lands on Titan	
9	The situation in Iraq	
10	Crude price hits record high	

Table 4. International hot events provided by the State Council Information Office of China

No	International	No correspond to table 3
1	South Asia Quake	4
2	Global Avian Flu	
3	EU Constitution	5
4	Bomb explode in London	2
5	North Korean Nuclear Challenge	
6	Hurricane Katrina	1
7	President Hu calls for a harmonious world on UN conference.	
8	Trial of Saddam Hussein	
9	United Nations reformation	7
10	Paris Rebellion	3

Table 5. Financial hot events promulgated by using our method

No	Finance	No correspond to table 6
1	Renminbi's rise	1
2	China Legislature Abolishes Agricultural Tax	7
3	The issue of textile trade	8
4	Fortune Global Forum	
5	House Price Growth Rate Slows Down in China	5
6	Reformation of Individual income tax categories	
7	Reformation of retirement pension categories	
8	Direct sale law is promulgated	
9	Economic Survey of China 2005	
10	The changes of Housing Loan policy	

Table 6. Financial hot events provided by Shanghai Securities News

No	Finance	No correspond to table 5
1	Renminbi's rise	1
2	China's economy grows 9.9% to US\$2.3 trillion	
3	Advice of 11th Five-Year Plan is carried out	
4	CNPC's purchase of PetroKazakhstan	
5	House Price Growth Rate Slows Down in China	5
6	Construction Bank of China comes into the market	
7	China Legislature Abolishes Agricultural Tax	2
8	The issue of textile trade	3
9	Shareholding allocation reformation	
10	China tries out Circle Economy	

Table 7. Result Comparison

Our result	Top 10	Top 15	Top 20	Top 40
International	6	8	10	
Finance	4	6	6	10

In Table 7, the column “top 10” show the comparison between our “top 10” result and the results provided by other organizations. In international hot events, our international top 10 hot events contain 6 events of the International top 10 hot events provided by the State Council Information Office of China. In financial hot events, there are 4 events contained in our result. In international news, if we take our top 20

events to compare with the data in table 4, all events occur in table 4 are contained in our result. In financial news, if we take top 40 events to compare with the data in table 6, all events occur in table 4 are contained in our result.

Our results show that it is feasible to use our method to detect hot events. The ranking of the event lists got by using statistical information show the most important events of the year objectively.

6 Conclusions

In this paper, we propose a new method to detect hot events automatically. In such case we can reduce the reliance on manual intervention. Every event is treated equally by the system. The effect of the human emotion is weakened to a certain extent during the process of judging whether an event is a hot or not. Also our method is domain independent. It can be used to detect economic hot events, entertainment hot events and also hot events in other domains. In future work, we plan to improve precision of our methods and to weaken the impact of the threshold value.

References

1. Allan, J., Jin, H., Rajman, M., Wayne, C., Gildea, D., Lavrenko, V., Hoberman, R., and Caputo, D. (1999). Topic-based novelty detection. 1999 summer workshop at CLSP, final report. Available at <http://www.clsp.jhu.edu/ws99/tdt>.
2. Jin, H., Schwartz, R., and Walls, F. (1999). Topic tracking for radio, TV broadcast, and newswire. In Proceedings of the DARPA Broadcast News Workshop, pages 199-204. Morgan Kauffman Publishers.
3. Allan, J., Lavrenko, V., and Jin, H. (2000). First story detection in TDT is hard. In Ninth International Conference on information Knowledge Management (CIKM'2000), Washington, D.C.ACM.
4. Y. Yang, T. Pierce, and J. Carbonell. A study on retrospective and on-line event detection. In Proceedings of SIGIR-98, Melbourne, Australia, 1998.
5. Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, Edmonton, Canada, 2002.
6. Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In Proceedings of SIGIR-02, pages 81-88, Tampere, Finland, 2002.337
7. He Tingting, Zhu Yi, Zhang Yong and Ren Han. Study On Computer-aided Extracting Popular Words and phrases Based on Determinant attributes. In Proceedings of HNC& linguistics Research Workshop 2005.
8. Ayman Farahat, Francine Chen, Thorsten Brants, Optimizing Story Link Detection is not Equivalent to Optimizing New Event Detection. In Proceedings of ACL-2003
9. W. B. Croft, S. Cronen- Townsend, and V. Larvrenko.Relevance feedback and personalization: A language modeling perspective. In DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001.
10. V. Lavrenko, J. Allan, E. DeGuzman, D. LaFlamme, V. Pollard, and S. Thomas. Relevance models for topic detection and tracking. In Proceedings of HLT-2002, San Diego, CA, 2002.

Profile-Based Security Against Malicious Mobile Agents

Hua Li¹, Glenna Greene¹, and Rafael Alonso²

¹ Sarnoff Corporation, 201 Washington Road, Princeton, NJ 08543, USA
{hli, egreene}@sarnoff.com

² SET Corporation, Arlington, VA 22203, USA
ralonso@setcorporation.com

Abstract. In this paper we describe a security system against malicious agents in a wireless ad hoc network. This system monitors the activities of mobile agents. By mining the log data, the system builds dynamic behavioral profiles for each class of agents. The profiles are then used to classify each agent instance and detect anomalous agents. The malicious nature of identified anomalous agents are further determined by a threat assessment module. For mobile agent profiling, we presented two approaches: one based on rule-learning and another on histograms of features related each agent class. We have implemented a prototype system using the histogram-based approach. The prototype is also described in the paper.

1 Introduction

In this paper we apply a modeling approach to software agents in a wireless, mobile network. Our mobile agent network employs the Extendable Mobile Agent Architecture (EMAA) [2]. Agents are carriers of mobile code with the ability to “migrate” from host to host, collect and provide data, monitor status, and use computing resources on visited hosts [6]. As a result, mobile agents enable systems to dynamically relocate computation tasks from nodes with fewer resources to nodes with greater resources. In a mobile agent network comprised of devices with varying capabilities (for example, a network of PDAs and laptops) the use of agents allows the workload to be distributed more efficiently.

The security risk can be high in a mobile agent network. There are at least three types of vulnerabilities. First, the mobile agents can be malicious. Second, the host can be compromised. And third, a user of the network can be malicious. The focus of this paper is to address the first type of vulnerability: i.e., malicious mobile agents.

Until very recently, few studies have approached the security of wireless networks and mobile agent systems from the perspective of intrusion detection. One study has utilized only two features of mobile agents for agent profiling: the different agent’s movement patterns and the variety of agent’s host residence time [6]. This study is preliminary in the sense that the number of features is too small, the profiles are not dynamically adjusted, and no attempts are made to reduce the false alarm rate.

Researchers have used certificates and security policies to prevent malicious behavior at runtime [4, 8]. Signed code is a fundamental technique devised for protecting the agent platform. An agent is assigned a digital signature that confirms its authenticity, its origin, and its integrity. Typically the code signer is either the creator of

the agent, the user of the agent, or some entity that has reviewed the agent. A security policy specifies permission and privileges for accessing services and resources in a platform. Each visiting agent must subject to the platform's security policy. The platform must first authenticate a mobile agent's identity before it is instantiated.

Our approach to agent security is to profile a class of agents based on behavioral features and then classify every instance. We then identify agents that do not fit into a known class and those that deviate from their class profile (because they no longer fit the known class). One important feature of our system is that we analyze the security risk to the detected anomalies using a threat assessment module to reduce false alarms. We have investigated two effective profiling methods: one rule-based and another histogram-based.

2 System Architecture

The system architecture is shown in , 2006.

© Springer-Verlag Berlin Heidelberg 2006. Network and resource features are monitored and stored. For each agent class, a profile is generated to model typical behavior. The anomaly detection module flags agents with behavior inconsistent with profile. Alerts are triggered when flagged agents whose behavior is deemed as threatening by the threat analysis module.

2.1 Feature Data Monitoring

We monitor a comprehensive set of features that characterize three key aspects of a mobile agents' behavior in order to catch those feature that are truly discriminating. The three aspects are Life Cycle Activities, Service and Resource Usage, and Network Activity. Life Cycle Activity features related to the unique life events of an agent. Service and Resource Usage features capture characteristics of the work an agent is performing. Network Activity features represent the mobility behavior of an agent. As a result, the monitored features are grouped three corresponding categories. Each category is described below in more detail.

2.2 Agent Class Profiling and Anomaly Detection

We have investigated two approaches for agent profiling: the rule-based and the histogram-based approaches. The former applies available rule induction software to produce classification rules for agent classes. These rule sets are used to classify the agents and to detect anomalies. We choose the rule-based method over many other classification techniques because it has a number of desirable properties. In addition, rule-learning systems outperform decision tree learners on many problems. Rule sets

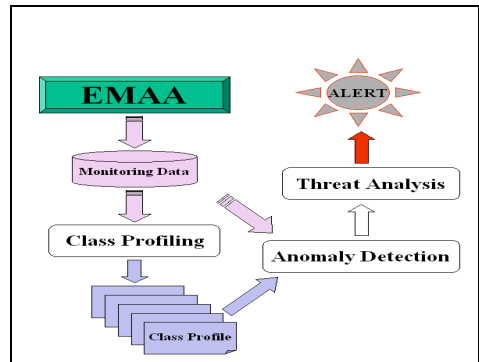


Fig. 1. Agent security architecture

are relatively easy for people to understand. As a result, this method allows us to identify discriminating features by studying the generated rules in experiments.

One shortcoming of the rule-based approach is that the learning is performed off-line in a non-incremental fashion. This problem limits its use in dynamic learning applications with streaming data. To overcome this problem, we developed the histogram-based approach that uses the histograms of feature values within an agent class to estimate the statistics of the underlying random process. The histogram models are used to distinguish self (agent that belongs to this class) and non-self (agent that does not belong to this class). These models can be built dynamically and incrementally. This is the method was implemented for our prototype system. Forrest et al. also proposed self/non-self discrimination in the context of intrusion detection [3]. In their work, normal behavior was defined in terms of short sequences of system calls of a certain length in a running Unix process, and a separate database of normal behavior was built for each process of interest.

The profiles we built for a class of agent with the above approaches represent the normal behavior of the agents. They are used to detect anomalies. Essentially, an agent whose behavior significantly deviates from the profile for the agent class will be labeled as an anomaly. The details of these two learning approaches and their anomaly detection methods are presented in section 3 below. Results of experimental evaluation of the rule-based approach are provided in section 4.

2.3 Threat Assessment

The purpose of threat assessment is to reduce the rate of false positives, which is usually high in anomaly detection systems. Anomalies may or may not be harmful or malicious to the system. Only those that are harmful should trigger an alert. Threat assessment is a tool to determine the threat level (TL) of an anomalous event to the system. We conceptualize threat level of an event as a function of several variables:

- The potential HARM implied by the event. This is analyzed in two aspects: a) the critical system resources accessed by the event; b) the involvement in the system operations by the event
- The RARITY of the event
- WHO is engaged in the event
- The TIME at which the event took place
- The TL of a composite event is a non-linear function of the TL of the component events, e.g., a combination of low and high TL events is very likely to have a high TL

In our prototype security system, each anomaly is assessed to determine the level of threat implied: Threat is based on resource usage (type and quantity), data access, etc. An anomalous activity that is also threatening triggers an alert.

3 Agent Class Profiling

In this section, we describe two approaches for agent class profiling and their respective anomaly detection methods. To describe the behavior of each running

agent, we maintain a vector of values, each corresponding to one of the monitored features listed above. For off-line analysis of log files, a complete record of each agent is available. However, for on-line profiling, the record is built up over time with approximations of each feature continually refined over time. These two cases can be treated separately and the off-line version can be used to evaluate the on-line version.

3.1 Rule-Based Approach

In this approach, the profile for each class of agents is represented as a set of propositional rules. One example rule (output from RIPPER, see below) is given below.

```
[covers 6 pos, 0 neg examples]
swat.walkietalkie.AudioDeliveryAgent:-
DURATION_START_SERVER_CALL<=239,    BIRTH_HOST=ipaq25.
```

The rules are induced from a training dataset, which consists of examples of each class of agents. Each example consists of a vector of the values for the selected features and a class label. The rule sets are evaluated by examining the precision and recall metrics when they are applied to predict the class of unseen examples.

3.1.1 Profiling: Learning Rule Sets for Agent Classes

The method we use for inducing rules for agent classes is the RIPPER rule induction algorithm [1]. RIPPER can work with set-valued features and generate concise rule sets. It is very stable and has shown to be consistently one of the best algorithms in its class. Lee et al. applied RIPPER to intrusion detection in Unix system calls [5].

The training data for our work come from the log files that record the agent activity. We assume that there is a period of normalcy when the system is not under attack. The logs from this period are processed to produce training data. A sub-sample of the data is used as test data to evaluate the rule sets learned.

3.1.2 Anomaly Detection

For anomaly detection, the rule sets are learned using training data reflecting normal agent activity. They are then used for detection of abnormal agents. If an agent claims to be a member of an agent class, its activity is expected to be consistent with that class. Application of the learned rules to the current agent activity provides a measure of this consistency. The agent is judged to be anomalous if its activity profile does not fit the rule set for the claimed class.

3.2 Histogram-Based Approach

The profile for a particular class is composed of the means and variances of each of the features. During evaluation of an individual agent (anomaly detection), a cost function is calculated. This cost function combines normalized deviations from the mean in each of the feature categories (normalized by variance). The entire cost function is then weighted by a certainty factor that is based on the number of exemplars used to generate the profile.

3.2.1 Profiling

The profiling feature set described above is represented as two array's of real numbers, $mean[]$ and $var[]$. The profile for a given class includes these two arrays as well as a counter, $count$, of the number of exemplars used in its creation.

At step $j+1$, when a new exemplar becomes available, the profile is updated as follows for each parameter, i :

$mean_{j+1}[i] = (count_j * mean_j[i] + exemplar[i]) / (count_j + 1)$
$var_{j+1}[i] = ((count_j * var_j[i]) + count_j * mean_j[i] + exemplar^2[i] - (count_j + 1) * mean_{j+1}[i]) / (count_j + 1)$
$count_{j+1} = \min(max_count, count_j + 1)$

These equations simply provide a sequential method for calculating the mean and variance of the i^{th} parameter of the exemplars.

3.2.2 Anomaly Detection

We assume that each agent running in the SWAT must declare its agent class. We further assume that each individual instance of an agent is assigned a unique identifier. While this assumption may be true, it exposes a potential attack point for an adversary. Agents must be prevented from changing or deleting their assigned unique identifier.

Agent activity will be monitored and logged, indexed by the unique identifier. After each log entry, the anomaly detector will update the feature vector, $exemplar$, and compare each entry to the corresponding agent class profile. This comparison results in an Anomaly Factor, AF , defined as follows:

$$AF = \frac{count}{max_count} \sum_i \frac{\alpha |exemplar[i] - mean[i]|}{var[i]}$$

Each time it is updated, the Anomaly Factor is compared against a threshold. Any agent for which the Anomaly Factor exceeds the threshold is considered anomalous and flagged as such.

4 Experimental Results for Rule-Based Profiling Approach

In this section, we evaluate the performance (precision and recall) of rule learning method in classifying agents using experimental data. To this goal, we applied RIPPER to a data set generated by mobile agent applications (i.e. Walkie-Talkie and Collaborative Whiteboard) created and tested by colleagues at Drexel University [7].

There are at least 8 agent classes involved in the experiment and we considered 6 of them in our analysis using RIPPER. The six agent classes are: Audio Delivery Agent (which sends voice data in the Walkie-Talkie application), Line Delivery Agent (which sends drawing data in the Collaborative Whiteboard application), Ping Agent, Do Nothing Agent, Group Display Agent, and Group Display Vote Agent.

4.1 Data Preparation

The agent features are extracted using a Java program from the log files on a per agent basis. We identify an agent by its unique identifier. The features that are used for rule learning include BIRTH_HOST, ARRIVE_HOST (host this agent migrated from), START_STATUS (whether this agent foreign or endogenous), EXEC_DURATION, NUM_START (number of executions by this agent in its lifetime), NUM_SERVER_CALL, DURATION_ARRIVE_START (duration from arrival to start of 1st execution), DURATION_START_SERVER_CALL (duration from start of 1st execution to first server call). Other derived features include averages and standard deviations of EXEC_DURATION, START_INTERVAL, and SERVER_CALL_INTERVAL.

4.2 Learning Approaches

In a multi-class approach, all 6 types of agents i.e., classes are considered. The system learns rule sets for each class. The 6 class labels are specified in the names file explained above.

In a single-class approach, tuples of classes other than the class at questions are combined and relabeled as the default class. The system learns a rule set for the class in question. We also refer to this approach as “self-non-self” approach since tuples in the class in question form “self” whereas tuples of all other classes form “non-self”. Only two class labels are specified in the names file above: the class in question and the default class.

4.3 Result Summary

The multi-class approach is comparable to the single-class approach. The former is more efficient in producing rule sets because it takes one run to generate rule sets for all involved classes (**Table 1**).

Within the multi-class approach, the setting “Learning from Single Log File, Test on Sub-sample of the Same Log File,”

Table 1. Performance of Single- (in bold) vs. Multi-class (in parentheses) approaches

Setup	Classes	% Recall	% Precision	tuples
<i>Learning from Single Log File, Test on Sub-sample of the Same Log File</i>	AudioDeliveryAgent	57 (57)	50 (80)	7
	LineDeliveryAgent	93 (96)	90 (90)	28
<i>Learning from Multiple Log Files, Test on Different Log File</i>	AudioDeliveryAgent	69 (64)	59 (61)	71
	LineDeliveryAgent	94 (89)	89 (92)	331

Test on Sub-sample of the Same Log File” gives acceptable performance. The rule sets generated this way have a host-dependency and do not transfer well to new hosts. On the other hand, the setting “Learning from Multiple Log Files, Test on Different Log File” avoids the host-dependency but the performance is acceptable with some exceptions. Also, for this setting to work the learner needs access to logs on multiple if not all hosts. Since the training is based on data from multiple hosts, the overhead for data transfer across the network and the time required for training will be much higher than the setting involving only the local host.

5 A Prototype Security System

Our prototype mobile agent network consists of one IBM laptop and two handheld devices (iPAQ 3800) (Fig. 2). The physical network infrastructure is based on 802.11b wireless LAN technology, using Cisco network interface cards. The software infrastructure is mostly OpenSource: Linux Familiar OS 0.5.3 (Kernel 2.4.18), Blackdown Java 1.3.1, OpenSSL, and EMAA. The software is uniform across nodes.

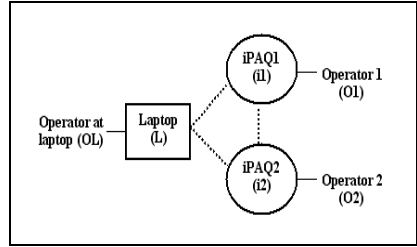


Fig. 2. The demonstration setup

The agent monitoring software is installed on all three nodes. In order to collect training data, the monitoring software is configured to save its output to log files. An off-line classifier to establish classification boundaries delineating normal behavior for agent class uses this data. At run-time, the monitoring software is configured to compare observed behavior to stored boundaries to classify behavior as “normal” or “anomalous”. Normal and anomalous agents can be launched from any of the three nodes.

Each of the three nodes also has a “user emulator” which can automatically launch mobile agents into the network. Finally, the laptop node has additional software that can be used to display alert details (Fig. 3).

We have created three classes of mobile agents: text delivery agent, line delivery agent, and nearest neighbor agent. The text delivery agent carries an ASCII string to each of the hosts on its itinerary. The line delivery agent carries a vector to each of the hosts on its itinerary. The nearest neighbor agent travels to each of the hosts on its itinerary and, at each host, gathers some information and performs some iterative calculation before moving on to the next host. The demonstration consists of 4 acts run sequentially: 1) presentation of agents; 2) presentation of malicious agents; 3) presentation of user emulation; and 4) detection of malicious agents.

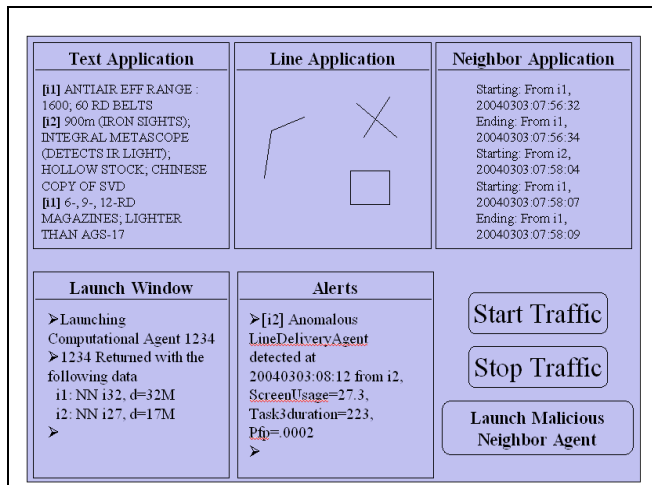


Fig. 3. The Laptop demo application user interface

6 Future Work

There are three directions we are planning to extend our work in the future. The first is to generalize our system to work with **new agent classes**. This refers to the phenomenon that a host in the mobile agent network does not have a class library and thus no class profile for the arriving agent. Our approach is as follows. The receiving host can request the class library for this agent from the source host of this agent. In addition, this host can request class profile for this agent in a P2P sense and then selects most similar source (based on version, most recent update, reliability of profile, host description features, distance in hops, etc.). If no profile is available, agents can be allowed to run under the condition that all its activity is examined by threat assessment module. The data of this agent is collected and used to build profile on-line.

The second direction is to provide **proactive network protection**. Decisions for handling reported alerts could be made while agent is running. Threatening activity can trigger agent termination.

Acknowledgments

The authors would like to thank Dr. Jeffrey Bloom for his active participation in the research and development of this work. We are also indebted to Drs. William Regli and Moshe Kam and their SWAT team for their invaluable advice and collecting agent log data as well as their help in setting up our mobile agent network. We are also indebted to ATL for allowing us to use EMEA software under CECOM - ACIN III project. This work was completed under the Contract No.: DAAB07-01-9-L504.

References

1. Cohen, W.: Learning trees and rules with set-valued features, Proc. 13th AAAI, 1996
2. Gray, R. S., Kotz, D., Peterson, R. A., Barton, J., Chacon, D., Gerken, P., Hofmann, M., Bradshaw, J., Breedy, M. R., Jeffers, R., Suri, N.: Mobile-agent versus client/server performance: Scalability in an informationretrieval task. Picco, G. P., ed., 5th Int'l Conf. on MobileAgents, Lec. Notes in CS, 229-243, 2001
3. Forrest, S., Hofmeyr, S. A., Somayaji, A., Logstaff, T. A.: A Sense of Self for Unix process, Proceedings of 1996 IEEE Symposium on Computer Security and Privacy, 1996, 120-128
4. Jansen, W.: Countermeasures for mobile agent security. In Computer Communications, Special Issue on Advances in Research and Application of Network Security, November 2000
5. Lee, W., Stolfo, S.J., Chan, P.K.: Learning Patterns from Unix Process Execution Traces for Intrusion Detection, Proceedings of AAAI97 Workshop on AI Methods in Fraud and Risk Management, 1997, 50-56
6. Li, T.-Y., Lam, K.-Y.: Detecting Anomalous Agents in Mobile Agent System: A Preliminary Approach. AAMAS'02, July 15-19, 2002, Bologna, Italy
7. Sultanik, E., Artz, D., Anderson G., Kam, M., Regli W., Peysakhov M., Sevy J., Belov, N., Morizio, N., Mroczkowski, A.: Secure Mobile Agents on Ad Hoc Wireless Networks, The 15th Innovative Applications of Artificial Intelligence Conference (IAAI). August 11-13, 2003
8. Tan, H. K. and Moreau, L.: Certificates for mobile code security. In Proceedings of the 2002 ACM Symposium on Applied Computing SAC '02. ACM Press, New York, NY, 2002, 76-81

A Comprehensive Categorization of DDoS Attack and DDoS Defense Techniques*

Usman Tariq, ManPyo Hong, and Kyung-suk Lhee

Digital Vaccine and Internet Immune System Laboratory,
Graduate School of Information and Communication, Ajou University, Korea
{usman, mphon, klhee} @ajou.ac.kr

Abstract. Distributed Denial of Service (DDoS) attack is the greatest security fear for IT managers. With in no time, thousands of vulnerable computers can flood victim website by choking legitimate traffic. Several specific security measurements are deployed to encounter DDoS problem. Instead of specific solution, a comprehensive DDoS cure is needed which can combat against the previously and upcoming DDoS attack vulnerabilities. Development of such solution requires understanding of all those aspects which can help hacker to activate zombies and launch DDoS attack.

In this paper, we comprehensively analyzed the DDoS problem and we proposed a simplified taxonomy to categorize the attack scope and available defense solutions. This taxonomy can help the software developers and security practitioners to understand the common vulnerabilities that encourage the attackers to launch DDoS attack.

1 Introduction

Denials of service attacks are rapidly increasing threat for the productivity and the profitability of the internet infrastructure. The main aim of DoS attack is disruption of services by consuming the bandwidth of legitimate client. The scope of DoS attack is only dependent on the creator of this attack. The attackers achieve their goal by sending a stream of illegitimate packets to certain victim to cut the supply of legitimate packets. A highly flexible and distributed defense is the only solution to cope against this threat. During last few years, several high scale attacks had been launched to target high profile internet sites [1].

Different type of attacks made internet highly vulnerable. The most common type of attack is Denial of Service (DoS). Yankee group analyst's researchers estimated that total 2.7 billion \$ loss in year 2004. According to CSI/FBI studies, the average successful DDoS attack results in \$94,000 in damages due to lost productivity. While some enterprises may have devised temporary solutions to specific DDoS attacks, these makeshift alternatives are usually costly and always fail under a heavy, sustained attack. The sophistication of DDoS attacks today is such that no one piece of equipment can effectively cover an enterprise, as attacks often target multiple network layers simultaneously, using a variety of attack types.

* This research is supported by the Ubiquitous Computing and Network (UCN) Project, the Ministry of Information and Communication (MIC) 21st Century Frontier R&D Program in Korea.

Spoofed hosts used for the attack establishment are the key problem to mitigate the DoS attacks. Spoofing make the compromised machine undetected. An attacker can spoof packet within his domain even if an ISP poorly deploy the egress or ingress filtering.

If attack is coming from a constant source than the victim can block that IP but the attackers adopt a new technique called DDoS. In this technique, thousands of compromised computers from all around the world used to attack a particular victim.

It is open truth now that DDoS has become major threat to internet communication world. Widely available DDoS attacking software’s are destroying network traffic and on the other hand defense systems are not powerful enough to distract and mitigate all attack packets. Thus, DDoS problem will be more harmful in near future. Lack of evidence is also a beerier for the establishment of perfect defense solution. We proposed a well arranged and well defined DDoS attack and Defense taxonomy, which will help researchers to make a relatively better solution to combat against DDoS problem.

In section 2, we classified the DDoS attack categorization. In section 3, we illustrate the DDoS Defense mechanism categorization. In section 4 discussed the contribution of our proposed taxonomy. We focused on previous research work on DDoS taxonomy in section 5. Finally, in section 6 we conclude our research paper.

2 Categorization by DDoS Attack

To make an effective defense, it is highly recommended to know the classified nature of attacks. We described the attacks and classified these attacks into following domains. Domains: Level of Computerization, attack networks, Oppressed vulnerability, Influence of DDoS attack, attack Intensity dynamics and DDoS attack methods:

DDoS Attack Categorization			
Level of Computerization	Instruction Based Attack		
	Semi-Preset Attack	Direct Attack	
		Indirect Attack	
Preset Attack			
Attack Network	Agent Handler	Client-Handler Communication	TCP
			UDP
			ICMP
	IRC Based	Agent-Handler Communication	TCP
			UDP
			ICMP
Oppressed Vulnerabilities	Flood Attack	UDP Flood	Random Port
			Same Port
		ICMP Flood	

	Intensification Attack	Smurf Attack	
		Fragile Attack	Direct Attack Loop Attack
	Protocol Exploitation Attack	TCP SYN	
		PUSH + ACK	
	Address Spoofing	Routable Address Spoofing	
		Non-Routable Address Spoofing	
Malicious Formed Packet Analysis	IP Address		
	IP Packet Option		
Influence	Disorderly		
	Degrading		
Attack Intensity Dynamics	Continuous Attack Intensity		
	Variable Attack Intensity		

2.1 Categorization by Level of Computerization

According to level of computerization, DDoS attack can be classified in to instruction based; semi preset and preset attacks [4].

- ❖ Instruction based DDoS Attacks -- Initially all the DDoS attacks were Instruction based. Attacker has to perform scanning to find the vulnerability in expected slave machine, broke into that machine, install and command the malicious code by him.
- ❖ Semi-Preset DDoS Attacks -- In semi-preset DDoS attacks, the master computer scans the vulnerable machine and installs the malicious code on slave computer. Later attacker has to specify which type of attack he wants to establish. Semi-preset attacks can be further subdivided into two more branches: direct and indirect communication. In direct communication attack, the attacker hard code in the IP addresses of handler machine so that, both agent and handler machine become familiar with each other’s identity to communicate. A level of indirection is achieved in in-direct attacks by avoiding the identity from network scanners. IRC channels are the good example of in-direct communication attacks.
- ❖ Preset DDoS Attacks -- In preset attack, all the attack features like attack type and victim identity are specified into the attack code. The limitation of attacker phase reduced to attack command. In this way, attacker’s identity in most of cases will be unidentified.

2.2 Categorization by DDoS Attack Network

DDoS attack networks can be classified into two main types: Agent-handler model and IRC (internet relay chat) based model [5].

- ❖ Agent handler attack network is composed of client, handler and agents. Handlers are software packages spread all around the internet, used for indirect communication between the agents. Agent is located in slave machine which will finally establish the DDoS attack. The communication between client and handler or handler and agent takes place via TCP, UDP or ICMP protocols. The agents use negligible system resources due to which the computer users face nominal performance change.

- ❖ IRC-based DDoS attacks are similar to agent handler DDoS attack with the only difference that in agent-handler attack network use network server for handler installation. Later on, IRC communication channel is used for client and agent connectivity. In IRC-based DDoS attack architecture, the agents are referred as 'Bots'.

2.3 Categorization by Oppressed Vulnerability

DDoS oppressed vulnerability attacks can be divided into flood attack, intensification attack, protocol exploitation attack, address spoofing and malicious formed packet attack.

- ❖ In flood attack, the slave machine sends large amount of internet packets (IP) to victim to consume its bandwidth. This flood can either slow down the system or completely exhaust it. UDP and ICMP attacks [6] are famous for flooding attacks. In UDP flooding attack large numbers of UDP packets are sent to victim to exhaust its resources. UDP attack packets are usually sent to random ports but it can be sent to the same port. After receiving the UDP packets the victim system will find out the appropriate application which needs those UDP packets. When it finds out that there is no application waiting for that UDP packet at port, it generates the ICMP packets towards the destination. Most of the time the source IP is spoofed which makes ICMP packet unreachable. Thus huge amount of UDP packets can easily exhaust the system. ICMP flood attack exploits the internet control message protocol by sending large number of echo packets to the victim. During ICMP attack the source IP addresses are spoofed to hide the identity of attacker machine.
- ❖ In intensification attack, the attacker or agent exploits the routers IP address broadcast feature to send packets to a broadcast IP addresses. The bandwidth of victim is reduced due to malicious attack traffic. Attacker can send the broadcast message either directly or by using the agents to increase the traffic. Smurf and fragile attacks are famous examples of intensification DDoS attacks. Reflectors are used as an attack launcher in intensification attacks. Web servers, DNS servers and routers are said to be reflectors.
Fraggle attacks [7] are very similar to ICMP attack with the only difference that it uses UDP echo packets instead of ICMP echo packets.
- ❖ Protocol exploit attacks exploit a bug or a feature of some protocol installed at victim to consume extra resources. TCP SYN [8] is a real time protocol exploitation example.

Through IP spoofing the attackers generates a half open connection. These requests quickly exhaust the buffer space of connection and server stop accepting the further incoming packets. To keep the buffer occupied a steady amount of SYN packets are generated towards the victim. Random port TCP SYN flooding is another type of TCP SYN flooding attack.

- ❖ In malicious formed attacks [9], attackers use the incorrectly formed IP packets to exhaust the victim machine. In this attack, the IP packet contains the same source and victim address. This act confuses the operating system of victim machine, which leads to the system crash.

IP packet option malicious formed attack is another DDoS attack type. Attacker set all the 'Quality of Service' bits equal to 1 of IP packet. Because of generated vulnerability, victim has to perform extra processing in order to analyze the traffic.

2.4 Categorization by Influence

DDoS Influence attack can be divided into two domains: Disorderly Attack and degrading attacks.

- ❖ If there is complete cutoff of bandwidth is observed at Destination End than this DDoS attack is known to be Disorderly attack.
- ❖ If DDoS attack cause the partial bandwidth consumption than attack is said to be degrading attack. It is hard to detect degrading attack because it slowly cuts off legitimate bandwidth.

2.5 Categorization by Attack Intensity Dynamics

Based on attack intensity dynamics, a DDoS attack can be divided in to continuous and variable Attack Intensity attacks.

- ❖ During continuous attack intensity, same amount of IP packets are sent to victim that cause a collateral damage with out any break. The effects of this attack appear immediately but network agents configure attack in no time.
- ❖ It is clear with the name of variable attack intensity [10] that the density of attack varies through out the attack time. This feature makes the attack itself invisible from network defense mechanisms. The outcomes of such attack vary by attack density.

3 DDoS Defense Categorizations

DDoS is hard problem to solve because there is no common characteristic of DDoS attack. Distributed nature of attacks makes them difficult to combat or trace back.

DDoS Defense Categorization		
Submissive Defense Mechanism	Identifying Mechanism	Traffic Degree Monitoring
		Source IP Address Monitoring
		Packet attributes analysis
	Counter Mechanism	Filter
		Congestion Control
		Submissive Trace Back
Active Defense Mechanism	Base end defense	
	Mapping Trace Back	
	Packet Marking Trace Back	
	Protocol Based Defense	
Categorization by Action	Invasion Detection	
	Invasion Prevention	
	Invasion Response	
Defense Deployment Position	Basis End Network Defense	
	Intermediate End Network Defense	
	Destination End Network Defense	

More over attackers use the spoofed packets to hide their identity. This feature also makes the compromised machines hidden for ever. More over TCP protocol is vulnerable which helps attackers to follow the legitimate communication patterns to launch a successful DDoS attack.

3.1 Categorization by Submissive Defense Mechanism

Submissive defense mechanisms are those defense actions which triggers only after the DDoS attack is observed. Submissive defense mechanisms perform traffic limiting, blocking and filtering by observing inbound network traffic.

Submissive defense mechanisms can be classified into two categories: Identifying Mechanism and counter mechanism. Identifying Mechanisms include traffic Degree monitoring, source IP address monitoring and Packet attributes analysis.

After detecting a DDoS attack the detecting defense convert itself into the counter defense. Filtration of illegitimate IP packet from the traffic is one of famous DDoS defense technique.

3.1.1 Identifying Mechanism

We discussed before that Identifying mechanism can be classified into three techniques: traffic degree monitoring, source IP address monitoring and Packet attributes analysis.

- ❖ Traffic degree monitoring – Detects DDoS attack by monitoring sudden increase in network traffic degree. Detectable features of traffic are required for accurate attack detection [11].
 - Simple but fast, traffic degree monitoring is easy to deploy but it can not differentiate between real attack and flash crowds.
- ❖ Source IP address monitoring – Detects attack by monitoring the incoming packets IP address. This technique can differentiate between the flash crowd and spoofed IP addresses attack.
 - Invalid when attack comes from real source IP address. It is also less effective when attach traffic degree is extremely high.
- ❖ Packet attributes analysis – It detects attack by analysis of features of packet contents like ramp up, spectral contents etc. although the computational work is too complex to deploy but for some certain attack patterns it has good logical, precise pertinence.

3.1.2 Counter Mechanism

After detecting the attack signs, defense mechanism should perform some counter measures. Filtration, congestion control etc. are the counter defense methods which lie in the domain of submissive defense mechanism.

- ❖ Filtering [12] – After the attack detection the spoofed packets are dropped in routers. Defense mechanism can filter out the spoofed IP packet. It has potential to defend against highly distributed attack.
 - Filtering method is only effective when it is deployed on global scale. Such scheme is highly inappropriate when attack is coming from real source IP address.

- ❖ Congestion Control [13] – By analyzing the flow it regulates the traffic flow. Instead of solving the DDoS attack problem this technique solves only congestion problem. Congestion control technique is ineffective when a low bandwidth attack is observed. It can not differentiate between legitimate and malicious traffic.
- ❖ Submissive trace back [14] – Identifies the root of the DDoS attack source and start functionality after the attack begins. Compatibility to current network protocols makes it easy to install. The draw back submissive trace back technique is that it requires long time to reach attack source and heavy computation time to establish the path which makes it unsuitable for distributed defense scheme.
- ❖ Reproduction [15] – prepare some extra resources to bear the flood of attack packet and provide the resources to legitimate traffic. This technique requires no extra computational overhead. The only draw back of this scheme is that it's difficult to allocate extra resources and if attack nature is highly distributed, the resources still may exhausted by attack.

3.2 Categorization by Counter Defense Mechanism

Unlike the Submissive defense mechanism, counter defense mechanism tries to control the attack as soon as possible to reduce the damage. Counter defense methods usually share attacking information like attack signatures, so, the defense deployment should be on large scale.

Current active defense mechanisms are base end defense, Mapping trace back, packet marking trace back and protocol based defense.

- ❖ Base end defense [16] – Both the detection and filtrations components are deployed at source end network to avoid collateral damage due to attack outcomes. Base end defense methods are less sensitive to encounter the attack symptoms and some times faces high false alarm.
- ❖ Mapping trace back [17] – It collects the malicious packets to reconstruct the attack path. This method is highly effective even there is limited amount of attack packets. Mapping trace back methods also have some limitations. It requires huge storage space to store mapping data. More over it requires high processing time to construct the back path.
- ❖ Packet marking trace back [18] – Insert finger print of routers n-IP address into IP packet. For packet marking no extra storage is required and can later be used for packet filtering. It is highly suitable for distributed DoS attacks. Packet marking trace back encounters high difficulty when numbers of attack nodes are increased.
- ❖ Protocol-based defense [19] – After considering the protocol exploitation features used by attackers, researchers proposed several DDoS mitigation techniques. These proposed protocols are not widely deployed which is a barrier to control the DDoS problem fundamentally.

3.3 Categorization by Action

According to DDoS defense action, we divide the process into three categories.

- ❖ Invasion prevention – Best solution of DDoS attack is to completely eliminate the attack privileges. In invasion prevention method, researches tried to stop the

attack at its launching stage. Using globally coordinated filters, attack traffic can be blocked before there bombardment. Egress filtering [20], ingress filtering [21], router based packet filtering [22], history based IP filtering and source overlay services lies in global coordinated filters category.

- ❖ **Invasion Detection** – IDS detect attack by analyzing the old attack signatures or by recognizing the malicious system behavior. Data mining techniques, D-Ward base end defense, congestion triggered packet sampling and filtration and heuristic data structure [MULTOPS] [23] are some of defense technique which lies in invasion detection category.
- ❖ **Invasion Response** – After attack detection the defense system should identify the attack root and block the attack traffic. Both instructions based and preset invasion response systems are proposed and deployed by researchers. IP trace back, ICMP trace back, link-test, probabilistic packet marking, hash based IP trace back and traffic pattern analysis techniques are said to me famous and effective invasion response methods.

3.4 Categorization by Defense Deployment Position

DDoS attacks are usually high Degree IP traffic, so, sudden growth of traffic is one of the sign of DDoS attack. Communication always takes place between source and destination and we need a transitional node to establish communication between them. Logically we can classify and deploy the DDoS defense mechanisms at three different positions: source end network, transitional network and at Destination End network.

❖ **Basis Network Mechanism**

In this category the attack is detected before the attack traffic enters into internet layer. The early suppression of the attack minimizes the collateral damage. Defense is deployed near source machine which leads to easier trace back [24]. Routers close to source get small amount of packets, which facilitate more complex per-packet processing. Egress filtration eliminates the spoofed packets coming from nearest source [25]. Source-end defense saves bandwidth by drooping illegitimate traffic near the source machine. Prevention measurements can be applied easier and efficiently right after the alert is generated. It results false/positive because it's hard to decide about attack near source.

❖ **Transitional Network Mechanism**

The defense mechanisms deployed at transitional network can track the attack traffic back ward [26]. This phenomenon of back tracking is called pushback [27].

Deployment of defense solution at this position also has some disadvantage. The traffic at transitional routers is really high which prevents high computational sophisticated solutions installation. Transitional routers are highly resistive to huge traffic load so they some time do not feel the attack.

❖ **Destination Network Mechanism**

This is most important and highly deployed defense point. Attacker wants to target victim network. Destination End edge routers also observe less traffic than the intermediate routers. This advantage helps to deploy high computation algorithms to combat with attack. Protocol security mechanisms [28] and resource measurement

[29] are problems which are usually mitigated at Destination End defense. Trace out, rate limiting, Pi-Marking filtering etc. most effective at that position.

4 Contribution of Taxonomy

We presented a comprehensive taxonomy which classifies DDoS attacks, DDoS attacking tools and defense against DDoS attacks. Categorization of attacks mechanisms and defense solutions will provide the better understanding of this emerging problem. Upgraded knowledge will lead researchers to make more efficient solution. We also presented common information that hackers use to facilitate DDoS attack. This taxonomy will help security software programmers to well define the firewall rules which include some security checks for source and destination address.

During research it is observed that some of the attacks not only focus on wired network but also they disturb the wireless equipments. Fully understanding of attacker's behavior and attacking techniques can help researcher to defend their wireless network from the DDoS attack.

Taxonomy reviewers can predict future industry security fears and hackers approach to establish attacks. We suggest a proactive, real-time DDoS mitigation in which service provider detects attack and mitigates Influences on network rapidly; grounding the attack before network recourses are overwhelmed. With the passage of time attackers change their attack tools and attacking techniques. The taxonomy should be continuously upgraded as new attacks and defense mechanisms are available.

5 Related Work

DDoS attacks have evolved from random hacker exploits into organized criminal activities aimed to disrupt business and government organizations communication. In this section we first review the contributions of previously proposed DDoS attack and DDoS Defense taxonomies.

J. Mirkovic and P. Reiher [30] presented taxonomy of DDoS attacks and DDoS defense according to oppressed vulnerabilities, network communication, prevention measurements and defense deployment positions. Author classifies the attacks to mention commonalities and the important features of attack strategies that will lead to create a better solution. Defense categorization is done on the bases of currently available defense tools and the basic reason and the specific mitigation approach of that tool against DDoS attacks. In this paper authors also indicate some defense challenges and proposed that if these challenges can be overcome, we can completely eliminate the DDoS problem.

R. Ruby [31] defined attack architectures and then categorized the attack taxonomy by bandwidth depletion, recourse depletion. One key contribution of this paper is its DDoS software tools taxonomy. Authors categorized DDoS defense by Detect/prevent victim, Deflector attack and post attack forensics. The author describes the similarities and patterns of different DDoS attacks and the software used to launch the attack, to help the developer to create more generalized solution to fight against the DDoS attacks.

LC. Chen, TA. Longstaff and KM. Carley [32] focused only on DDoS attack and counter measure categorization. Authors classified DDoS attacks by Congestion based, anomaly based and source based techniques and defense is classified by destination network filtering and source work filtering.

P. Zaroo [33] is not published paper but it is cited by [34, 35] some of state of art papers about DDoS taxonomy. Author presented a very detailed survey about DDoS attacking and Defense techniques. The main contribution of Christos Douligeris research paper is that he identify the major attack tools available to establish the DDoS attack and evaluate them by there features and attacking nature. On the other hand Y Xiang just only focuses to the DDoS counter measures. The author described the main features, advantages and disadvantages of many defense mechanisms available in market.

6 Conclusion

For many years a point-to-point (P2P) denial of service attacks have been used by attackers to halt the flow of network traffic or to crash the system recourses. DDoS attack is the twist to similar theme. Most of DDoS attacks relay on trojan horse programs that reside in slave computer. The zombie machines used for DDoS attack are often located significant distance from one another and are using different ISP services.

Our current need is to make such DDoS solution which can encounter current and upcoming DDoS attacks. In, this paper, we analyzed the DDoS problem by attack nature and the defense mechanisms proposed to prevent the attack and proposed comprehensive taxonomy which can lead towards the development next generation DDoS security solution.

References

1. D. Moore, G. Voelker, S. Savage, Inferring Internet Denial of Service activity, *in: Proceedings of the USENIX Security Symposium, Washington, DC, USA, 2001, pp. 9-22.*
2. David Karig, Ruby Lee :Remote Denial of Service Attacks and Countermeasures: *Princeton University*
3. D. Davidowicz: Domain Name System (DNS) Security, 1999: <http://compsec101.antibozo.net/papers/dnssec/dnssec.html>
4. HH Lee, EC Chang, MC Chan : Pervasive Random Beacon in the Internet for Covert Coordination: <http://www.comp.nus.edu.sg>
5. Distributed Denial of Service attacks and their defenses: <http://www.lancs.ac.uk/postgrad/pissias/netsec/ddos/>
6. HCJ Lee, VLL Thing, Y Xu, M Ma: ICMP Traceback with Cumulative Path, an Efficient Solution for IP Traceback: *in Proceedings of the international conference on Information and Communication Security, Oct. 2003*
7. Smurf Attack and Fraggle Attack: <http://www.networkdictionary.com/security/SmurfAttack.php>
8. Andras Korn, Gabor Feher: RESPIRE – a Novel Approach to automatically Blocking SYN Flooding Attacks

9. Glenn Carl, George Kesidis, Richard R. Brooks, Suresh Rai: Denial-of-Service Attack-Detection Techniques: in Proceedings of the IEEE Computer Society, Jan./Feb. 2006
10. Xiapu Luo and Rocky K. C. Chang: On a New Class of Pulsing Denial-of-Service Attacks and the Defense
11. Intel, ReadySys, IP Fabrics: A Modular, Flexible Internet Traffic-Monitoring Solution for Networks of Today and Tomorrow An Advanced TCA®-Based Security Solution from RadiSys and IP Fabrics: *March 2005, ICSA Labs*.
12. Laura Chappell: Advanced Packet Filtering: <http://www.packet-level.com>.
13. Dongmei Wang, K.K. Ramakrishnan, Charles Kalmanek: Congestion Control in Resilient Packet Rings: *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP'04)*
14. Rajesh Kumar Dilli: Passive Monitoring and Detection of Spoofed IP attacks
15. M. Baentsch et al.: Enhancing the Web's Infrastructure: From Caching to Reproduction: *Proceedings of the IEEE Internet Computing, vol. 1, no. 2, 1997*
16. Jelena Mirkovic , Gregory Prier , Peter L. Reiher, Attacking DDoS at the Source, *Proceedings of the 10th IEEE International Conference on Network Protocols, p.312-321, November 12-15, 2002*
17. C Kai, H Xiaoxin, H Ruibing: DDOS SCOUTER: A SIMPLE IP TRACEBACK SCHEME: *Bell-labs Research China, Lucent Technologies, Beijing, China*
18. DX Song, A Perrig: Advanced and Authenticated Marking Schemes for IP Traceback: *Proceedings, IEEE INFOCOM 2001*.
19. S Kamara, D Davis, L Ballard, R Caudy, F Monroe: An Extensible Platform for Evaluating Security Protocols: *Proceedings of the 38th Annual Simulation Symposium (ANSS'05)*
20. Cisco PIX 500 Series Security Appliances: http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/prodlit/pix22_ds.pdf
21. Internet Security System: Distributed Denial of Service Attack Tools: <http://documents.iss.net/whitepapers/ddos.pdf>
22. A Yaar, A Perrig, D Song: Pi: A Path Identification Mechanism to Defend against DDoS Attacks: *In Proceedings of the IEEE Security and Privacy Symposium. IEEE Computer Society Press, Los Alamitos, Calif.*
23. .M. Gil, M. Poletto, MULTOPS: a data-structure for bandwidth attack detection, *in Proceedings of 10th Usenix Security Symposium, Washington, DC, August 13-17, 2001, pp. 23-38.*
24. Jelena Mirkovic, Gregory Prier, Peter L. Reihe: Source-End DDoS Defense*: *Proceedings of 2nd IEEE International Symposium on Network Computing and Applications, April 2003.*
25. P. Ferguson and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing," *RFC 2827, May 2000.*
26. K.A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, R.A. Olsson, Detecting Disorderly routers: a distributed network monitoring approach, in: *Proceedings of the 1998 IEEE Symposium on Security and Privacy, Oakland, CA, IEEE Press, New York, 1998, pp. 115-124.*
27. S. Floyd, S. Bellovin, J. Ioannidis, K. Kompella, R. Mahajan, V. Paxson, Pushback messages for controlling aggregates in the network, *Internet Draft, Work in progress, 2001.*
28. D.K. Yau, J.C.S. Lui, F. Liang, Defending against Distributed Denial of Service attacks with max-min fair server-centric router throttles, *in Proceedings of the Tenth IEEE International Workshop on Quality of Service (IWQoS), Miami Beach, FL, 2002, pp. 35-44.*

29. A. Garg, A.L.N. Reddy, Mitigating Denial of service Attacks using QoS regulation, in *Proceedings of the Tenth IEEE International Workshop on Quality of Service, 2002*, pp. 45-53.
30. J. Mirkovic, J. Martin, P. Reiher, A taxonomy of DDoS attacks and DDoS defense mechanisms, *UCLA CSD Technical Report no. 020018*.
31. SPECHT, S. M. and LEE, R. B. 2004. Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures. *Proc. PDCS, San Francisco, CA.*
32. LC Chen, TA Longstaff, KM Carley: A Taxonomy of DDoS Attack and DDoS Defense Mechanisms: *Computers and Security, 2004*
33. P. Zaroo, A survey of DDoS attacks and some DDoS defense mechanisms, *Advanced Information Assurance (CS 626)*.
34. C Douligeris, A Mitrokotsa :DDoS attacks and defense mechanisms: classification and state-of-the-art: *Proceeding of Computer Networks: The International Journal of Computer and Telecommunications Networking*
35. Y Xiang, W Zhou, M Chowdhury: A Survey of Active and Passive Defense Mechanisms against DDoS Attacks

Retracted: Structural Analysis and Mathematical Methods for Destabilizing Terrorist Networks Using Investigative Data Mining

Nasrullah Memon and Henrik Legind Larsen

Software Intelligence Security Research Center
Department of Software, Electronics and Media Technology
Aalborg Universitet Esbjerg
Niels Bohrs Vej 8, 6700 Esbjerg Denmark
{nasrullah, legind}@cs.aau.dk

Abstract. This paper uses measures of structural cohesion from social network analysis (SNA) literature to discuss how to destabilize terrorist networks by visualizing participation index of various terrorists in the dataset. Structural cohesion is defined as the minimum number of terrorists, who if removed from the group, would disconnect the group. We tested bottom-up measures from SNA (cliques, n-cliques, n-clans and k-plex) using dataset of 9-11 terrorist network, and found that Mohamed Atta, who was known as ring leader of the plot, participated maximum number of groups generated by the structural cohesion measures.

We discuss the results of recently introduced algorithms for constructing hierarchy of terrorist networks, so that investigators can view the structure of non-hierarchical organizations, in order to destabilize terrorist networks. Based upon the degree centrality, eigenvector centrality, and dependence centrality measures, a method is proposed to construct the hierarchical structure of complex networks. It is tested on the September 11, 2001 terrorist network constructed by Valdis Krebs. In addition we also briefly discuss various roles in the network i.e., position role index, which discovers various positions in the network, for example, leaders / brokers and followers.

1 Introduction

This paper introduces and studies the investigative data mining techniques; for example, cohesion analysis, role analysis and power analysis; and propose to construct hierarchy of non-hierarchical networks. (These analysis techniques are borrowed from social networks and graph theory.) In addition we also propose a mathematical method to find various position roles in the networks, for example, leaders, brokers and followers.

Cohesion analysis (also called structural cohesion) is often used to explain and develop sociological theories. Members of a cohesive subgroup tend to share information, have homogeneity of thought, identity, beliefs, behavior, even food habits and illnesses (Wasserman, S., Faust, K, 1994). Cohesion analysis is also believed to influence emergence of consensus among group members. Examples of cohesive sub-

groups include religious cults, terrorist cells, criminal gangs, military platoons, tribal groups and work groups etc.

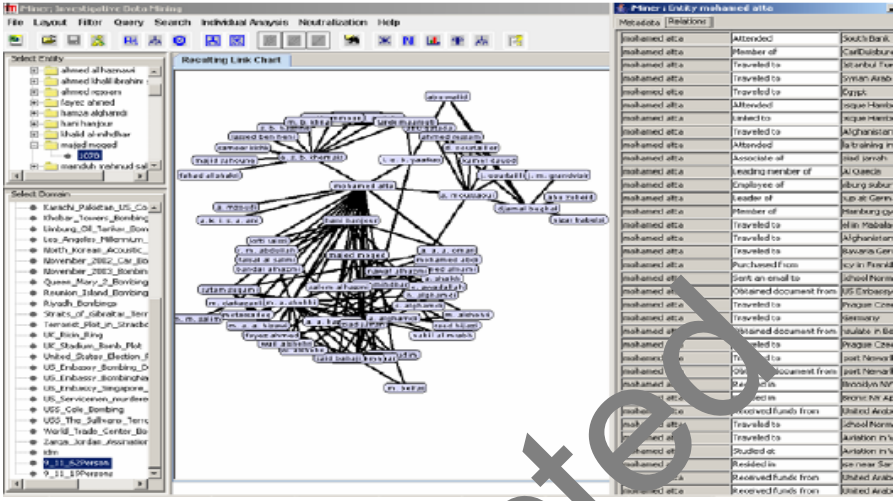


Fig. 1. The dataset of 9-11 hijackers and their affiliates. The dataset originally constructed by Valdis Krebs, but re-constructed in our investigative data mining prototype i.e. iMiner, using meta data of every terrorist.

Role analysis can be used for finding various roles in a network. On the other hand, power analysis is used to discover who is powerful / influential in the network.

Some direct application areas of social networks include studying terrorist networks (Sageman, M., 2004, Berry, N. et al., 2004), which is essentially an special application of criminal network analysis that is intended to study organized crimes such as terrorism, drug trafficking and money laundering (McAndrew, D., 1999, Davis, R. H., 1981). Concepts of social network analysis provide suitable data mining tools for this purpose (Chen, H., et al., 2004).

Figure 1 shows an example of a terrorist network, which maps the links between terrorists involved in the tragic events of September 11, 2001. This graph was constructed by Valdis Krebs (Krebs, V., 2002) using the public data that were available before, but collected after the event. Even though the information mapped in this network is by no means complete, its analysis may still provide valuable insights into the structure of a terrorist organization. This graph is reconstructed in this paper, using metadata of every terrorist involved in the attacks, using our investigative data mining software prototype known as iMiner.

The remainder of the paper is organized as follows: Section 2 discusses the background; Section 3 introduces about Investigative Data Mining and Social Network Analysis techniques. The Section 4 discusses practical approaches for neutralizing terrorist networks and Section 5 concludes the paper.

2 Background

After tragic terrorist attacks by kidnapped airlines on New York and Washington in September 2001 the interest for Al Qaeda in public and media rose immediately. Experts and analysts all over the world started to offer various explanations of Al Qaeda's origins, membership recruitment, modes of operation, as well as of possible ways of its disruption. Journalists in search of hot topics took over and publicized most of the publicly available materials, often revising them further and making them even more exciting and attractive for wide audiences.

One could thus read or hear that Al Qaeda is "a net that contains independent intelligence", that it "functions as a swarm", that it "gathers from nowhere and disappears after action", that it is "an ad hoc network", "an atypical organization" (Memon N., H. L. Larsen, 2006), extremely hard to destroy, especially by traditional anti-terrorist / counterterrorist methods.

One common criticism of efforts for analyzing terrorism by focusing on tensions in defined hierarchies is to argue that the current terrorist threat is not organized with clear lines of authority. Instead they are organized as loose networks and so belong to an analytically distinct category. According to many counterterrorism analysts today, Al Qaeda has evolved from a centrally directed organization into a worldwide franchiser of terrorist attacks (Grier P., 2005). Since war in Afghanistan, which significantly degraded Osama bin Laden's command and control, Al Qaeda does appear to have become increasingly decentralized. It is now seen by many as more of a social movement than coherent organization (Wikotorowicz Q., 2001).

Al Qaeda did not decide to decentralize until 2002, following the ouster of the Taliban from Afghanistan and the arrest of a number of key Al Qaeda leaders including Abu Zubaydhah, Al Qaeda's Dean of students, Ramzi bin Al Shibh, the organizer of the Hamburg cell of 9/11 hijackers, Khalid Sheikh Mohammed, the mastermind of 9/11 and the financier of the first World Trade Center attack, and Tawfiq Attash Kallad, the master mind of the USS Cole attack.

In response these and other key losses, Al Qaeda allegedly convened a strategic summit in northern Iran in November 2002, at which the group's consultative council decided that it could no longer operate as a hierarchy, but instead would have to decentralize (Joseph Felter et. al., 2005).

There is a need for tools which construct these non-hierarchical networks into hierarchical form, so that intelligence agencies and law enforcement officers can easily understand the structure of an organization.

Our recently introduced approaches and algorithms for Investigative Data Mining in the context of counterterrorism and homeland security (Memon, N., Larsen H. L., 2006) will be particularly useful for law enforcement and intelligence agencies that need to analyze terrorist networks and prioritize their targets.

3 Investigative Data Mining

Investigative Data Mining (IDM) offers the ability to firstly map a covert cell, and to secondly measure the specific structural and interactional criteria of such a cell. This framework aims to connect the dots between individuals and "map and measure

complex, covert, human groups and organisations". The method focuses on uncovering the patterning of people's interaction, and correctly interpreting these networks assists "in predicting behaviour and decision-making within the network".

The method also endows the analyst the ability to measure the level of covertness and efficiency of the cell as a whole, and also the level of activity, ability to access others, and the level of control over a network each individual possesses. The measurement of these criteria allows specific counter-terrorism applications to be drawn, and assists in the assessment of the most effective methods of disrupting and neutralising a terrorist cell. In short IDM "provides a useful way of structuring knowledge and framing further research. Ideally it can also enhance an analyst's predictive capability". Investigative Data Mining usually uses SNA techniques and graph theory connecting the dots in order to disconnect them.

Covert networks like terrorist networks remain mingled with socially oriented networks (like families, organizations etc.) in the real world. The buzz word for covert networks is "secrecy" and hence to discover such networks (technically, to discern distinctive patterns in the activities and communications of such dark networks) can be very tricky and often misleading due to unavailability of authentic data or in some cases availability of "doctored" data. This issue has especially blown up in the recent past and after the September 11, 2001 tragedy, it has been in the limelight so much so that it is worthwhile to take a close look at the distinguishing properties of such networks. For Example:

(1) In bright networks, actors who are highly central are typically the most important ones. On the contrary, peripheral players (or "boundary spanners" as they are typically called) may be huge resources to a terrorist group although they receive very low network centrality scores. This is because they are well positioned to be innovators, since they have access to ideas and information flowing in other clusters. Similarly, in an organization, these peripheral employees are in a position to combine different ideas and knowledge into new products and services. They may be contractors or vendors who have their own network outside of the company, making them very important resources for fresh information not available inside the company (Krebs V., 2002, Hanneman, R., 2000).

(2) The role of a "broker" (Krebs V., 2002) is a very powerful role in a social network as it ties two hitherto unconnected constituencies / groups together but of course, it is a single-point of failure. These broker type roles are often seen in terrorist networks. Such nodes are also referred to as "cutpoints" (Hanneman, R., 2000). These cutpoints may be further categorized as coordinator (The person connects people within their group), gatekeeper (The person is a buffer between their own group and outsiders. This person is known as influential in information entering the group) and representative (The person conveys information from their group to outsiders. This person is also known as influential in information sharing). We are still working on the algorithms to categorize the categories of brokers.

The main purpose of our research is to study and analyze the structure of terrorist networks in order to devise mathematical methods for destabilizing these adversaries (and to assist law enforcement and intelligence agencies), that is, minimum number of terrorists who, if removed from the network, would disconnect the network.

4 Approaches for Destabilizing Terrorist Networks

4.1 Cohesion Analysis

The aim of the detection of dense clusters is to find maximal subsets of points (with their relationships) with a high density in the cluster and relatively few relationships to other parts of the network. Graph theory gives a number of concepts and procedures that aims to detect maximal subgraphs in a graph (or network) that have a certain property and loses this property by adding another point and its relationships to the subgraph. In an undirected network, a *clique* is a maximal subgraph of at least three points in which all points are directly connected with one another. The concept clique has been generalized to *n-cliques*. In an *n-clique*, between any pair of points in the clique a path of length n or less exists in the graph. Such a path may go through points outside the clique, thus causing a larger distance between the points in the clique itself (or even disconnected cliques). An *n-clan* is an *n-clique* where the distance in the clique is also maximally n . In this paper we use bottom up approaches of cohesion analysis (cliques, n-cliques, n-clans, and k-plex) on a dataset shown in Figure 1.

Modeling a cohesive subgroup mathematically has long been a subject of interest in social network analysis. One of the earliest graph models used for studying cohesive subgroups was the *clique* model (Luce, R., Perry A., 1949). A clique is a subgraph in which there is an edge between any two vertices. However, the clique approach has been criticized for its overly restrictive nature (Scott, J, 2000), Wasserman, S., Faust, K., 1994) and modeling disadvantages (Siedman, S. B., Freeman, L. C., 1992).

Alternative approaches were suggested that essentially relaxed the definition of cliques. Clique models idealize three important structural properties that are expected of a cohesive subgroup, namely, *familiarity* (each vertex has many neighbors and only a few strangers in the group), *reachability* (a low diameter, facilitating fast communication between the group members) and *robustness* (high connectivity, making it difficult to destroy the group by removing members).

Different models relax different aspects of a cohesive subgroup. Luce R. introduced a distance based model called *n-clique* (Luce, R., 1950). This model was also studied along with a variant called *n-clan* by Mokken (Mokken, R., 1979).

However, their originally proposed definitions required some modifications to be more meaningful mathematically.

Table 1. Statistics from the results

	Groups	Groups , max. size	Groups , min. size
clique	41	6	3
n-clique	38	23	5
n-clan	22	23	5
k-plex	493	7	3

These drawbacks are pointed out and the models are appropriately redefined in (Balasundaram, B. et al, 2005). All these models emphasize the need for high reachability inside a cohesive subgroup and have their own merits and demerits as models of cohesiveness. In this paper we also discuss on a degree based model and called *k-plex* (Wasserman, S. et al, 2004). This model relaxes familiarity within a cohesive subgroup and implicitly provides reachability and robustness.

In this paper we discuss the use of the four concepts: cliques, *n*-cliques, *n*-clans, and *k*-plex. The dataset, which describes the network from Figure 1, has been applied to each of the four concepts. The statistics from the results are listed in Table 1.

Each node represents a specific person from the dataset, so the number of nodes should be the same for all concepts. Each concept generated different number of groups: for example, *n-clan* concept generates 22 groups while the *k-plex* concept generates 493 groups. The *n-clique* generates 38 groups and the *clique* concept generates 41 groups. For each of the concepts the maximum size and minimum size of a group has also been collected and shown in Table 1.

The statistics indicates that even with relative small dataset a huge number of groups could be generated. The groups generated are analyzed, in order to identify the best candidate nodes for destabilizing the specific network.

Figure 2 shows to how many groups each member is participated, using respectively clique, *n*-clique, *n*-clan and *k*-plex. As we can see some of the members are participated to many groups while other members are participated in few groups. We say that a member being participated to many groups, compared to the total number of groups, has a *high participation index*, while a member participated in a few groups, compared to the total number of group has a *low participation index*. Participation index is defined as participation of a particular member of in different groups generated by the various concepts of structural cohesion / cohesion analysis.

For example we take a look at a member Mohamed Atta (node 33) in the matrix generated using the *k*-plex concept, has participated in 230 groups and the total number of groups is 493. This give a participation index equal to 230/493, approximately 0.467.

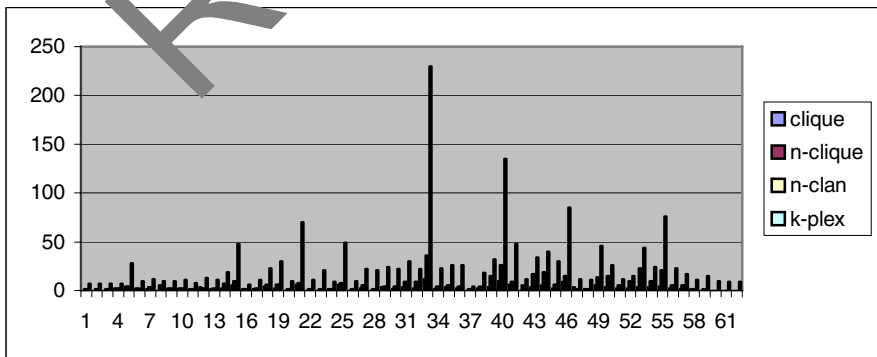


Fig. 2. The participation of the members of the 9-11 terrorists network in various groups using the concepts Clique, *n*-clique, *n*-clan and *k*-plex

Table 2. Participation index

	Member 33	Member 37	Member 55
clique	0.293	0.000	0.098
n-clique	0.947	0.026	0.553
n-clan	0.909	0.045	0.455
k-plex	0.467	0.008	0.152

If the participation index is closer to 1, it means that that member has participated in most of the groups, and if the participation index is closer to 0, it means that the member's participation is negligible.

From the variation seen in the participation index we conclude that the choice of concept has an important influence on the participation index. It seems like using the concepts n-clique and n-clan results in higher participation indices, while the concepts clique and k-plex results in lower participation indices.

The three members described in the Table 2, can roughly been seen as a picture of arch types or roles in the network. In most cases a member is not 100 percent an arch type, but a combination of the three types. What type a member will match best in a specific situation will also be dependent on other factors, e.g. the phase of the operation conducted by the network.

The arch types are named brokers (gatekeeper, representative or coordinator), leaders and followers. Brokers encompass members working with logistics, communications, etc. Leaders encompass leaders at all levels, using the military terms this means officers as well as NCOs. Followers encompass the members that can be compared to the infantry in military terms.

The task of the broker is to provide supplies of weapon, money, identity card, etc. to the network. Often a broker is also preparing houses and cars for the network. The broker sometimes is the member being the secure communication between the different groups in the network. As such the broker is often related to a large number of groups in the network, since a key member in setting up the platform for the operation. Member 33 could as such be an example of a broker.

The task of the leader is, of course, to lead one or more groups in the network. As described, leaders in the network can be found at several levels, from the member leading a group to the leader running the network. Leaders tend to "hide in the crowd", and in some cases they are related to a large number of groups, in other cases they are related to only a small number of groups. As such they can be harder to find. Though in most cases the leader related to many groups, still will have a lower participation index than the broker, and the leader related to few groups will have a participation index higher than the followers. Nawaf Alhazmi (node # 55) could as such be an example of a leader.

The task of the follower is to be the executing part, or the muscles to say in a popular way. A follower is following orders from leaders has usually very limited knowledge about the overall plan. The follower is member of just a few groups since he has

no direct importance for other groups in the network, like for instance the broker. Mamduh Salem (node # 37) could as such be an example of a follower.

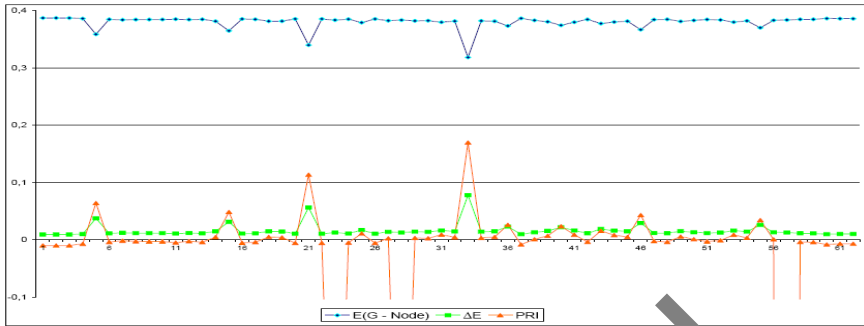


Fig. 3. The efficiency of the original network $E(G) = 0.395$. The removed node is shown on x-axis, the efficiency of the graph once the node is removed is shown as $E(G - \text{node})$; while importance of node, i.e. the drop of efficiency is shown as ΔE . The newly introduced measure position role index is shown as PRI.

4.2 Role Analysis

In role analysis we'll discover who is who in a network.

The Efficiency $E(G)$ of a network

The network efficiency $E(G)$ is a measure to quantify how efficiently the nodes of the network exchange information (Latora, V., et al, 2004). To define efficiency of G first we calculate the shortest path lengths $\{d_{ij}\}$ between two generic points i and j . Let us now suppose that every vertex sends information along the network, through its edges. The efficiency ϵ_{ij} in the communication between vertex i and j is inversely proportional to the shortest distance: $\epsilon_{ij} = 1/d_{ij} \forall i, j$ when there is no path in the graph between i , and j , we get $d_{ij} = +\infty$ and consistently $\epsilon_{ij} = 0$. N is known as the size of the network or the numbers of nodes in the graph. Consequently the average efficiency of the graph of G can be defined as (Latora, V., et al, 2004):

$$E(G) = \frac{\sum_{i \neq j \in G} \epsilon_{ij}}{N(N-1)} = \frac{1}{N(N-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}} \tag{1}$$

The above formula gives a value of E that can vary in the range $[0, \infty]$, while it be more practical to normalize E in the interval of $[0, 1]$.

The Critical Components of a network

Latora V. et al recently proposed a method to determine network critical components based on the efficiency of the network briefly discussed in the previous subsection. This method focuses on the determination of the critical nodes. The general theory and all the details can be found in Ref. (Latora, V., et al, 2004).

The main idea is to use as a measure of the centrality of a node i the drop in the network efficiency caused by deactivation of the node. The importance I (node_i) of the i th node of the graph G is therefore:

$$I(\text{node}_i) \equiv \Delta E = E(G) - E(G - \text{node}_i), i = 1, \dots, N, \quad (2)$$

Where $G - \text{node}_i$ indicates the network obtained by deactivating node i in the graph G . The most important nodes, i.e. the critical nodes are the ones causing the highest ΔE .

Position Role Index (PRI)

The PRI is our newly introduced measure (Memon, N., Henrik, L. L., 2006) which highlights a clear distinction between followers and gatekeepers (It is a fact that leaders may act as gatekeepers). It depends on the basic definition of efficiency as discussed in equation (1). It is also a fact that the efficiency of a network in presence of followers is low in comparison to their absence in the network. This is because they are usually less connected nodes and their presence increases the number of low connected nodes in a network, thus decreasing its efficiency.

If we plot the values on the graph, the nodes which are plotted below x-axis are followers, whereas the nodes higher than remaining nodes with higher values on positive y axis are the gatekeepers. While the nodes which are on the x-axis usually central nodes, which can easily bear the loss of any node. The leaders tend to hide on x-axis there.

We applied this measure on the network of alleged 9-11 hijackers (Krebs, V., 2001) and results are shown in Figure 3.

It is to note that after introduction of these measures, now it is possible to find the efficiency of a network as well as if we remove a particular node, then intelligence agency can find how much efficiency of the network is affected.

The analysis shows that Mohamed Atta (node 33) was ring leader of the plot, that is, he played his role as broker in the network. But we still need to know who was influencing who in the 9-11 plot. For this we analyze the influence of the nodes using power analysis.

4.3 Power Analysis

As terrorists establish new relations or break existing relations with others, their position roles, and power may change accordingly. These node dynamics resulting from relation changes can be captured by a set of centrality measures from SNA. The centrality measures address the question, "Who is the most important or central person in the network?" There are many answers to this question, depending on what we mean by important. Perhaps the simplest of centrality measures is *degree centrality*, also called simply *degree*.

Though simple, *degree* is often a highly effective measure of the influence or importance of a node: in many social settings people with more connections tend to have more power.

A more sophisticated version of the same idea is the so-called *eigenvector centrality* (which is also known as centrality of a centrality). Where degree centrality gives a

simple count of the number of connections a vertex has, eigenvector centrality acknowledges that not all connections are equal.

To neutralize terrorist network, we have used centrality measures from SNA literature i.e. degree centrality and Eigen vector centrality.

Using undirected graph (as shown in Figure 1), we first convert it into directed graph using degree centrality and Eigenvector Centrality. For Example, if degree centrality of one node is higher than other, then simply the directed link is originated from that node and point towards other. If they are equivalent in terms of degree, the link will originate from the node with higher Eigenvector centrality. If Eigenvector centrality values for both nodes are equal, then we ignore the link.

Then we identify the parents and children pairs. For example, if we have two nodes, which are competing for being parent of a node, then we have to identify its correct parent. The correct parent will be the one which is connected with maximum neighbours. This represents the fact that the true leader, with respect to a node, which is more influential on its neighbourhood.

When we identify parents, in such a way we traverse all the nodes. Then a tree structure is obtained using dependence centrality, which we call *hierarchical chart*. For more details about algorithms used for conversion of the network (un-directed graph) to hierarchical chart as shown Figure 4, the details are available in our recently published article (Memon, N. and Henrik L. Larsen, 2006).

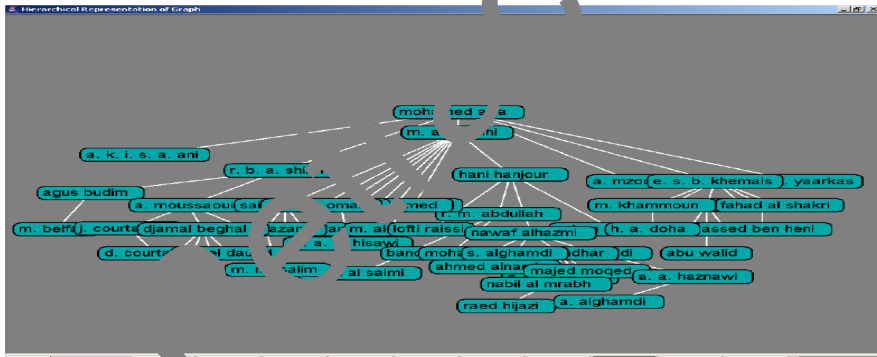


Fig. 4. The hierarchy clearly suggests that Muhammad Atta was the most powerful person (leader) of the plot. While M.A. Shehhi was assisting him, as he is below in the hierarchy. They both were found the key leaders of the plot by 9-11 commission.

Dependence centrality (Memon N. and Henrik L. L, 2006) of a node is defined as how much that node is dependent on any other node in the network. Mathematically it can be written as:

$$DC_{mn} = \sum_{m \neq p, p \in G} \frac{d_{mn}}{N_p} + \Omega \tag{2}$$

Where **m** is the root node which depends on **n** by **DC_{mn}** centrality and **N_p** actually is the Number of geodesic paths coming from **m** to **p** through **n**, and **d_{mn}** is geodesic

distance from m to n . The Ω is taken 1 if graph is connected and 0 in case it is disconnected. In this paper we take Ω as 1, because we consider that graph is connected. The first part of the formula tells us that:

How many times m uses n to communicate other node p of the network? In simple words p is every node of the network, to which m is connected through n (The connection represents the shortest path of node m to p , and n is in between). N_p represents the number of alternatives available to m to communicate to p and d_{mn} is the multiplicative inverse of geodesic distance ($1/d$).

5 Conclusion

In this paper we have discussed structural analysis and mathematical methods for destabilizing terrorist networks, which will assist law enforcement agencies in understanding the structure of terrorist networks. We presented three different approaches for destabilizing terrorist networks. The cohesion analysis of the dataset shows that Mohamed Atta participated in maximum groups generated by structural cohesion measures, which is a clear indication of working of this node as supplier/ broker / gatekeeper in the network. In reality he was also an important and found as ring leader by 9-11 commission.

The position role index measure assists in finding about who is who in a network (for example, leaders, gatekeepers and followers). This measure also proved that the role of Mohamed Atta was an important and he worked as gatekeeper. The importance of this node can be seen from Figure 3, because deactivating the node, the efficiency of the graph is drastically decreased.

The power analysis concept also proved that Mohamed Atta was the most powerful node in the network and worked as leader in the network and this node is shown as the top node in the hierarchical chart generated by the algorithms recently introduced by the authors.

The mathematical methods and algorithms discussed in the paper are implemented in the investigative data mining prototype known as iMiner. The *iMiner* demonstrates key capabilities and concepts of a terrorist network analysis tool. Using the tool investigating officials can predict overall functionality of the network along with key players. Thus counterterrorism strategy can be designed keeping in the mind that destabilization not only means disconnecting network but disconnecting those key players from the peripheries by which maximum network could be disrupted.

References

1. Balasundaram, B., Butenko, S., Trukhanov, S.: Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization* 10, 23–39, 2005.
2. Berry, N., Ko, T., Moy, T., Smrcka, J., Turnley, J., Wu, B.: Emergent clique formation in terrorist recruitment. The AAI-04 Workshop on *Agent Organizations: Theory and Practice*, July 25, 2004, San Jose, California, 2004.
3. Bonacich, P., Power and Centrality. *American Journal of Sociology* 92: 1170-1184, 1987.
4. Burt, R. S., Structural Holes, *Cambridge, MA: Harvard University Press*, 1992.

5. Burt, R. S., Structure, A General Purpose Network Analysis Program. *Reference Manual*, Newyork: Columbia University, 1990.
6. Chen, H., Chung, W., Xu, J.J., Wang, G., Qin, Y., Chau, M.: Crime data mining: A general framework and some examples. *Computer* 37(4), 50–56, 2004.
7. Davis, R.H.: Social network analysis: An aid in conspiracy investigations. *FBI Law Enforcement Bulletin* pp. 11–19, 1981.
8. Freeman, L.C.: The sociological concept of “group”: An empirical test of two models. *American Journal of Sociology* 98, 152–166 ,1992.
9. Grier, P. “The New Al Qa’ida: Local Franchiser,” *Christian Science Monitor* (11 July 2005). Online at: <http://www.csmonitor.com/2005/0711/p01s01-woeu.html> (Accessed on May 26, 2006).
10. Hanneman, R. E., Introduction to Social Network Methods. *Online Textbook Supporting Sociology 175*. Riverside, CA: University of California, 2000.
11. Joseph Felter et. al., *Harmony and Disharmony: Exploiting al-Qa’ida’s Organizational Vulnerabilities* (West Point, N.Y.: United States Military Academy, 2006), p. 7-9.
12. Robert Windrem, “The Frightening Evolution of al-Qa’ida,” *MSNBC.com*, (24 June 2005). Online at: <http://msnbc.msn.com/id/8307333> (Accessed on May 26, 2006).
13. Krebs, V.: Mapping networks of terrorist cells. *Connections* 24, 45–52, 2002.
14. Latora, V., Massimo Marchiori How Science of Complex Networks can help in developing Strategy against Terrorism, *Chaos, Solitons and Fractals* 20, 69–75, 2004.
15. Luce, R., Perry, A.: A method of matrix analysis of group structure. *Psychometrika* 14, 95–116, 1949.
16. Luce, R.: Connectivity and generalized cliques in sociometric group structure. *Psychometrika* 15, 169–190, 1950.
17. McAndrew, D.: The structural analysis of criminal networks. In: D. Canter, L. Alison (eds.) *The Social Psychology of Crime: Groups, Teams, and Networks, Offender Profiling Series, III. Aldershot, Dartmouth*, 1999.
18. Memon, N., Detecting Terrorist Related Activities using Investigative Data Mining Tool, In Proceedings of Symposium 5, Data/Text Mining from Large Databases IFSR 2005, Kobe, Japan, 2005.
19. Memon, N. Henrik Legind Larsen, Practical Algorithms for Destabilizing Terrorist Networks, *Lecture Notes in Computer Science (LNCS)* 3975, ISI 2006, Eds. S. Mehrotra et al. pp. 389-400, 2006.
20. Mokken, R.: Cliques, clubs and clans. *Quality and Quantity* 13, 161–173, 1979.
21. Newman, M. E. J. The structure and function of complex networks, *SIAM Review* 45, 167-256, 2003.
22. Scott, J.: *Social Network Analysis: A Handbook*, 2 edn. Sage Publications, London 2000.
23. Seidman, S.B., Foster, B.L.: A graph theoretic generalization of the clique concept. *Journal of Mathematical Sociology* 6, 139–154, 1978.
24. Sageman, M.: *Understanding Terrorist Networks*. University of Pennsylvania Press, 2004.
25. Wasserman, S., Faust, K.: *Social Network Analysis*. Cambridge University Press.1994.
26. Wiktorowicz, Q. "The New Global Threat: Transnational Salafis and Jihad," *Middle East Policy* 8, no. 4 (2001: 18-38)

Alert Correlation Analysis in Intrusion Detection

Moon Sun Shin¹ and Kyeong Ja Jeong²

¹ KonKuk University, Korea
msshin@kku.ac.kr

² ChungCheong University, Korea
kjeong@ok.ac.kr

Abstract. With the growing deployment of host and network intrusion detection systems, managing the reports from these systems become critically important. Current intrusion detection systems focus on low-level attacks or anomalies. As a result, it is difficult for users or intrusion response systems to understand the intrusion behind the alerts and take appropriate actions. In this paper, we propose alert correlation analysis based on data mining techniques for the management of alerts. Because data mining tasks deal with the discovery of implicit data, we can discover the interconnection and inter relationships among the alerts. So the results of analyzing the alert data are used for the security policy server to construct the security policy rule efficiently in the framework of PBNM(Policy Based Network Management). It helps not only to manage the fault users and hosts but also to discover possible alert sequences.

1 Introduction

Recently, due to the open architecture of the internet and wide spread of the internet users, the cyber terror threatening to the weak point of network tends to grow [1,2]. Until now the information security solutions have been passive on security host and particular security system.

Intrusion detection systems collect the information from various advantages within network, and analyze the information. There are no perfect intrusion detection systems or mechanisms, because it is impossible for the intrusion detection systems to get all the packets in the network system. Especially, the unknown attacks can hardly be found. Currently building the effective IDS is an enormous knowledge engineering task. Recent data mining algorithms have been designed for the application domains involved with the several types of objects stored in the relational databases. All IDSs require a component that produces basic alerts as a result of comparing the properties of an input element to the values defined by their rules. Most of systems perform the detection of basic alarms by each input event to all rules sequentially, and IDSs raise the alarm when possible intrusion happens. Consequently, IDSs usually generate a large amount of alerts that can be unmanageable and also be mixed with false alerts. Sometimes the volume of alerts is large and the percentile of the false alarms is very high. So it is necessary to manage alerts for the correct intrusion detection. As a result, nearly all IDSs have the problem of managing alerts, especially false alarms, which cause seriously to impact performance of the IDSs. A general solution to this problem is needed. We describe an approach that extract high-level alert correlation rules by

applying data mining techniques. The data mining techniques is to discover useful information from huge databases. We provide the groundwork to apply the data mining techniques for alert correlation analysis.

The rest of this paper is organized as follows. Section 2 describes alert correlation of IDS as related works. In section 3 we propose an alert data mining framework for alert correlation analysis. Section 4 and 5 presents the implementation and the experiments of our system. In the last section, we will summarize our works.

2 Security Policies and Alert Correlation

Alert Correlation

IDSs products have become widely available in recent years. These systems monitor hosts, networks and critical files and these systems deal with a potentially large number of alerts. These systems should report all alerts to the security policy server or operator. So the security policy server has to manage the reporting alerts in order to build the new security policy rule. But current intrusion detection systems do not make it easy for operators to logically group the related alerts. Also the existing intrusion detection systems are likely to generate false alerts, be it false positive or false negative. To solve these critical problems, the intrusion detection community is actively developing standards for the content of the alert messages and some researches is on going about the alert correlation. In [4] they introduced probabilistic approach for the coupled sensors to reduce the false alarm. An aggregation and correlation algorithm is presented in [3] for acquiring the alerts and relating them. The algorithm could explain more condensed view of the security issues raised by the intrusion detection systems.

The current intrusion detection systems usually focus on detecting the low-level attacks and/or the anomalies. None of them can capture the logical steps or attack strategies behind these attacks. Whereas actual alerts can be mixed with false alerts and also the amount of alerts will also become unmanageable. As a result, it is difficult for the human users or intrusion response systems to understand the intrusions behind the alerts and take the appropriate actions. In [6] they propose the intrusion alert correlator based on the prerequisites of intrusions. This paper presents applying data mining techniques to analyze the alert correlation.

Security Policies

This section describes security policies that enable intrusion detection and automated responses. As mentioned above, policy rules are typically expressed as condition/action pairs (or if/then clauses). Security policy rules are also expressed as same manner. This can be generated from the administrators or the security manager and enforced at the security agents.

This paper classifies the security policy into two categories as “static” and “instant” from its behavior. Static policy rules stored in repository servers and loaded to each security agent. It is built up by administrators to reflect their volition. On the other hand, instant policy rules are generated instantly by resulting of automated analysis. It is adopted temporally and recovered when the situation is over. That is instant policy rules are not kept in repository servers and used temporally if needed.

The security policy comprises multiple sections. Different policy sections cover different concerns, such as intrusion detection, alert control, or filtering control. The following sections describe security policy functionality and mechanisms.

Intrusion Detection Policy provides to detect intrusions. This is the most general and basic policy for security management and can be defined within security signatures that are provided by the security agent. The intrusion detection policy rules define the attack types. That is the rules understood as detection requirements by security agents. For the same attack type, administrators direct different actions as basis of the importance of the asset. This means this policy defines intrusion detection condition and value of network resources by describing different actions. Figure 1 shows the intrusion detection policy rule examples. The example 1) defines DOS Teardrop attack and the example 2) and 3) define FTP password retrieval attempt attack. From the example 2) and 3), the actions can be described differently even though the attack is same as the FTP password retrieval attempt. Against the attack, alert message sending is just directed in general. For the particular host, however, the corresponding actions are defined as blocking the packets and sending alert messages.

<p>1) If Protocol is UDP and Id is 242 and FragBit sets 'M', Then Alert and Block;</p> <p>2) If Protocol is TCP and TCPFlags set 'ACK' and Content includes "RETR" and "passwd", Then Alert;</p> <p>3) If Protocol is TCP and TCPFlags set 'ACK' and Content includes "RETR" and "passwd" and Target is 129.254.1.1/32, Then Alert and Block;</p>
--

Fig. 1. Example intrusion detection policy

Alert Control Policy provides alert traffic reduction mechanisms to protect resources such as network bandwidth, processing power and so on. This policy consists of two parts of alert suppression and aggregation. Intrusion detection system stores attack information to databases, it called alert data. The operator is able to analyze alert correlation, create attack information and make statistical data using alert data. Attributes of alert data are listed in Table 1.

Table 1. Object Attributes of Alert Data

Object Attribute	Contents	Object Attribute	Contents
SGSID	ID of SGS	ATTACKID	Signature ID
ATTACKTYPE	Category of Attack	DETECTDATE	Intrusion detection date
DETECTTIME	Intrusion detection time	SRCADDR	Source IP address
SRCPORT	Source port number	PROTOCOL	Protocol
TARGETADDR	Target IP address	TARGETPORT	Target port number
IMPACT	Intensity of attack	TYPE	Kind of attack
ACTCATEGORY	Kind of response	URL	Reference URL

The alert suppression mechanism reduces repetitive alerts that can cause administrator overload. In some instances, security agents may generate a large number of alerts that are all related to the same event. For example, a full port scan of a subnet can cause detectors to emit thousands of essentially identical port scan reports. These alerts can be used to flood the alert messaging and thereby inhibit automated response. These alerts also flood the administrator's console with unnecessary information. The alert control policy provides alert suppression and this can be accomplished by the administrators and the result of the automated responses. This mechanism can also be used for friendly vulnerability scanning. Sometimes organizations often perform friendly scanning to help identify and close security vulnerabilities. For this friendly vulnerability scanning, this mechanism can summarize the port scanning alerts from the friend devices. The alert aggregation mechanism enables an administrator to aggregate a set of alerts into one alert message. To protect valuable resources, the administrators command to perform alert aggregation when alerts are generated exceeding threshold within a period. Then the security agent received the policy rule performs aggregation whenever it faces that situation. Figure 2 shows alert suppression and alert aggregation examples. For the first example, the security agent sends the first port scan alert, and then sends an suppressed alert after 30 times or 30 seconds, whichever comes first. This would reduce the number of alerts in a full port scan by a factor of 30. The second policy rule commands to perform alert aggregation by every minute against the port probing of the particular host.

```

1) If attack is PortProbing and Target is 129.254.1.1/32,
Then AlertSuppress(times=30, time=30);

2) If attack is PortProbing and Target is 129.254.1.1/32,
Then AlertAggregate(time=60);

```

Fig. 2. Example alert control policy

3 Alert Correlation Analysis Based on Data Mining

In this chapter, we describe the outline of an alert data mining framework for alert correlation analysis of IDS. The proposed alert data mining framework consists of four components such as the association rule miner, the frequent episode miner, the clustering miner and sequential pattern miner. In order to perform alert correlation analysis the four components performs each tasks. The association rule miner can find the correlation among the attributes in the record, although the frequent episode miner searches event patterns in records. And sequential pattern miner searches the sequential pattern of alert. In addition, the clustering miner discovers the similar attack patterns by grouping the alert data with similarity among the alert data. The clustering analysis provides the data abstraction from the underlying structure. And it groups the data objects into the clusters so that the objects belonging to the same cluster are similar, while those belonging to the different cluster are dissimilar.

Because we consider the characteristics of the alert data, we improve the existing data mining algorithms to create the candidate item sets that include the only interesting attributes.

Axis Based Association Rule Mining and Frequent Episodes Mining

The existing association rule mining algorithms search for interesting relationships in the transaction database. We expanded the Apriori algorithm without grouping the items by T_id because of the characteristics of the alert data, so we need to make the rules only with attributes of interest.

Mining frequent episodes is to search a series of event sequences for the frequently occurring episodes. Using episodes, an infiltration detection system can detect the frequently repeated patterns, and apply them to the rule or use them as the guidelines for the service refusal attacks. When the existing algorithms are used to apply the data mining to a search for the useful patterns from the alert data, the correlations among the attributes must be considered. The alert data comprise the various attributes, and each of these attributes has many values. Because all of these data cannot be converted into a binary database, we propose an expanded algorithm using row vector. Row vector is the data structure used in searching for frequent items, which contains bits recording transactions that include the item sets. Using row vector has the advantage to consider the correlations among the tuples rather than the correlations among the attributes. In addition, as the standard attributes are applied, only the items including the standard attributes have to be considered in generating the candidate items. This reduces the number of the unnecessary episodes in generating the rules. Table 2 presented an example of the final rules after mining task.

Table 2. Example of Final Rules

(a) Association Rule	
Association Rule	Meaning
50<=>21 (supp:49, conf:100%)	Attribute 50(Atid) correlated with attribute 21(dsc_port)
21<=>tcp (supp:49, conf:100%)	Attribute 21(dsc_port) correlated with attribute tcp(protocol)
(b) Frequent Episodes Rule	
Frequent Episode Rule	Meaning
5001:210.155.167.10:21:tcp => 5007:210.155.167.10.21 :tcp (fre:10, conf:100%, time:10sec)	If 5001(Ftp Buffer Ovrflow) occur, then 5007(Anonymous FTP) occur together.

Sequential Pattern Search of Alerts

Sequential pattern miner mines sequential pattern of alert data using PrefixSpan algorithm after select necessary attributes. Especially, sequential pattern miner create sequence database via preprocessing process that consider attribute of alert data and mines frequent scenario.

Alert analyzer is able to predict sequence behaviors and change patterns of sequences that were not visibly checked by sequential pattern mining in the new sequence sets.

The algorithm is to acquire the sets of sequential patterns from 1 to l-length when alert data and minimum support are given. In this algorithm, the subsets of sequential patterns in all available lengths are acquired with the input values of the sequences in preprocessed sequence database and by preprocessing converting alert data to sequence database.

Order Based Clustering Miner

A clustering mining system that analyzes the similarity of the alert data is composed of Data Processor, Alert Cluster, Cluster Analyzer and Alert Classifier. Data Preprocessor preprocesses the input dataset so that the dataset can be clustered by Alert Cluster. Here, extended attributes based on domain knowledge are added for efficient and accurate clustering, and selected attributes are normalized. Alert Cluster clusters the data preprocessed by Data Preprocessor. The final output of this module is a set of grouped data. The output is stored in the rule database, and is used in the automatic classification of the new alerts and the analysis of the relationships among generated clusters. Cluster Analyzer analyzes the causes of generation of clusters. The output of the module is represented by the sequence of clusters. The output is used in analyzing the relationships among clusters, and in predicting a set of possible alerts for a specific alert. Alert Classifier classifies new alerts into appropriate clusters using the cluster model generated by Alert Cluster, and abstracts possible alerts to occur next by using the sequence generated by Cluster Analyzer.

4 Experimental Results

Evaluation of Sequential Pattern Search

Table 3 is a part of sequential patterns searched through sequential pattern mining.

Table 3. The Result of Sequential Pattern Mining

[Attack Type/Target Port/Protocol]	(Support)
1) 1/161/1 ⇒ 3/1080/6 ⇒ 4/161/17	(14 %)
2) 1/161/1 ⇒ 3/1080/6 ⇒ 3/8080/6 ⇒ 4/162/6 ⇒ 4/161/17	(10 %)

As analyzing the searched patterns as shown in the Table 3, for the pattern 1) in Table 3, the attacks type No. 1, No. 3 and No. 4 consisted of 14% in all data groups. With respect to these results, it is not possible to mention about the availability of searched patterns, but for the pattern 2) in Table 4, the attack No. 1, No. 3 and No. 4 were processed in order. As compared to the pattern 1) in Table 3, it is possible to find out the similarity of attack scenarios between two patterns. Multiple attacks by scenarios are more meaningful than a single attack on network. In addition, the attackers put meaningless behaviors in a scenario to hide their attack scenarios. Given

that the sequential pattern like the pattern 1) in Table 3 is an attack scenario, an attacker tries to attack with the scenario explained in the pattern 2) in Table 3 including the meaningless behaviors during attack to prevent the intrusion detection system from detecting attack.

The reliability of the efficiency of a searched pattern is based on support. For shorter sequences, the processing time in Frequent Episode is faster than that in PrefixSpan algorithm, but as sequences became longer, the efficiency of PrefixSpan algorithm is better.

Since Frequent Episode method compares the support as creating candidate items by phase and eliminates unnecessary parts and PrefixSpan creates all items satisfying the minimum support and searches the sequences in the next phase without creating candidate items, the processing time of PrefixSpan is longer than those in other algorithms for the short sequences, but PrefixSpan is more effective than other algorithms in terms of processing time required to create candidate items for the long sequences.

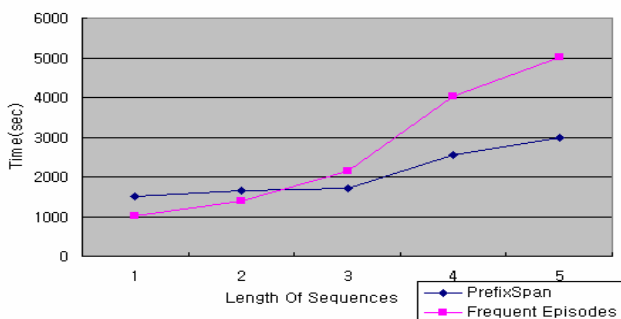


Fig. 3. The elapsed time of increasing sequence length

PrefixSpan algorithm demonstrated more efficient performance in the pattern mining for longer sequences. However, PrefixSpan has a limit because it requires more time than other methods for shorter sequence pattern mining. As a result, adding time constraint and setting appropriate thresholds are required to reduce costs and to solve the problem. Moreover, PrefixSpan depends on DBMS and Machine for saving data of the implemented programs. Then, the future works has to focus on data saving structures using trees such as indexing to increase the efficiency of performance.

Evaluation of Clustering Miner

The experiment was to define the cluster previous to each cluster generated, and to determine if the sequence of clusters could be generated based on the sequence.

In addition, the test dataset contained approximately 2,000,000 data instances, which were based on network traffic for two weeks. For the experiment, the real alert data were used as the input dataset, and the values derived from the experiment above were used as the user-defined variables. This experiment generated sequences of clusters by analyzing the distribution of previous alerts, which were the cause of the generation of the resulting sequences, and showed that it was possible to provide the

method of forecasting the future type of the alerts occurring by abstracting the sequences of clusters through integrating each sequences of clusters generated.

Here, we can find out potential alert sequences, which might mean attack scenario or strategies behind attack. It provides for the intrusion detection system to capture the high-level detection.

5 Conclusion

In this paper we propose alert correlation analysis by applying data mining in order to improve the performance of the intrusion detection systems.

The proposed alert correlation analysis were performed by using mining tasks such as axis-based association rule, axis-based frequent episodes, sequential pattern search and order-based clustering. We also implemented alert data mining framework to support alert analyzer for alert correlation analysis.

We have applied data mining methods for the alert correlation analysis. We improved not only new mining mechanisms in the specific domain of the alert correlation, but also proposed an alert analyzer that analyzes the alert data which were stored in RDBMS with mining system. The implemented mining system supports the alert analyzer and the high level analyzer efficiently for the security policy management. The alert analyzer supports the proceeding and making of security policy efficiently in the framework of Policy Based Network Management. It is a helpful system to manage the fault users and hosts.

The contribution of this paper is that we have adapted and extended the notions from data mining for the alert correlation. The approach has the ability to aggregate the alerts and to find out the alert sequences. We can predict possible attack sequences in the intrusion detection domain.

References

1. M.J. Lee, M.S. Shin, H. S. Moon, K. H. Ryu "Design and Implementation of Alert Analyzer with Data Mining Engine", in Proc. IDEAL'03, HongKong, March. 2003
2. E.H. Spafford and D. Zamboni., "Intrusion detection using autonomous agents", Computer Networks, pp. 34:547-570, 2000.
3. H. Debar and A.Wespi, "Aggregation and correlation of intrusion-detection alerts", In Recent Advances in Intrusion Detection, in LNCS, pp. 85 - 103, 2001.
4. A. Valdes and K. Skinner, "Probabilistic alert correlation", in Proc. The 4th International Symposium on Recent Advances in Intrusion Detection (RAID 2001), pp. 54-68, 2001.
5. Tcpdump/Libpcap, Network Packet Capture Program, <http://www.tcpdump.org>, 2003
6. P. Ning and Y. Cui., "An intrusion alert correlator based on prerequisites of intrusions", Technical Report TR-2002-01, Department of Computer Science, North Carolina State University
7. H. S. Moon, M.S. Shin, K. H. Ryu and J. O. Kim "Implementation of security policy server's alert analyzer", in Proc. of ICIS, Seoul, Aug. 2002

OSDM: Optimized Shape Distribution Method

Ashkan Sami¹, Ryoichi Nagatomi², Makoto Takahashi¹, and Takeshi Tokuyama³

¹Graduate School of Engineering, Tohoku University, Japan

²Department of Medicine and Science in Sports and Exercise, Graduate School of Medicine,
Tohoku University, Japan

³Graduate School of Information Sciences, Tohoku University, Japan
ashkan.sami@most.tohoku.ac.jp

Abstract. Comprehensibility is vital in results of medical data mining systems since doctors simply require it. Another important issue specific to some data sets, like Fitness, is their uniform distribution due to tile analysis that was performed on them. In this paper, we propose a novel data mining tool named OSDM (Optimized Shape Distribution Method) to give a comprehensive view of correlations of attributes in cases of uneven frequency distribution among different values of symptoms. We apply OSDM to explore the relationship of the Fitness data and symptoms in medical test dataset for which popular data mining methods fail to give an appropriate output to help doctors decisions. In our experiment, OSDM found several useful relationships.

1 Introduction

Human medical data are at once the most rewarding and difficult of all biological data to mine and analyze. Even though knowledge discovery from medical data is very rewarding in a lot of aspects, great challenges face the data miners who would like to work in this field. Since specific challenges of medical data have been discussed in [1], we will not go into detail of specific characteristics that medical knowledge discovery projects have. However as you will see in this paper, comprehensibility is a driving force in overall theme of this work.

Fitness tests are mainly designed for elderly people to find the status of their Fitness and risk to fall. These measures can also be used as indicators of overall 'healthiness' and physical activity of elderly people. What makes majority of these measurements different from regular tests is the way they are interpreted.

Interpretation of Fitness tests are based on tile analysis which is a quite popular in medical studies on regular people (in contrast to patients) [2] which is called social or community study in medical terms. In case of Blood analysis or Urine analysis, a clear threshold separating healthiness or suspect of illness exists. In contrast, for Fitness such clear thresholds are not given. Fitness tests are interpreted using quartile. To perform quartile analysis, first, for each variable we make the data set arranged in an ascending (or descending) order. Then, the 25th percentile, the median and 75th percentile are used to divide the data set into 4 groups, with each group containing one-fourth (25%) of the observations. Tertile (division based on three categories) is another popular method of interpretation.

In social studies, people with severe symptoms are much less than healthy people. Furthermore, due to tile analysis that was explained the overall distribution changes to uniform distribution. Table 1 and 2 are typical examples of these distributions.

Interrelationship of Fitness test and symptoms is very important for medical doctors in Sports Medicine. These relationships can lead them to design or suggest activities that can improve or stop the symptoms or at least slow the progress of the symptom.

Table 1. An example of strong relationship with OSDM’s values

Value	Quartile 1	Quartile 2	Quartile 3	Quartile 4	<i>OSDM_i</i> 's
0	73	65	53	43	-10.2544
1	32	30	27	25	-2.41421
2	1	5	5	4	1.825556
3	2	4	6	9	2.321973
4	1	4	9	10	3.360109
5	1	2	10	19	6.760814
Sum	110	110	110	110	

Table 2. An example of no relationship

Value	Quartile 1	Quartile 2	Quartile 3	Quartile 4	<i>OSDM_i</i> 's
0	59	54	50	50	-3.49574
1	30	32	32	33	0.972608
2	1	2	8	10	3.540057
3	5	5	7	2	-2.59904
4	2	9	5	11	4.051189
5	10	8	8	4	-2.04464
Sum	110	110	110	110	

The main purpose of this paper is to present a general method named Optimized Shape Distribution Measure (*OSDM*) to be used for finding relationships when of the variables of interest has uniform distribution due to quartile analysis and the distribution of values are highly skewed. *OSDM* consists of set of numerical values that each presents how an overall uniform distribution has been spread among a specific group. Each specific value of *OSDM* is a real number that its magnitude presents how far the distribution is tilted and sign of the measure presents which side. As an example a very large *OSDM* value compare to smaller values indicates a major shift of data toward higher values, where a smaller *OSDM* value presents frequencies that are slightly tilted toward lower tile values.

To have an appreciation of the method, we find relationships between, Fitness test with Lower Urinary Tract Symptoms (LUT). The use of regular data mining algorithms even though provide answers, the answers are not comprehensive in a way to convey knowledge concerning the data. Thus, in this paper, first we will talk about Fitness dataset. Section 3 presents the motivation. Section 4 presents *OSDM* and two indexes. Section 5 describes the experimental results and summary comes in section 6.

2 Fitness Dataset and Its Pre-processing

The file, Fitness dataset, contained results of different Fitness tests on 960 Japanese aged over 70 years. The data was collected through one of the largest Medical research projects named Tsurugaya Project, at Tohoku University; Sendai, JAPAN. We used this dataset based on two reasons. First, the data quality was much higher than regular datasets available due to various quality control routines that were performed on them. Secondly, results of examinations from other departments were also available so that interrelationship finding can be performed.

The Fitness Dataset which originally had 64 fields. Unrelated items and items related to illnesses and “pain conditions at the test date” were also deleted. Of course some items related to the condition of measurement may be used to compensate or to adjust the measured values.

The remaining data consisted of the results of tests that were performed on the volunteers. Due to space limitation, we will not go through details of the data, the main characteristics of the items in file are as follows:

1. The length that volunteers could reach in different direction without losing balance.
2. The time or number steps that it took them to perform a task, like walking 10m.

3 Motivation

Even though extensive number of well established statistical algorithms [3] exist that can find correlations between the two values, results of these tests will obey the major distributions that are concentrated around low values of the symptoms. The high value symptoms that have much less frequency and may contain a lot of knowledge will have no significant contribution to the results. Although in Statistics we do not consider small values, in data mining these values with small frequencies are regarded.

Decision tree analysis to find rules sharing both categories can be performed by deploying some objective measures like Quality of Life [4] as the consequent of the rules. However these analyses will lead to results that are very hard for experts to interpret if we use the data that is not tiled. In contrast, using tiles will lead to severe loss of accuracy.

Clustering of data in high dimension like [5] is hard to interpret and subspace clustering like Close Frequent Itemsets (CFI) [6], although provides results, no knowledge can be gained from it. Other methods of driving rules like association rule classifiers [7] may indeed lead to no answer.

4 Optimized Shape Distribution Measure and Its Indexes

OSDM is a method of knowledge discovery that by assigning *OSDM* values for each distribution of values of the symptom finds if an interrelationship exists between them. Table 1 and 2 present two sample sets and their *OSDM* values.

4.1 OSDM

By *OSDM* values, we can express how the shape of distribution is tilted. Basically we designed *OSDM* values so that irrespective of the frequencies of each symptom value and its shape the value resides in the range of real numbers. Large positive *OSDM* values show a complete tilt toward higher quartiles and negative numbers with high absolute value represent a complete tilt toward the left. It is called optimized since with one degree of freedom, it minimizes the square of shifted distance function.

To provide a clearer definition, consider *A* is a desired symptom and *B* is a result of a specific Fitness test after quartile. When there is no relationship between *A* and *B*, distribution of *B* with respect to each specific value of *A* should not vary in a consistent way or the shape of distribution may stay the same. In contrast, when a relationship exists, the variation of distribution should follow a pattern. To find the pattern, we should provide a quantitative way of explaining the distribution. Then based on the change in the measure (constantly increasing or decreasing), we can say a pattern exists. If no consistent change or no change at all exists, then no relationship between the two exists.

In other words, consider in general *A* can have values A_0, A_1, \dots, A_n and *B* is distributed among B_1, B_2, \dots, B_m . At the same time f_{ij} is the number of samples for a specific *i* and *j*, where $0 \leq i \leq n, 1 \leq j \leq m, A$ is A_i and *B* is B_j . In case of a relationship, the measure for distribution of f_{ij} 's for a specific value *i* when *i* changes from 0 to *n* should constantly increase or decrease.

First, we will simply assign 1 to *m* to B_1, B_2, \dots, B_m respectively. This assignment is performed so that contribution of each f_{ij} will be appointed to a specific B_j . This assignment can be done anyway that we want as long as *B*'s are strictly increasing integers and the larger assignments represent higher tiles that we are working with.

$OSDM_i$ values are derived by first defining shifted square distance through one degree of freedom by $OSDM_i$ values and minimizing the function to derive the values. $OSDM_i$ values are presented by:

$$OSDM_i = \frac{\sum_{j=1}^m f_{ij}^2 - \sum_{j=1}^m B_j^2 + \frac{1}{m} \left[\left(\sum_{j=1}^m B_j \right)^2 - \left(\sum_{j=1}^m f_{ij} \right)^2 \right]}{2 \left(\frac{1}{m} \sum_{j=1}^m B_j \sum_{j=1}^m f_{ij} - \sum_{j=1}^m B_j f_{ij} \right)} \pm \left(\sqrt{\frac{\left[\sum_{j=1}^m f_{ij}^2 - \sum_{j=1}^m B_j^2 + \frac{1}{m} \left[\left(\sum_{j=1}^m B_j \right)^2 - \left(\sum_{j=1}^m f_{ij} \right)^2 \right] \right]^2}{2 \left(\frac{1}{m} \sum_{j=1}^m B_j \sum_{j=1}^m f_{ij} - \sum_{j=1}^m B_j f_{ij} \right)}} + 1} \right) \tag{1}$$

As noticed $OSDM_i$ provides two sets of numerical values which is due to square nature of the function. It is very important to note that just one the values is acceptable. It can be shown for majority of cases a very good estimate of $OSDM_i$ is given by:

$$OSDM_i estimate = \frac{\sum_{j=1}^m \left(f_{ij} - \frac{1}{m} \sum_{j=1}^m f_{ij} \right) B_j}{\sum_{j=1}^m B_j^2 - \frac{1}{m} \left(\sum_{j=1}^m B_j \right)^2} \tag{2}$$

Definition of the terms and derivations are outlined in Appendix.

A pattern is nothing but a strict change through all the values of $OSDM_i$'s. In other words, we should obtain all the values of $OSDM_i$ for different i 's. If one of the below mentioned inequalities hold, a relationship exists.

$$OSDM_0 < OSDM_1 < \dots < OSDM_n \tag{3}$$

$$OSDM_0 > OSDM_1 > \dots > OSDM_n \tag{4}$$

Inequality (3) indicates a direct relationship. That is, by increase in i , the distribution shifts from lower values of B_j to higher values of B_j . As an example, in our case by increase in desired symptom values the distribution shifts to higher quartiles. Or as symptoms get worse, the indicated Fitness becomes worse. This kind of relation is of importance for Physical Activities that are measured in seconds. Thus by increase in those we see an improvement. Stated naively, for the cases like Max Speed that is measured in Seconds, any direct relationship with any desired symptom value will indicate that category of Fitness can improve the symptoms. Thus by assigning that type of exercise we can reduce the symptoms. For the measures that count steps, direct relationship does not provide any beneficial insight and it is only an observation.

In contrast, Inequality (4) presents a reverse relationship which by increase of i the distribution has a shift to lower values. These kinds of relationships, for the measures that count steps, will provide an indicator that exercises of that category will reduce the symptoms. The Fitness tests that are measured in seconds like TUGT will not give insight by this measure and is just an indication of status.

4.2 Indexes for OSDM

An index will present the strength of the measure. We present two indexes for $OSDM$ in cases the relationship is found. The measures are based on sum of square shifted distances. By substituting $OSDM_i$ value in Equation (8) and (7), we find R_i^2 for f_{ij} 's of each i . Thus, we define the following indexes:

$$I_1 = \sum_{i=0}^n R_i^2 / (n + 1) \tag{5}$$

$$I_2 = \frac{\sum_{i=0}^n \left(R_i^2 \times \sum_{j=1}^m f_{ij} \right)}{\sum_{i=0}^n \sum_{j=1}^m f_{ij}} \tag{6}$$

Where I_1 is the Index of overall fitness of *OSDM* and I_2 is, Index of overall error of *OSDM*. Definitely the higher the value of indexes the less concrete the analysis results.

5 Experimental Results

To evaluate our method we have tested our approach to find interrelationship between Fitness and one set of LUT related symptoms specifically benign prostatic hyperplasia (BPH). BPH is fundamentally a disease that causes morbidity through the urinary symptoms with which it is associated. IPSS[8] index system is used to evaluate status of BPH. The index includes seven questions covering frequency, nocturia, weak urinary stream, hesitancy, intermittence, incomplete emptying and urgency with answers ranging from zero to five, where zero means no symptom and five indicates severe problem with the urinary symptom.

In order to make the algorithm more robust, frequencies of IPSS values of 2 and 3 were grouped and considered as one category and frequencies of values equal to 4 and 5 were grouped and were considered as one category. Even after this aggregation, most of the summed categories were still much less than 10% of the population. It makes sense to consider more populated samples than solely rely on few numbers that existed in the high values of IPSS since these aggregates will dampen the huge effect of small variations in few data. It is important to emphasize that the proposed method is not dependant on the percentage of the distribution among values; however, the summation will make the analysis more robust.

In the male population, a strong relationship was between Incontinence and 10m walking test. Moreover, the algorithm showed males suffering from incontinence were walking more slowly at shorter steps than regular people. However, we found no major interrelationship between walking speed and nocturia. Even though a pattern between Urgency and Fitness was observed, the pattern was not interesting just descriptive. In other words, the people who suffered from Urgency showed they are faster walkers. A slight relationship between Functional reach category and hesitancy was observed. Since the relationship was not very strong it was not regarded highly. Weak stream had relationship with stretching, so it can be concluded exercises that improve dynamic body balance can improve stream problems. Frequency symptoms also become worse for people who are slow walkers. Emptying symptom affects Lateral extension and Max Speed.

The relation between incontinent and Fitness tests was very important for physicians. Since, urine incontinence or unintended leak of urine makes patients embarrassed, and occasionally results in home-bound without going out of home. This relationship shows, appropriate amount of exercise not only keeps them fit but also can prevent the embarrassment they encounter due to incontinent.

6 Summary

This paper presents *OSDM* (Optimized Shape Distribution Method) that can be deployed in situations where relationship between two variables, one having uniform

distribution due to tile analysis, is under investigation. This method especially is suitable when the distribution of the values with respect to each other is uneven. *OSDM* value quantitatively represents the distribution. We presented some experimental results based on LUT and Fitness tests and found very useful relationships. Moreover, the robustness of the measure was provided by the indexes.

References

1. Krzysztof J. Cios, and G. William Moore, 'Uniqueness of medical data mining', *Artificial Intelligence in Medicine* 26(1-2), (2002) 1–24.
2. StatSoft, Inc. *Electronic Statistics Textbook*. Tulsa, OK: StatSoft. WEB: <http://www.statsoft.com/textbook/stathome.html>. (2006).
3. American Society of Consultant Pharmacologists, *Health Trends*, <http://www.ascp.com/public/pubs/tcp/1998/jun/trends.shtml>, (1998).
4. Okamura K, Usami T, Nagahama K, Maruyama S., Mizuta E., "Quality of life" assessment of urination in elderly Japanese men and women with some medical problems using International Prostate Symptom Score and King's Health Questionnaire', *European Urology*, Apr;41(4), (2002) 411-9.
5. Agrawal, R. Gehrke, J., et al, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," *SIGMOD RECORD*, VOL 27; NUMBER 2, (1998) 94-105.
6. J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation.", *Proc. 2000 ACM SIGMOD Int. Conf. Management of Data (SIGMOD'00)*, Dallas, TX, May (2000) 1-12.
7. 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.
8. Barry, M. J. , Fowler, F. J. Jr, et al 'The American Urological Association symptom index for benign prostatic hyperplasia,' *Journal of Urology*, Nov;148(5), (1992) 1549-57.

Appendix

Shift square distance based on $OSDM_i$ value and one degree of freedom is given by

$$R_i^2 = \sum_{j=1}^m \left[f_{ij} - (C_i + OSDM_i \times B_j) \right]^2 \times g(OSDM_i)^2 \quad (7)$$

To minimize the distance function , $g(OSDM)$ is given as:

$$g(OSDM_i) = \frac{1}{\sqrt{1 + OSDM_i^2}} \quad (8)$$

To minimize R_i^2 , we should have the followings:

$$\frac{\partial R_i^2}{\partial C_i} = 0 \quad (9)$$

$$\frac{\partial R_i^2}{\partial OSDM_i} = 0 \quad (10)$$

Equation 9 leads to definition of C_i .

$$C_i = \frac{\sum_{i=1}^m f_{ij} - OSDM_i \times \sum_{i=1}^m B_j}{m} = \mu_{f_i} - OSDM_i \times \mu_B \tag{11}$$

Where Equation (10) leads to:

$$-OSDM_i \sum_{i=1}^m B_j^2 + (1 + OSDM_i^2) \sum_{i=1}^m B_j f_{ij} + OSDM_i \sum_{i=1}^m f_{ij}^2 + C_i (OSDM_i^2 - 1) \sum_{i=1}^m B_j - 2C_i OSDM_i \sum_{i=1}^m f_{ij} + m OSDM_i C_i^2 = 0 \tag{12}$$

Substituting Equation (11) into (12) is presented as Equation (13).

$$-OSDM_i \sum_{i=1}^m B_j^2 + (1 + OSDM_i^2) \sum_{i=1}^m B_j f_{ij} + OSDM_i \sum_{i=1}^m f_{ij}^2 + \frac{1}{m} (OSDM_i^2 - 1) \sum_{i=1}^m B_j \times \left(\sum_{i=1}^m f_{ij} - OSDM_i \times \sum_{i=1}^m B_j \right) - 2OSDM_i \left(\sum_{i=1}^m f_{ij} - OSDM_i \times \sum_{i=1}^m B_j \right) \sum_{i=1}^m f_{ij} + m OSDM_i \left(\sum_{i=1}^m f_{ij} - OSDM_i \times \sum_{i=1}^m B_j \right)^2 = 0 \tag{13}$$

Which after quite a bit of algebra leads to:

$$OSDM_i^2 \left(\frac{1}{m} \sum_{i=1}^m f_{ij} \sum_{i=1}^m B_j - \sum_{i=1}^m B_j f_{ij} \right) + OSDM_i \left(\sum_{i=1}^m f_{ij}^2 - \sum_{i=1}^m B_j^2 + \frac{1}{m} \left(\left[\sum_{i=1}^m B_j \right]^2 - \left[\sum_{i=1}^m f_{ij} \right]^2 \right) \right) \sum_{i=1}^m B_j f_{ij} - \frac{1}{m} \sum_{i=1}^m f_{ij} \sum_{i=1}^m B_j = 0 \tag{14}$$

$OSDM_i$ is the solution of this equation, which is presented as Equation (1).

View-Angle of Spatial Data Mining

Shuliang Wang¹ and Haning Yuan²

¹ International School of Software, Wuhan University, Wuhan 430072, China
slwang2005@whu.edu.cn

² School of Economics, Wuhan University of Technology, Wuhan 430072, China
yhn1979yhn@163.com

Abstract. In order to discover the knowledge with various granularities from amounts of spatial data, a view-angle of spatial data mining is proposed. First, the view-angle of spatial data mining is defined. In its context, the essentials of spatial data mining are further developed. And the view-angle based algorithms are also presented. Second, the view-angles of Baota landslide-monitoring data mining, and their pan-hierarchical relationships, are given. Finally, view-angle III is taken as a case study to discover quantitative, qualitative and visualized knowledge from Baota landslide-monitoring databases. The results indicate that the view-angle based data mining is practical, and the discovered knowledge with various granularities may satisfy spatial decision-making at different hierarchies.

1 Introduction

Under special problems and circumstance, the million spatial data is a kind of disorderly and unsystematic nature accumulation, the data resource which has some modes like relation quality, order quality, rule quality and so on. Spatial data mining is to extract previously unknown, potentially useful, and ultimately understood rules from spatial data (Ester et al., 2000). And it is further a process of discovering a form of rules plus exceptions at hierarchal perspectives with various thresholds by using some theories and techniques under different backgrounds (Wang, 2002; Wang et al., 2004).

It has been acknowledged that human beings could observe and analyze the same entity from very different perspectives. And these observations are on various cognitive levels of granularity. The discovered knowledge is associations with spatial objects at the cognitive hierarchy, and they may be description and prediction, for example, association rule, clustering rule, classification rule, characteristics rule, serial rule, predictive rule, and outlier. As a computerized simulation of human cognition, spatial data mining discovers the patterns not only in a granularity world, but also among various granularity worlds. When dealing with different granularity worlds, the discovering manipulations of generalization and summarization are often soft computing with discrete linguistic terms instead of continuous data (Han, Kamber, 2001).

As to the same set of spatial data, different people may mine them by using different theories and techniques. In these contexts, different people may get different results, and different methods may also lead to different results. The different

knowledge from different perspective may have different applicable usage under different backgrounds. For instance, seen from the same monitoring databases, a landslide firstly shows data mining continuously observed data instead of discrete symbolic parameters or qualitative concept. A macrocosmic decision-making to guide the final direction may ask for the most generalized knowledge for the whole landslide, e.g., a sentence, even a word. Simultaneously, a mid- cosmic decision-making, who is a connecting link between the preceding and the following, may demand the common knowledge for each breaking-sector of the landslide. And a microcosmic decision-making to monitor the exact deformation rules may desire the detailed knowledge of each monitoring points. The abovementioned is caused by the discovery view-angles at different hierarchal levels.

In this paper, the view-angle of spatial data mining is proposed to be an alternative to study the essential virtue on how to discover the different knowledge from the same dataset at different cognitive perspectives. How do we describe and implement these differences in the view-angle of spatial data mining? In the view of applications, when experts and information technique personnel summarize and narrate the knowledge and rules which are exteriorly input to the system and become the repository, traditional methods often come up against serious difficulties because of the complexity and the illegibility and the difficulty to express of knowledge. In the following, section 2 describes the concept of the view-angle of spatial data mining, paying more attention to the mechanism and algorithms of spatial data mining based on view-angles. Section 3 presents the view-angles of landslide-monitoring data mining. Section 4 gives a case study on Baota landslide-monitoring data mining in the context of view-angle three. Finally we come to the conclusions in section 5.

2 View-Angle of Spatial Data Mining

The view-angle of spatial data mining is an angle to discover the knowledge from spatial databases when the same people with different background, or different people with the same background, mine the same spatial dataset at different cognitive perspectives under the given objective granularity. The view-angle is hierarchal, which may provide multiple-knowledge for decision-making when people study, resolve and interpret the natural, human and social problems at different cognitive levels.

2.1 Levels of View-Angle

Human thoughts have different levels which rest with cognitive stature. Decision-making persons in different levels and under different knowledge backgrounds may need different spatial knowledge. Simultaneously, if we mine the spatial data from dissimilar view-angles, there may be some knowledge of different levels. Spatial data mining is a knowledge discovering process about spatial decision-making which is based upon the same set of data, a dealing which cannot predict the data in the computer spatial database from the functions of the most basal silicon CMOS chip, the same as the analysis of qualities of the single ion and nerve fiber cannot deduce the cognition and thinking of human brain. Therefore, the view-angle of spatial data

mining based on multiple and different levels mines the related, multi-granularity and multi-category spatial knowledge modes which are concealed in the data to concentrate the function of each ration data from different acquaintanceship to varies granularity decision thinking in order to fit the variety of users' requirements or application aims.

2.2 Granularities of View-Angle of Spatial Data Mining

The process that human cognize the spatial data is a process of macrocosmic, mid-cosmic and microcosmic discovering knowledge for the relationship of complex objects, and also the microcosmic data in the objected located characteristic space express structure characteristics under the non-linearity reciprocity by using the abstracted concepts in nature language.

The granularity mainly expresses human's cognitive levels, reflects the precision of interior details of the spatial data mining, and describes the spatial data mining geometrical transforming course of multi-scale, multi-pixels, or from thin to coarse (Fig.1).

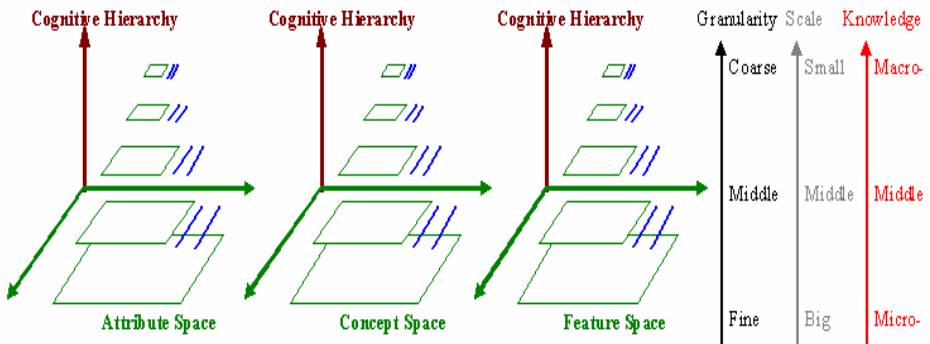


Fig. 1. Knowledge discovery with changeable cognitive levels

The achievement of spatial data mining at different cognitive levels is to observe and analyze spatial data with view-angles of different granularities, examine the combination of the same dataset at different observing distances and attain the spatial knowledge based on different knowledge background. Observing data with fine granularities is to compress the lens and reduce the perspective distance, which utilizes the accurate data mining algorithm to uncover the complicated external phenomenon and distinguish distinctions better and truly for commonly gaining individual knowledge. Contrarily, analyzing data with coarse granularities is to elongate the lens and increase the perspective distance, which utilizes the smooth data mining algorithm to ignore hairlike distinctions and seek the commonness for commonly gaining common knowledge. Microcosmic levels will become macrocosmic levels when levels of concepts ascend, and the knowledge cyclostyle will go up to knowledge hierarchy of higher abstract levels.

2.3 Mechanism of View-Angle

The implementation of spatial data mining in different cognitive levels is to observe spatial data from different distances, to analyze the same block of dataset or the group of several datasets from different perspectives, and to discover different knowledge from the spatial databases under different background. In the context of different view-angles, spatial data mining may gain a variety of knowledge with different hierarchies. And they have their own characteristics and applications. For instance, the knowledge discovered from landslide-monitoring database with the macrocosmic view-angle, mid-cosmic view-angle, and microcosmic view-angle, may separately satisfy the demands from the top decision-makers, the middle decision-makers, and the bottom decision-makers.

Human thinking is hierarchical, and the background knowledge may decide the levels of human cognition. Under the umbrella of these hierarchical characteristics, the virtue of spatial data mining may be taken as the decision-making on the basis of the same pile of spatial data. The decision-makers at different levels, who might have different backgrounds on theories, techniques, experience and so on, will demand different knowledge from the same data. That is, they want different view-angles of the discovered knowledge. With the fine view-angle, the observed distance to spatial data decreases to permeate the numerous and complicated surface phenomena in order to truly distinguish the difference among the data, which may often discover the individual knowledge. However, with the coarse view-angle, the observed distance to spatial data increase to lose sight of the hairlike difference in order to well find out the shared rules among the data, which may often discover the common knowledge.

The decision-making is a process from the theory to the practice, i.e. a hierarchical decision from knowledge to data. On the contrary, spatial data mining is a process from the practice to the theory, i.e. a cognitive discovery from data to knowledge. In the context of the view-angles, spatial data mining is therefore able to concentrate the power of each quantitative data into the qualitative thinking of decision-making at different cognitive hierarchies.

2.4 Mining Algorithms of View-Angle

Input: Spatial dataset

Output: Spatial Knowledge, view-angle types

Process:

- (1) Choose the view-angle of spatial data mining under the given demands;
- (2) Mine the given dataset with suitable theories or techniques;
- (3) Interpret the discovered knowledge under the chosen view-angle;
- (4) Evaluate the satisfactory degree of the discovered knowledge;
- (5) Change the view-angle, repeat step (1)-(4) until the discovered knowledge match the given demands.

Hence, on the basis of hierarchal view-angles, people can mine data sets in the world of one view-angle, and further among the different worlds of different view-angles simultaneously, even agilely jump back and forth among different worlds of

different view-angles. The roll-up is carried out hierarchy-by-hierarchy, and the linguistic atoms also become linguistic term, further concept. The higher the roll-up, the more generalized the qualitative concept. The concept that can attract the interesting, match the demand, and support the decision-making, will be the knowledge. On the contrary, the lower the drill-down, the more detailed the knowledge. That is to say, the top hierarchy of spatial data mining is the most generalized knowledge, while the bottom hierarchy of spatial data mining is the objective data in the spatial database.

3 View-Angles of Landslide-Monitoring Data Mining

A landslide may lead to great hazards if it moves too much. In order to avoid the landslide hazards, people have been monitoring the displacements of some landslides. With the continuous increase and accumulation, the huge amounts of the monitoring-data have far exceeded human ability to completely interpret and use. So landslide-monitoring data mining is a typical and necessary sample of spatial data mining.

The deformation of landslide is a comprehensive result that is caused by many factors, and its outer appearance is the observed data when the landslide is being monitored. Thus, it is necessary to look for a suitable technique to analyze the monitoring dataset from different perspectives, for example, view-angle of spatial data mining. And all the attributes are numerical displacements, i.e. dx , dy , and dh .

$$dx = x_{i+1} - x_0, \quad dy = y_{i+1} - y_0, \quad dh = h_{i+1} - h_0 \quad (1)$$

where, 0 denotes the first observed date, and i denotes the i th observed date. Respectively, the properties of dx , dy , and dh , are the measurements of displacements in X direction, Y direction and H direction of the landslide-monitoring points.

3.1 View-Angles

In landslide-monitoring data mining, the background knowledge is the monitoring-data on the stability of landslide. There are three basic standpoints when observing the dataset, and they include the point, date, and direction for monitoring landslide deformation. Each basic standpoint has two values, same and different.

- Point-Set: {same point, different point};
- Date-Set: {same date, different date};
- Direction-Set: {same moving-direction, different moving-direction}.

The different combination of these three basic standpoints may produce all the view-angles of landslide-monitoring data mining. And the number of view-angles is

$$C_2^1 \cdot C_2^1 \cdot C_2^1 = 8 \quad (2)$$

Each view-angle has different perspective meaning when mining the dataset.

- View-angle I: same point, same date, and same moving-direction;
- View-angle II: same point, same date, and different moving-direction;
- View-angle III: same point, different date, and same moving-direction;

- View-angle IV: same point, different date, and different moving-direction;
- View-angle V: different point, same date, and same moving-direction;
- View-angle VI: different point, same date, and different moving-direction;
- View-angle VII: different point, different date, and same moving-direction;
- View-angle VIII: different point, different date, and different moving-direction.

3.2 Pan-Hierarchical Relationship

All the view-angles show a pan-hierarchical relationship, which may be depicted in Fig.2 by using cloud model (Wang, 2002).

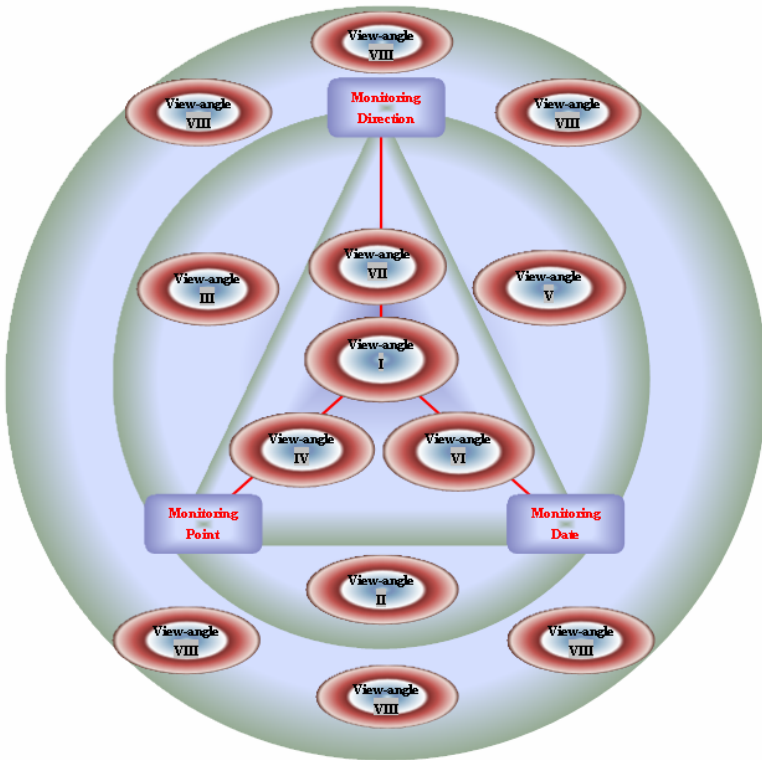


Fig. 2. View-angles of landslide-monitoring data mining and their pan-hierarchical relationship

In Fig.2, from the central View-angle I to the outer View-angle VIII, the observed distance becomes farther and farther while the mining granularity gets bigger and bigger. On the contrary, from the outer View-angle VIII to the central View-angle I, the observed distance becomes closer and closer while the mining granularity gets bigger and bigger. In details, View-angle I, View-angle II, View-angle III, and View-angle IV focus on the different attributes of the same monitoring-point, and they may discover the individual knowledge of the landslide in a conceptual space. Moreover, View-angle V, View-angle VI, View-angle VII, and View-angle VIII pay

attention to the different characteristics of multiple monitoring-points, and they may discover the common knowledge of the landslide in a characteristics space.

3.3 Basic View-Angle

Among the eight view-angles, View-angle I is the basic view-angle because other seven view-angles may be derived from it when one, two, or three basic standpoints of View-angle I change. While the landslide-monitoring data are mined in the context of View-angle I, the objective is a single datum of an individual monitoring-point in a given date. If three moving-directions dx , dy , dh are taken as a tuple (dx, dy, dh) , then View-angle IV, View-angle VI, and View-angle VIII may be also derived from View-angle II when monitoring-point, or (and) monitoring-date change. Thus, View-angle II is called a basic composed view-angle. At this time, the objective is on the total displacement of an individual monitoring-point in a given date. In the visual field of the basic view-angle or basic composed view-angle, the monitoring data are single isolated data instead of piles of data. In spatial data mining, they are only the fundamental unit or composed unit but not the final destination. So the practical view-angles are the remained six ones, i.e. View-angle III, View-angle IV, View-angle V, View-angle VI, View-angle VII, and View-angle VIII. With cloud model and data field (Wang, 2002), Table 1 and Table 2 describe the examples of the basic view-angle and basic composed view-angle.

Table 1. Basic view- angle

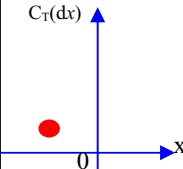

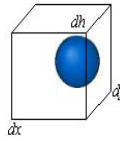
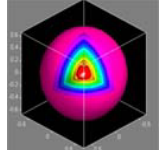
View-angle I	Numerical characters	“T=landslide stability” cloud model	Data field
Monitoring-point: BT21 Monitoring-date: 1997.06 Displacement in X direction: dx	$Ex=-3$ $En=0$ $He=0$		

Table 2. Basic composed view-angle

View-angle II	Numerical character			“T=landslide stability” cloud	Data field	
Monitoring-point : BT21 Monitoring-date : 1997-01	Ex	En	He			
Different displacement direction	dx	-3	0			0
	dy	4	0			0
	dh	-2	0			0
Total displacement	d	5.4	0	0		

4 Baota Landslide

In this section, Baota landslide-monitoring data mining is studied as a case to apply the view-angle of spatial data mining. Baota landslide locates in Yunyang,

Chongqing, China, and the region of Three Gorge on Yangtze River. And the landslide was monitored from June 1997. Wang(1997), Zeng(2000), and Jiang and Zhang (2002) had studied these monitoring-data, the results of which might be used to be comparison. Up to now, this database on the deformation has amounted to 1 Gigabytes. Because Yangtze River flows in the direction of west-east, the moving direction of Baota landslide is geological north-south. And the displacements of landslide mainly appear in X-direction (north-south), which matches the conditions of View-angle III.

In the eyes of View-angle III, the monitoring-data is a multi-dimensional displacement vector $(dx_1, dx_2, \dots, dx_m)$ for the same monitoring-point on different monitoring-date in the same moving X-direction. The basic unit is the atomic data in the eyes of View-angle I. Among the theories and techniques of spatial data mining (Li et al, 2002), cloud model is more suitable to mine Baota landslide-monitoring dataset under the umbrella of View-angle VIII. Based on cloud model, the large amounts of consecutive data be replaced by discrete linguistic terms, and the efficiency of spatial data mining can therefore be improved. Further, let the $|dx|$ -axis, $|dy|$ -axis respectively depict the absolute displacement values of the landslide-monitoring points. The certainty of the cloud drop $(dx_i, C_T(dx_i))$, $C_T(dx_i)$ is defined as Equation (3).

$$C_T(dx_i) = \frac{dx_i - \min(dx)}{\max(dx) - \min(dx)} \quad (3)$$

where, $\max(dx)$ and $\min(dx)$ are the maximum and minimum of $dx = \{dx_1, dx_2, \dots, dx_i, \dots, dx_n\}$. Then the rules on Baota landslide-monitoring in X direction can be discovered from the databases in the conceptual space.

Derived from section 2.2 "Mining algorithms of view-angle", the algorithms may be the following.

Input: Baota landslide-monitoring dataset

Output: Baota landslide-monitoring knowledge, View-angle VIII

Process:

- (1) Discover the three numerical characters {Ex, En, He} from the Baota landslide-monitoring dataset with backward cloud generator;
- (2) Interpret {Ex, En, He} qualitatively on the basis of some annotation rules;
- (3) Produce the visual knowledge cloud with forward cloud generator;
- (4) Evaluate the satisfactory degree of the discovered knowledge.

Where, Expected value (Ex), Entropy (En), and Hyper-Entropy (He) respectively depict the displacements levels, the scattering levels, and the stabilities of the displacements. According to the landslide-monitoring characteristics, we may let the linguistic concepts of "smaller (0~9mm), small (9~18mm), big(18~27mm), bigger(27~36mm), very big(36~50mm), extremely big(> 50mm)" with Ex, "lower (0~9), low(9~18), high(18~27), higher(27~36), very high(36~50), extremely big(>50)" with En, "more

stable (0~9), stable (9~18), instable(18~27), more instable (27~36), very instable (36~50), extremely instable (>50)” with He. With the above algorithms, the rules on Baota landslide-monitoring in X direction can be discovered from the databases in the conceptual space (Table 3, Fig.3).

Table 3. Quantitative and qualitative knowledge under View-angle VIII Monitoring

Monitoring Points	Numerical characters			Rules on the displacements
	Ex	En	He	
BT11	-25	18.1	19	big south, high scattered and instable.
BT12	-22.1	19.4	41.7	big south, high scattered and very instable.
BT13	-9.3	8.8	8	small south, lower scattered and more stable.
BT14	-0.3	3.7	6.7	smaller south, lower scattered and more stable.
BT21	-92.8	66.4	145.8	extremely big south, extremely high scattered and extremely instable.
BT22	-27	20.8	21.1	bigger south, high scattered and instable.
BT23	-26.5	21.6	53	big south, high scattered and extremely instable.
BT24	-20.5	20.2	27.4	big south, high scattered and more instable.
BT31	-40.3	28.4	92.2	very big south, higher scattered and very instable.
BT32	-22.9	18.7	38.2	big south, low scattered and more instable.
BT33	-25	22.2	26.4	big south, high scattered and very instable.
BT34	-20.9	20.7	32.8	big south, high scattered and more instable.

Fig. 3 visualizes the displacing rule of each point with 30,000 pieces of cloud-drops, where the symbol of “+” is the original position of monitoring point, the different rules are represented via the different pieces of cloud, and the level of color in each piece of cloud denotes the discovered rules of a monitoring-point. Fig.3 indicates that all monitoring points move to the direction of Yangtze River, i.e. south, or the negative axle of X. Moreover, the displacements are different from each other. BT21 are extremely big south, extremely high scattered and extremely instable, and followed by BT31. At least, BT14 is smaller south, lower scattered and more stable. In a word, the displacements of the back part of Baota landslide are bigger than those of the front part in respect of Yangtze River, and the biggest exceptions are BT21. So these may be taken as the microcosmic knowledge.

Fig. 4 shows the deformation standard of collective displacement of the landslide on three monitoring cross-sections of Baota landslide. Compared with every two datasets of three monitoring cross-sections, the level of the displacement, the degree of scattering deformation and the level of the monitoring of cross-section 2 are the largest, highest and most instable. The displacement of three cross-sections in a direction(X direction, Y direction or H direction) resetting to the displacement of cross-section 1, cross-section 2 and cross-section 3 in three directions (X direction, Y direction or H direction) can attain new sense. Thus, all monitoring points of three cross-sections have different degree of displacement. The rule of their range of displacement, scattering degree of displace and levels of monitoring is X direction>Y direction > H direction, and in H direction and Y direction: cross-section 2>cross-section 1>cross-section 3. Hereinto, the monitoring point BT21 of the cross-section 2

in X direction and Y direction changes most greatly about the degree of the displacement, scattering degree of displacement and levels of monitoring.

Fig.5 is the most generalized result at a much higher hierarchy than that of Fig. 3 in the character space, i.e. the displacement rule of the whole landslide. It is “the whole displacement of Baota landslide are bigger south (to Yangtze River), higher scattered and extremely instable”. So they can be taken as the macrocosmic knowledge.

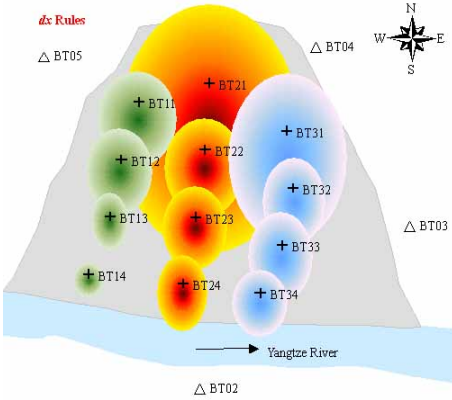


Fig. 3. Microcosmic knowledge

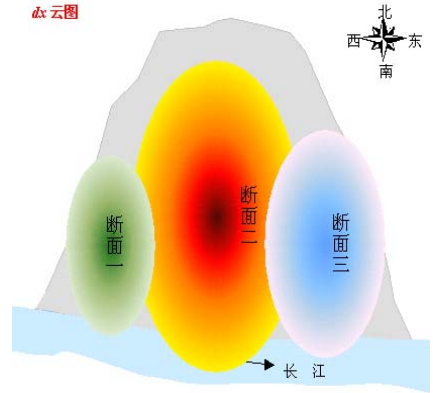


Fig. 4. View angle rule of cross-section

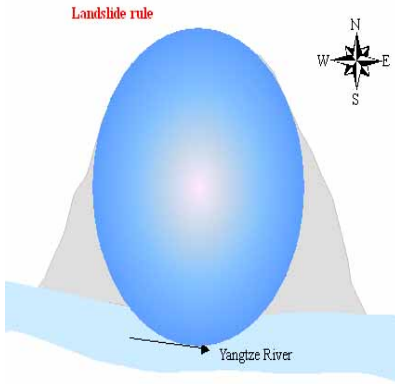


Fig. 5. Macrocosmic knowledge

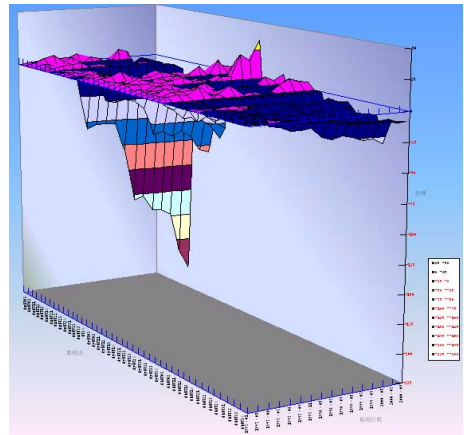


Fig. 6. Exception of Landslide view angle

Let us make a comprehensive view of Fig. 3, Fig. 4, Fig. 5 and Fig. 6, we could make a conclusion which helps the subliming of the spatial knowledge granularity along with the acquaintanceship level - that is “Baota landslide moves runs south by west (Yangzi River)”, along with a small quantity of sinking, and also the displacement of back edge is longer than the one of front edge, but the point BT21 is an exception. So far this is the most comprehensive conclusion of the Baota landslide

inspected data. Moreover it is also a piece of the most concentrated spatial knowledge described by conceptual language which is close to the human thinking and could be used in decision making directly. This “rule + exception” in Baota landslide spatial knowledge based on the inspected data, has been obtained by different level mining. It explains further that the Baota land slider moves south in horizon and sinking down in perpendicularity. The horizontal displacement is not coherence and fluctuates irregularly which could be explained that the horizontal deformation of the most inspected points of Baota landslide are similar, mainly moving to the direction of Yangzi River. It is the press landslide. The space around point BT21 is the high incidence area of small-scale landslide disaster. Actually, the terrene of Baota landslide is perpendicularity and the obliquity is top precipitous like chairs. These landslide characteristics are totally the same as the above knowledge which explain that the internal forces including the land slider's material character, geological structure and gradient are the main causes of formation of landslide disaster. Contemporarily, we could find out in the patrol process of landslide area, the nature realism in the area and the spatial knowledge gained from the view angle mining is very similar. So the research on spatial data mining view angle is necessary, practical and realistic. Furthermore, relative to the fake distributing Fig.[4] of displacement absolute value, the three data characteristics in the cloud model reserve the direction of the displacement.

When the Committee of Yangtze River (Zeng, 2000) investigated in the region of Yunyang Baota landslide, they found out that the landslide had moved to Yangtze River. Nearby the landslide-monitoring point BT21, a small landslide hazard had taken place. Now there are still two pieces of big rift. Especially, the wall rift of the farmer G. Q. Zhang's house is nearly 15 millimeters. These results from the facts match the discovered spatial knowledge very much, which indicates that the techniques of view-angles of spatial data mining are practical and creditable. Furthermore, besides they are closer to human thinking in decision-making support, the knowledge discovered from Baota landslide-monitoring dataset is also consistent with the results in the references (Wang, 1997; Zeng, 2000; Jiang, Zhang, 2002).

5 Conclusions

This paper proposed the view-angle of spatial data mining. Under the umbrella of View-angle III, Baota Landslide monitoring dataset were mined. The results indicate that the view-angle could reduce the task complexity, improve the implementation efficiency, and enhance the comprehension of the discovered knowledge. And the discovered knowledge not only included multiple thinking worlds, but also matched the practical phenomena of Baota lanslide very much. If more than one view-angle is applied, the discovered knowledge may be more practical and richer.

Acknowledgement

This study is supported by 973 Program (2006CB701305), the State Key Laboratory of Software Engineering Fund (WSKLSE05-15), and the Ministry Key GIS Lab Fund.

References

1. Ester, M. et al.: Spatial data mining: databases primitives, algorithms and efficient DBMS support. *Data Mining and Knowledge Discovery*, 4 (2000), 193-216
2. Han J., Kamber M.: *Data Mining: Concepts and Techniques*. Academic Press, San Francisco (2001)
3. Jiang Z., Zhang Z.L.: Model recognition of landslide deformation. *Geomatics and Information Science of Wuhan University*, 27 (2002), 127-132
4. Li D Y: Knowledge representation in KDD based on linguistic atoms. *Journal of Computer Science and Technology*, 12(1997): 481-496
5. Li D.R., et al.: Theories and technologies of spatial data mining and knowledge discovery. *Geomatics and Information Science of Wuhan University*, 27(2002): 221-233
6. Miller H J, Han J (eds.): *Geographic Data Mining and Knowledge Discovery*, Taylor and Francis, London (2001)
7. Wang S.L., Data field and cloud model based spatial data mining and knowledge discovery. Ph.D. Thesis, Wuhan University, Wuhan (2002)
8. Wang S.L. et al.: A try for handling uncertainties in spatial data mining, *Lecture Notes in Computer Science*, Vol. 3215 (2004) 513-520
9. WANG Sangqing: *Landslide Monitor and Forecast on the Three Gorges of Yangtze River*. Earthquake Press, Beijing (1999)
10. Zeng X.P.: *Research on GPS Application to Landslide Monitoring and its Data Processing*, Dissertation Master, Wuhan University, Wuhan (2000)

Maintaining Moving Sums over Data Streams

Tzu-Chiang Wu¹ and Arbee L.P. Chen^{2,*}

¹ Institute of Information Systems and Applications, National Tsing Hua University,
Hsinchu, Taiwan 300, R.O.C.

mr926712@cs.nthu.edu.tw

² Department of Computer Science, National Chengchi University,
Taipei, Taiwan 116, R.O.C.

alpchen@cs.nccu.edu.tw

Abstract. Given a data stream of numerical data elements generated from multiple sources, we consider the problem of maintaining the sum of the elements for each data source over a sliding window of the data stream. The difficulties of the problem come from two parts. One is the number of data sources and the other is the number of elements in the sliding window. For massive data sources, we need a significant number of counters to maintain the sum for each data source, while for a large number of data elements in the sliding window, we need a huge space to keep all of them. We propose two methods, which shares the counters efficiently and merge the data elements systematically so that we are able to estimate the sums using a concise data structure. Two parameters, ϵ and δ , are needed to construct the data structure. ϵ controls the bounds of the estimate and δ represents the confidence level that the estimate is within the bounds. The estimates of both methods are proven to be bounded within a factor of ϵ at $1-\delta$ probability.

1 Introduction

The problems over data streams have been studied for years [1,2,3,4,5,6,7,8]. In these studies, it is usually assumed data streams are unbounded but there is only limited space to store data. Also, the data input rate is assumed to be so fast that the processing time for each data element is very short. Under these constraints, the proposed methods primarily focus on how to keep necessary information in the limited space so that we can estimate the statistics that we are interested.

Three data models are studied under the data stream environment, i.e., landmark models, sliding window models and damped window models [4,7]. A landmark model considers the data in the data stream from the beginning until now. A sliding window model, on the other hand, considers the data from now up to a certain range in the past. A damped window model associates weights with the data in the stream, and gives higher weights to recent data than those in the past.

Different data stream applications have different data stream environments. We consider applications in which the statistics of the elements in a sliding window over a data stream of multiple sources need to be estimated. In the following, we present two examples as the motivations of our work.

* Contact author.

In [7], the authors mentioned that the NEXRAD Doppler data are continuously generated by 158 radar systems. Because these meteorological data, especially the latest data, are crucial to the prediction of weather, it is desirable to have a summary of the current data. Suppose we are interested in the average temperature over the past 24 hours in an area covered by one of the radar systems. If the data over the period from 158 radar systems can be summarized using a concise data structure and maintained continuously, such kind of queries can be answered quickly.

In [8], a system that monitors thousands of time series in real time was proposed. One of its functions is to continuously report closest time series in a sliding window using Euclidean distance. That is, for time series i and j , each having k data elements, $e_{i,m}$ and $e_{j,m}$, $1 \leq m \leq k$, respectively in a sliding window, the Euclidean distance of the two time series is $(\sum_{1 \leq m \leq k} (e_{i,m} - e_{j,m})^2)^{1/2}$. Thus, to find the closest pair among n time series, we have to find the smallest distance among $n*(n-1)/2$ series pairs. For thousands of time series, that is a huge amount of work.

Both applications can be generalized into a problem of maintaining the sums of a data stream with multiple data sources in a sliding window. As Figure 1 illustrated, it is desirable to have a concise summarization data structure to maintain the sums in the sliding window. This is the problem we are going to study in this paper, and we define our problem as follows:

Given a data stream, data in the stream are presented as (source, value), where the source is the index of the m data sources and the value is a positive integer, bounded by some integer R . Maintain at every time instant, the sum of the values for each data source in a sliding window of W time units. W is assumed to be positive integers and this sum is called a moving sum.

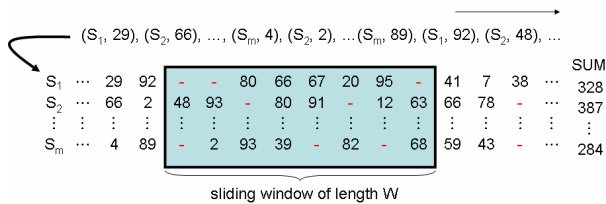


Fig. 1. The data stream environment of moving sums

Our approaches are based on a recent work of the data streams, i.e., the Count-Min(CM) sketch [1]. A CM sketch assumes a landmark data model over the same data stream environment as ours, i.e., a single data stream with multiple data sources. It is designed to estimate the sum of all the elements for each data source in the data stream. To construct a CM sketch data structure, a user has to supply two parameters, δ and ϵ , with which the estimates of the CM sketch can be bounded within a factor of ϵ at $1 - \delta$ probability. That is, a CM sketch is a randomized data structure. It means that there is a random procedure built into the data structure, so two instances of the CM sketch, constructed with the same δ and ϵ , under the same data stream can generate different estimates for one data source. However, through a carefully designed random procedure, we are sure that, on average, the estimate is bounded within the range controlled by the parameter, ϵ . Having the upper and lower bounds of an

expected estimate, by the Markov inequality, we can determine how large our data structure should be so that we are $1 - \delta$ percent sure that an estimate is within the bounds of its expected estimate. This is the basic idea of the CM sketch.

Thus, with a CM sketch, we “share” the counters among the data sources, and reduce the number of counters we have to maintain. Having a counter like this, to estimate the sum of a data source in a sliding window, all we need to do is to subtract the value of the counter at the start of the sliding window from the current value in that counter. This leads to our first method. However, the estimation of our first method is based on the data in the sliding window as well as the data out of the window. The estimation error gets larger as more and more data moving out of the window. That is, the bounds of the estimate grow as time proceeds. But, we want our method to be dependent on the data elements in the sliding window, not on the data elements out of the window.

A naïve solution to the estimation error problem is to keep all the elements in the sliding window for each counter in the CM sketch. However, in the sliding window we consider, the number of elements is unknown beforehand and can be very large. When the space is limited and the required space is uncertain, we should allocate this limited space efficiently. One mechanism is used to merge the elements systematically.

The merging mechanism is based on an early work over a sliding window of a data stream, i.e., the Exponential histogram(EH) [3]. The EH is proposed to estimate the sum of all the elements in a sliding window of a data stream. The basic approach is to use buckets of different sizes to hold the data in the data stream. When the number of buckets reaches a certain quantity, the buckets are merged together. Each bucket has a timestamp associated with it. This timestamp is used to decide when the bucket is out of the window and has to be dropped. We can estimate the sum of the data elements in the window, based on these buckets. This estimate is proven to be bounded within a user-specified parameter, ϵ . Thus, with the counters shared using the CM sketch, for each counter, we maintain an EH to merge the elements “exponentially.” The EH reduces the number of the data elements in the sliding window we need to keep for each counter. This leads to our second method.

The rest of this paper is organized as follows. Section 2 describes the mechanism of the CM sketch. In Section 3, two methods, the Discrete method and the Continuous method are introduced. We conclude this paper in Section 4.

2 Preliminaries

In this section, we are going to show how a CM sketch allows us to share counters among the m data sources in our data stream environment.

CM sketches assume a landmark data model over the same data stream environment as our problem definition, i.e., a single data stream with multiple data sources. It is designed to estimate the sum of all the elements for each data source in the data stream. The data structure is a matrix. To construct this matrix, a user has to supply two parameters, δ and ϵ , with which the estimates of a CM sketch are bounded within a factor of ϵ at $1 - \delta$ probability. The number of rows of this matrix, denoted as d , is $\lceil \ln(1/\delta) \rceil$ while the number of columns, denoted as l , is $\lceil e/\epsilon \rceil$. Each cell in the matrix contains a counter that is used to estimate the sums. Besides the matrix, there is

also a hash function for each row. This hash function is chosen randomly from a pairwise-independent family and maps a number from $[0 \dots m-1]$ to $[0 \dots l-1]$. That is, the hash function projects the data elements of the m data sources into the l cells for one row. The hash functions have the following form and their theoretical properties can be found in [6].

$h_i(x) = (a_i \cdot x + b_i \text{ mod } P) \text{ mod } l$ for $1 \leq i \leq d$, where P is a prime, $P \geq m$ and (a_i, b_i) is randomly chosen from $Z_P = \{0, \dots, P-1\}$

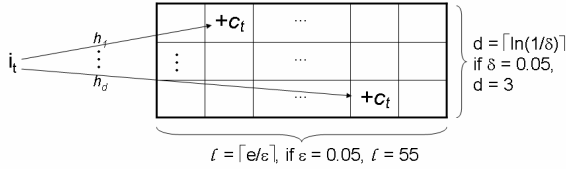


Fig. 2. The element, c_t , of data source i_t , is added to one cell in each row

Given a CM sketch as shown in Figure 2, when a data element (i_t, c_t) arrives, c_t is added to a counter in each row. The counter in each row is determined by the result of the hash function associated with that row, after we plug in the index of the data source, i_t . The number of columns of the CM sketch is usually far smaller than the number of data sources, and the hash functions allow us to share the counters among the data sources. In the following, we use an example to demonstrate the update procedure of the CM sketch and then we show how to estimate the sum of the elements of a data source.

Consider a data stream whose elements come from 7 different data sources. 7 is chosen for demonstration purpose and this number can be significantly larger than 7, for example, 1000. We want to estimate the sum of the elements of each data source using a CM sketch with $\epsilon = 0.5$ and $\delta = 0.05$. That is, we want to estimate to be bounded within a factor of 0.5 at 0.95(=1-0.05) probability. According to the formulas, the CM sketch is an matrix of $3(=\lceil \ln(1/0.05) \rceil)$ rows and $6(=\lceil e/0.5 \rceil)$ columns. Also, three hash functions are randomly chosen as follows:

$$\begin{aligned}
 h_1(i_t) &= (5 * i_t + 2 \text{ mod } 7) \text{ mod } 6 \\
 h_2(i_t) &= (3 * i_t + 1 \text{ mod } 7) \text{ mod } 6 \\
 h_3(i_t) &= (1 * i_t + 3 \text{ mod } 7) \text{ mod } 6
 \end{aligned}$$

Thus, as indicated in Figure 3, the elements of the fifth data source are projected into the first cell at the first row, the third cell at the second row and the second cell at the third row.

#5	#1,#4	#0	#3	#6	#2
#2,#4	#0	#5	#3	#1	#6
#3,#4	#5	#6	#0	#1	#2

Fig. 3. The CM sketch with $\epsilon = 0.5$ and $\delta = 0.05$ over a data stream of 7 data sources

Consider a stream of data elements (6,1), (5,5), (2,12), (4,15), (3,1), After the element (3,1) projected into the CM sketch, we have the result shown in Figure 4.

5	15		1	1	12
27		5	1		1
16	5	1			12

Fig. 4. The result after data elements are updated to the CM sketch

Now, if we want to estimate the sum of the elements of the fourth data source, we collect the cells that contains the elements of the fourth data source, i.e., the second cell in the first row, and the first cells in the second and third rows. The minimum value among the three cells is regarded as our estimate, i.e., $15 = \min \{ 15, 27, 16 \}$.

3 Proposed Methods

Having shown how to share the counters among the m data sources in a data stream, in this section, we show how to use this data structure to maintain the m sums in a sliding window.

3.1 The Discrete Method

Now, we consider the sliding window model over a data stream. Given a CM sketch started at the beginning of a data stream, at time T_{now-W} , we can estimate the sum of the elements for each data source in a data stream from the beginning up to T_{now-W} . Similarly, at time T_{now} , with the same CM sketch, we can estimate the sum for each data source from the beginning up to T_{now} . If the sums at time T_{now-W} are kept, we can estimate the moving sum of a data source by subtracting the estimate of the data source at time T_{now-W} from the estimate of that data source at T_{now} . That is, if we take a snapshot of the CM sketch at each time instant in the sliding window, we can estimate the moving sum of each data source by subtracting the estimate in the oldest snapshot from that in the current CM sketch.

However, we know that users may not be interested in the moving sums of the m data sources at every moment. For some applications, we may need to report the moving sums at every second, while for other applications, moving sums reported every 30 seconds are sufficient. That means we do not have to take a snapshot of the CM sketch at every moment. Instead, we take a snapshot at every t time units, assuming t divides W. Thus, in a sliding window of W time units, we keep W/t snapshots, with the latest snapshot taken at time T_0 , and the oldest snapshots taken at time T_0-W . The moving sum of a data source in the sliding window can be computed by subtracting the estimate of the data source in the snapshot at time T_0-W from the corresponding estimate in the snapshot at time T_0 and for T_0 not equal to T_{now} , we approximate the “continuous” moving sum of a data source between T_{now-W} and T_{now} with the sum between T_0-W and T_0 , which moves one step per t time units. We call this method the Discrete method. As illustrated in Figure 5, data elements are updated to only one

CM sketch, i.e., the matrix at T_{now} while other matrices are snapshots taken every t time units and we use the snapshots at $T_0 - W$ and T_0 to estimate the moving sums of the m data sources.

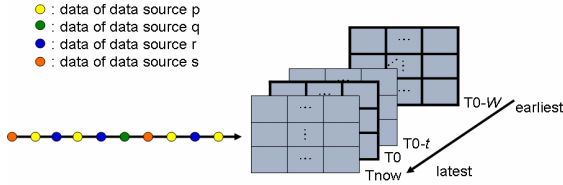


Fig. 5. The CM sketch with snapshots taken per t time units

In the following lemma, we prove that the estimate of the Discrete method is bounded with high probability.

Lemma 1. The estimate of the Discrete method for the i^{th} data source, $\hat{a}_{i,T_0} - \hat{a}_{i,T_0 - W}$, in the period between T_0 and $T_0 - W$, has the following guarantees:

$$a_{i,T_0} - a_{i,T_0 - W} \leq \hat{a}_{i,T_0} - \hat{a}_{i,T_0 - W}; \text{ and}$$

$$\hat{a}_{i,T_0} - \hat{a}_{i,T_0 - W} \leq a_{i,T_0} - a_{i,T_0 - W} + \epsilon a_{T_0} \text{ at } 1 - \delta \text{ probability.}$$

3.2 The Exponential Histogram

We have introduced the mechanism of the EH in Sec. 1. In this section, we briefly describe the update and query procedure of the EH.

Consider a simplified data stream environment where each element comes from the same data source and is either 0 or 1. In the EH, other than buckets, there are two additional variables, LAST, and TOTAL. The variable LAST stores the size of the last bucket. The variable TOTAL keeps the total size of the buckets. When a new data element arrives, we first check the value of the element. If the new data element is 0, ignore it; otherwise, create a new bucket of size 1 with the current timestamp, and increment the counter TOTAL. Given a parameter, ϵ , if there are $\lceil 1/\epsilon \rceil / 2 + 2$ buckets of the same size, merge the oldest two of these same-size buckets into a single bucket of double size. The larger timestamp of the two buckets is then used as the timestamp of the newly created bucket. If the last bucket gets merged, we update the size of the merged bucket to the counter LAST. We can see the size of the buckets grows exponentially, i.e., $2^0, 2^1, 2^2 \dots$

Whenever we want to estimate the moving sum, we check if the oldest bucket is within the sliding window. If not, we drop that bucket, subtract its size from the variable TOTAL and update the size of the current oldest bucket to the variable LAST. We repeat the procedure until all the buckets with timestamps outside of the sliding window are dropped. The estimate of 1's in the sliding window is $TOTAL - LAST/2$. It is shown in [3] that, for N 1's in the sliding window, we only need $O((\log N)/\epsilon)$ buckets to maintain the moving sum and the error of estimating the moving sum is proven to be bounded within a given relative error, ϵ .

3.3 The Continuous Method

While CM sketches allow us to share counters among the data sources, the remaining problem is how to merge the elements in the sliding window systematically so that we do not have to keep every element in the sliding window. Observe the update procedure of a CM sketch. Although the data stream environment of our problem is a data stream with multiple data sources, from a cell’s view point, it does not matter which data source a data element belongs to. What a cell does is adding the element up to the counter in that cell. Therefore, the data stream environment a cell faces is a single-source data stream and is exactly the same data stream environment assumed by the EH. After we add a value to the counter of a cell, we have to take it out when the value moves out of the window and the moving sum over a timestamp-based sliding window is exactly the problem the EH is proposed to solve.

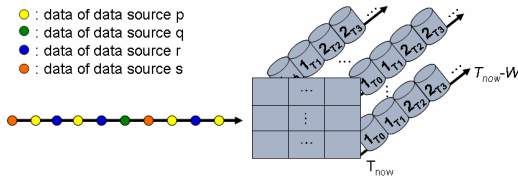


Fig. 6. The CM sketch with an EH embedded in each cell

Therefore, we embed EHs into the cells of the CM sketch. That is, we construct an EH for each cell using the same parameter, ϵ , as that of the CM sketch. To estimate the moving sum for the i^{th} data source, we collect the cells containing the elements from the i^{th} data source, and report the minimum of the values estimated by the EH in each cell as the estimate for the i^{th} data source. Since these EHs maintain the elements in the sliding window and drop them off after they move out of the window, the estimates of this method do not have the problem of error accumulation, which meets our goal. Moreover, by merging the elements “exponentially”, we do not have to keep all the elements in the sliding window for each cell. Unlike the Discrete method, the estimates in this method are based on the current elements in the window, and the moving sums can be approximated continuously. We call this method the Continuous method.

Denote the sum of all the elements in the data stream in the sliding window from $T_{\text{now}}-W$ up to T_{now} , as $a_{(T_{\text{now}}-W, T_{\text{now}})}$, the moving sum of the i^{th} data source in the window as $a_{i,(T_{\text{now}}-W, T_{\text{now}})}$ and the estimate of the Continuous method for the moving sum of the i^{th} data source as $\hat{a}_{i,(T_{\text{now}}-W, T_{\text{now}})}$. We prove in lemma 2, that the estimate of the Continuous method is also bounded with high probability.

Lemma 2. The estimated value of the Continuous method, $\hat{a}_{i,(T_{\text{now}}-W, T_{\text{now}})}$, for the i^{th} data source, in the sliding window between $T_{\text{now}}-W$ and T_{now} , has the following guarantees:

$$(1 - \epsilon) \cdot a_{i,(T_{\text{now}}-W, T_{\text{now}})} \leq \hat{a}_{i,(T_{\text{now}}-W, T_{\text{now}})}; \text{ and}$$

$$\hat{a}_{i,(T_{\text{now}}-W, T_{\text{now}})} \leq (a_{i,(T_{\text{now}}-W, T_{\text{now}})} + \epsilon a_{(T_{\text{now}}-W, T_{\text{now}})}) \cdot (1 + \epsilon) \text{ at } 1 - \delta \text{ probability.}$$

4 Future Work

Inspired by [2], we are considering the problem of maintaining biased moving sums. For example, in the data stream of m data sources, we are interested in, say, the moving sum of the i^{th} data source or the i^{th} largest moving sum, for $i \in [1 \dots m]$, so we want that estimate to have a tighter bound. After a user specified the preference as an input to the construction of a data structure, this data structure allocates space accordingly, that is, more space to the interested estimate and less space for others. By giving more space for the moving sum we are interested, we can achieve a better estimate than those of the methods proposed in this paper. Currently, we are investigating ways of modifying our methods to solve such a problem.

References

1. G. Cormode, and S. Muthukrishnan. "Improved Data Stream Summary: The CM sketch and its Applications." In *Journal of Algorithms*, April 2005.
2. G. Cormode, S. Muthukrishnan, F. Korn and D. Srivastava. "Effective Computation of Biased Quantiles over Data Streams." In *Proceedings of the 21st International Conference on Data Engineering*, pp. 20-31, 2005.
3. M. Datar, A. Gionis, P. Indyk, and R. Motwani. "Maintaining Stream Statistics over Sliding Windows." In *SIAM Journal on Computing*, Volume 31(6), pp. 1794-1813, 2002.
4. C. Lin, D. Chiu, Y. Wu, and A. Chen. "Mining Frequent Itemsets in Time-Sensitive Sliding Window over Data Streams." In *SIAM International Data Mining Conference*, 2005.
5. X. Lin, J. Xu, H. Lu, and J. X. Yu. "Continuously Maintaining Quantile Summaries of the Most Recent N elements over a Data Stream." In *Proceedings of the 20th International Conference on Data Engineering*, pp. 362-374, 2004.
6. R. Motwani, and P. Raghavan. "Randomized Algorithms." Cambridge University Press, 1995.
7. L. Qiao, D. Agrawal, and A. El Abbadi. "Supporting Sliding Window Queries for Continuous Data Streams." In *Proceedings of 15th International Conference on Scientific and Statistical Database Management*, pp. 85-94, 2003.
8. Y. Zhu and D. Shasha. "StatStream: Statistical monitoring of thousands of data streams in real time." In *Proceedings of the 28th International Conf. on Very Large Data Bases*, pp. 358-369, 2002.

MFIS—Mining Frequent Itemsets on Data Streams

Zhi-jun Xie, Hong Chen, and Cuiping Li

School of Information, Renmin University, Beijing, 100872, P.R. China

Abstract. We propose an efficient approach to mine frequent Itemsets on data streams. It is a memory efficient and accurate one-pass algorithm that can deal with batch updates. The proposed algorithm performs well by dividing all frequent itemsets into frequent equivalence classes and pruning all redundant itemsets except for those that represent GLB (Greatest Lower Bound) and LUB (Least Upper Bound) of the frequent equivalence classes. The number of GLB and LUB is much less than the number of frequent itemsets. The experimental evaluation on synthetic and real datasets shows that the algorithm is very accurate and requires significantly lower memory than other well-known one-pass algorithms.

1 Introduction

Many real applications need to handle data streams, such as stock tickers, click streams and sensor net works. Mining frequent itemsets forms the basis of algorithms for a number of association rules mining problems [7]. A main limitation of the existing work on frequent itemset mining is the high memory requirement when the support level is quite low or the number of distinct items is large. In this paper, we present a new approach for frequent itemset mining on data stream. Our work addresses three major challenges in frequent itemset mining in a streaming environment. Firstly, we propose a method to find frequent itemsets while keeping limited memory. The most significance of our algorithm is that it can handle a large number of distinct items and with small support levels using a relatively smaller amount of memory. Secondly, our algorithm is accurate in practice and in theory. Thirdly, we have designed a new data structure FIET (Frequent Itemset Enumeration Tree) and a novel approach to maintain it. A detailed evaluation using both synthetic and real datasets is carried out. Experimental results show that the proposed algorithm has significantly improved the performance over other approaches for mining frequent itemsets on data streams.

The work close to ours on handling streaming data is found in Jin and Agrawal [5]. They presented a one-pass algorithm that does not allow false negatives and has a deterministic bound on false positives. The difference between our approach and theirs are in two aspects. First our algorithm is very accurate in theory and in practice. Our experiments demonstrated an accuracy of 100%. While their algorithm could in a few cases reach 100% accuracy and when ϵ increases, the accuracy would decrease. Second our algorithm is very efficient with respect to space requirement. This difference also shows that our batch maintenance algorithm can deal with multiple tuples per update,

while the algorithm in [10] handles single-tuple updates. Manku and Motwani [9] also presented a one-pass algorithm that did not allow false negatives and had a provable bound on false positives. Their algorithm achieved this through an approach called lossy counting. Their algorithm needs an out-of-core structure, and lower than In-Core mentioned by Jin and Agrawal in accuracy and memory efficient [5]. Giannella *et al* had developed a technique for dynamically updating frequent patterns on streaming data [6]. Their algorithm creates a variation of FP-tree, called FP-stream for mining time-sensitive frequent patterns. As our experimental results show, the memory requirement of our approach is significantly lower than those of FP-tree approaches.

2 Problem Statement

Given a set of items Σ (where Σ is not fixed size), a data stream DS where tuples are of a subset of Σ , and a threshold s called minimum support (min-supp), $0 < s \leq 1$. The frequent itemset mining problem is to find all itemsets that occur in at least $s * |DS|$. Our problem is to mine frequent itemsets in N tuples in a data stream. Suppose N is the length of data stream DS at present time, i.e., $N = |DS|$. Each tuple has a time stamp, which is denoted as *tid* (tuples id). Figure 1 shows an example of $\Sigma = \{A, C, D, E, T, W\}$, with min-supp $s = 1/2$. To find frequent itemset on a data stream we firstly construct a frequent itemset enumeration tree (FIET) in the memory, and then divide all itemsets into equivalence classes according to their *tids*, then find the GLB and the LUB of every class and prune the other items except the frequent itemsets that represent the GLB and LUB of the classes. Secondly we update the equivalence classes in FIET.

3 Mining and Maintenance Algorithm

We give the concept and some properties of equivalence class first and then briefly explain the principles of maintenance and batch maintenance algorithms.

3.1 Equivalence Class and Frequent Itemset Enumeration Tree

Closure operator CS and the lemma of closure operator [10] defines a set of equivalence classes over the lattice of frequent itemsets, i.e., two itemsets belong to the same equivalence class iff they have the same closure, i.e. they are supported by the same set of tuples.

Definition 1(Equivalence class). A set of itemsets in a lattice of frequent itemsets is said to belong to the same equivalence class, C , if

- a. Given any two items I and I' in C which satisfy $I \prec I'$, any intermediate items I'' satisfying $I \prec I'' \prec I'$ will also be in C .
- b. All the items with the same closure belong to the same equivalence class.

Figure 1 shows that the itemsets with the same closure are grouped in the same equivalence class. Each equivalence class contains elements sharing the same *tidset*. All the possible frequent itemsets can be organized into the equivalence classes C and

each frequent item is represented with an element of the equivalence class. An equivalence class is a partially order set (C, \prec) , in which every pair of elements in C has a Least Upper Bound (LUB) and one or more Greatest Lower Bound (GLB) within C .

Definition 2 (LUB and GLB). Given a set of elements E in an equivalence class C , the least upper bound (LUB) of E is an element $u \in C$ such that $e \prec u$ for all $e \in E$ and there exists no u' such that $e \prec u'$ for all $e \in E$ and $u' \prec u$. Likewise, the greatest lower bound (GLB) of E is an element $l \in C$ such that $l \prec e$ for all $e \in E$ and there exists no l' such that $l' \prec e$ for all $e \in E$ and $l \prec l'$, LUB is unique.

Lemma 1. Given two itemset X and Y , and the closure operator CS , function f and g is the base function of CS . If $X \subset Y$, and $g(X)=g(Y)$, then $CS(X)=CS(Y)$.

Proof: if $g(X)=g(Y)$, then $f(g(X))=f(g(Y))$ then $CS(X)=CS(Y)$.

From Lemma 1, given a generator X , if we find an already mined Closed Itemsets Y that set includes X , where X and Y have the same support and *tidset*, we can conclude that $CS(X)=CS(Y)$. In an equivalence class, the closed itmeset Y is actually the LUB of the class [8], and all the other items in the equivalence class is the generator of LUB and the GLB is the most general among all generators [8]. Hence we can prune the generator except the GLB and LUB without losing any information, since the frequent itemsets which have been pruned can be inferred from them. For instance, for equivalence class $C5$ in Figure 1, its LUB is “TAWC” (which is unique in an equivalence class) and its GLB is “TA” and “TW”. We call “TAC”, “TAW”, “TCW”, “TA”, “TW” the generator of “TAWC”, but “TA”, “TW” is the minimal generator. So in a frequent class we only store the GLB and the LUB. In this example, the generators TAC , TWC and TAW can be inferred from $TAWC$ and TA , TC , since $TA \cup TW=TAW$ and $TAWC-TAW=C$, so $TAC=TA \cup C$, $TWC=TW \cup C$. Hence we can prune all other itemsets except for the itemsets which represent GLB and LUB in an equivalence class. Obviously, the number of LUB and GLB is much less than the number of items. Based of this concept, we first construct a FIET in the memory, plot out the equivalence classes in FIET, and prune redundant frequent items except for the itemsets which represent GLB and LUB of the equivalence classes.

The in-memory data structure-frequent itemset enumeration tree (FIET) was originally introduced in [10], which monitors a dynamically selected set of itemsets that enable us to answer the query “what are the current frequent itemsets” at any time. Similar to prefix tree, each node n_l in a Frequent Itemset Enumeration Tree (FIET) represent a frequent itemset I . However, unlike a prefix tree, which maintains all itemsets, a FIET only maintains the frequent itemsets. We further divide itemsets on the boundary into two categories, which correspond to the boundary between frequent and infrequent. We divide all infrequent boundary itemsets into the same equivalence classes and mark them as infrequent classes. According to the enumerate property [10], all the itemsets in infrequent class do not need to enumerate. Actually, all members of infrequent classes are single items (sensitive nodes) and the size of the infrequent classes is very small. As for the frequent itemsets, all the itmesets divided into the same

equivalence class if they have the same *tid*. We prune the redundant itemsets except for those itemsets that represent the GLB and LUB. For instance, in Figure 1, infrequent equivalence class *C5*, *TAWC* is the LUB and *TA*, *TW* is the GLB, other redundant itemsets can be omitted, we mark them by using double strikethrough notations.

Similar to the *IT_Tree* in the *Charm* [6], each node in the *FIET*, represented by an itemset-tidset pair, $X \times t(X)$, show in Figure 3, such as *TAWC* × 135. In order to efficiently maintain the *FIET*, we use different set of buckets to store the equivalence classes. We only store the GLB and LUB for each of the classes in the buckets. To build a *FIET*, we firstly create a root node n_0 . Secondly, we create $|\Sigma|$ child nodes for n_0 , i.e., each $i \in \Sigma$ corresponds to a child node n_i , we call these nodes **sensitive nodes**. The sensitive node has two statuses, inactive and active. In general, they are inactive. They will be active when the new tuples has changed their support. After having built the entire sensitive node, Enumerate procedure [10] on each node n_i will be called. During the enumeration, we can divide all the frequent itemset into equivalence classes and prune the redundant itemsets except for those represent GLB and LUB.

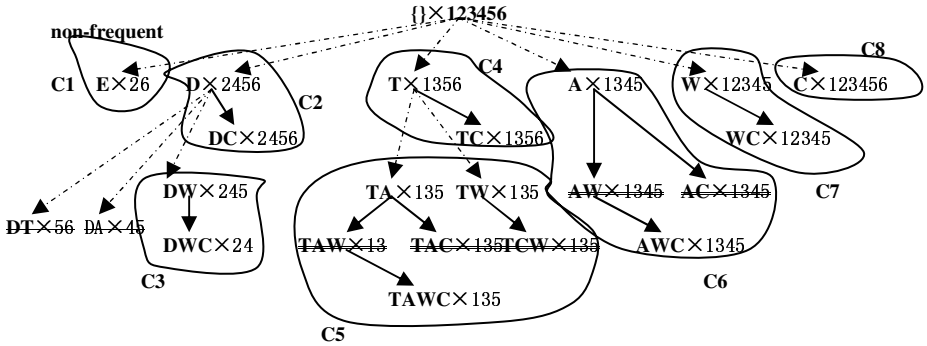


Fig. 1. The Frequent Itemset Tree Corresponding to the sample dataset

3.2 Maintenance of FIET

In this section, we denote the set of frequent equivalence classes in the *FIET* as Ψ , and the new set of frequent equivalence classes as Ψ' . We consider the case of incremental updates.

3.2.1 The Principle of Maintenance

As we discussed before, we divide the equivalence class into two categories, frequent equivalence class and non-frequent class. New tuple t can affect both the frequent equivalence classes and non-frequent classes. For the frequent equivalence class, a new tuple t can affect an equivalence class in Ψ in several ways. First, it can cause the support of the equivalence class to change without affecting the partition of the lattices. Second, it might cause the equivalence class to split; creating some new equivalence classes. The final possibility is that the equivalence class might not be affected at all.

When a new tuple matches [10] the least upper bound of an equivalence class, the new tuple t will cause the support of the equivalence to be changed. More importantly, we know that the tuples will match every pairs in the equivalence class since the class's least upper bound is the most specific pair in the whole equivalence class. In this case the equivalence class can't split.

Proposition 1 (VALUE MODIFIED CLASS). Given a frequent equivalence class C in Ψ , if a new tuple matches $C.lub$, C need not to be split and $C.tidset$ must be modified.

When t matches only a certain portion of $C.lub$, i.e. t can only match a portion of the pairs in C , C must be split into two portions, one in which all pairs match t and one in which all pairs do not. A new tuple t partly affects a class C only if there is some intersection between t and $C.lub$.

Proposition 2 (NEW CLASS GENERATOR). Given a new tuple t , an existing frequent equivalence class C must be split if (1) C intersects with t and C is the class that contributes to the GLB $\{Y|Y=C'.lub \cap t\}$, $C' \in \Psi$; and (2) there does not exist any class $C'' \in \Psi$ such that $C''.lub=C.lub \cap t$; If these two conditions are satisfied, we call C a new class generator since the splitting will result in a new equivalence class.

The first condition of Proposition 2 ensures that given all classes which generate the same upper bound for the new class C_n , the one that is the most general will be the new class generator, and the first condition according to the property of class generator [10].

Definition 3 (split operation). Given a generator class $Cg(Cg \subseteq \Psi)$ and a new tuples t , the split operation on it, and Cg generates a new class C_n and a modified generator $C'g$, the split operation as follows:

$$C_n.lub=Cg.lub \cap t; C_n.tidset= Cg.tidset \cup t.tidset; C'g.lub=Cg.lub; C'g.tidset=Cg.tidset$$

The last proposition involves a simple category of equivalence classes that neither match nor intersect the new tuple t .

Proposition 3 (DUMB CLASS). If an equivalence class C in Ψ is neither a modified class nor a generator, there is no need to change C . We call C a dumb class.

When the itemset in the frequent equivalence class becomes infrequent, we need to delete the whole class since all the itemsets in the same equivalence have the same support. However, if the classes contain a sensitive node, we must mark it infrequent and put it into non-frequent class, because we can get the potentially frequent itemset in the future by enumerate the sensitive node when it becomes frequent. When a tuples contains the item which is appeared first time, we put it into the non-frequent class and mark it sensitive node. When the new tuple t affects the non-frequent class, it only causes the support of the items to change. when the support of the items exceeds the threshold of min-supp, we need to change their status to active and the procedure Enumerate will be called to create new frequent equivalence classes in the end.

Above all lay the foundation for maintaining the FIET and we give the single maintenance algorithm in [10]. As we know, the maintenance time complexity is linear to the number of frequent equivalence classes, since the number of classes is much less than the number of the items, Our algorithm is much more time efficient than those algorithms linear to the number of items.

3.2.2 Batch Maintenance Algorithm

The Single maintenance algorithm [10] only handles with single tuples per update. In reality, there are situations in which data are burst and multiple tuples need to deal with during one update. However, it is not difficult to handle with multiple tuples in one update. Originally, if an update contains a batch of tuples, we can recursively partition on both the existing classes and the set of tuples, each partition do the update similar to the algorithm single maintain. We introduce Batch_maintenance algorithm for the FIET, Bach_maintain is inspired by the inc_Batch algorithm proposed by Cuiping Li [8] which recursively partitions tuples and classes in a depth-first manner so that tuples involved in computing the same cell are grouped together at the same time of computation for the cell's value. The partition is performed on different groupings can be formed. Our algorithm performs partition on both the existing classes in Ψ which represented by their upper bounds and the new set of tuples. We refer to the partition of the new tuples as tuples partition and the partition of equivalence classes as class partition. To ensure the effectiveness of the algorithm, we synchronize to partition the tuples and classes: a particular tuples partition that is being processed at one time is guaranteed to affect only the corresponding class partition that is being processed at the same time. As doing the partition of equivalence classes in synchronization with the tuples partitioning, the number of the equivalence classes that are being checked is substantially reduced, which will enhance the efficiency of our algorithm. The operation of synchronization is performed in the function of "partition_value()".

Algorithm: Batch_maintain($R, \Psi, Dims$),

{Input : R : the set of tuples; Ψ :the set of equivalence classes;
Dims: the total number of dimensions.

Output: a set of updated classes

1. Partition_value($\emptyset, R, \Psi, Dims$)
2. Return the the modified classes and new classes

}

Function : Partitinal_value($\bullet, tuples, classes, dim$)

{ \bullet :pairs in equivalence classes to be processed.

P_tuples: a tuple partition

P_classes: a class partition

dim: the starting dimension for this iteration

Add the virtual class VC to the classes

2. UpdateandGenerate($\bullet, tuples, classes$)

3. for $D=dim$ to $Dims$ do

4. partition tuples on dimension D ; partion classes on dimension D

```

5.   for i=0 to cardinality|D|-1 do
6.       P_tuples=tuple partition for value x of dimension D
7.       p_classes=class partition for value x of dimension D
8.       if |p_tuples|>0 then
9.           Partitinal_value( $\bullet \cup x, p\_tuples, p\_classes, D+1$ )
10.      end if ; end for; end for.
}

```

Function: UpdateandGenerate(pt_Value, tuples, classes)

```

{
1. Sort the class based on ascending cardinality
For each class C in classes do
  If  $C.lub \subseteq pt\_value$  then
    For each tuple t in tuples do
       $C.tidset = C.tidset \cup t.tid$  ;
      break for.
  Else
    If  $C.lub \supset pt\_value$  then
      Find the upper bound upbdPair of tuples
      If upbdPair==C.lub then
        break for;
      end if
    Generate temp class Ct,  $Ct.lub = upbdPair$ 
    If  $\neg \exists C' \in C.tempset$  and  $Ct.lub = C'.lub$  then
      Add Ct to C.tempset
    End if
    Break for;
  End if
End if
End for}

```

The main algorithm simply calls the function *partition_value()* and providing the set of new tuples R and the original set of equivalent classes Ψ , the number of dimension at the same time. The function *partition_value()* will perform recursive partition of both the tuples and equivalence classes until get to the virtual class. At last, the main algorithm will output the new generated classes and the modified classes.

As for the function *UpdateandGenerate()*, which is to determine the "tuples" how to affect the "classes", the approach is similar to the approach *single_maintain*. The main difference is that the *pt_value* (*partition Value*) is used to match with equivalence classes instead of the pairs which is used in *single_maintain* algorithm. In function *UpdateandGenerate()*, the tuples is match with classes again according to increasing order of cardinality for equivalence classes (line 1). If $C.lub \subseteq pt_value$, means all tuples match the class C , we need add all tuples' *tid* to the tidset of the class. If $C.lub \supset pt_value$, means that the class has the intersection with the tuples, we first find the upper bound (*upbdPair*) of the tuples by appending all dimensional values that have 100% appear in tuples to the *pt_value*. If $upbdPair == C.lub$, the Class C will be

updated in future recursion and need do nothing on it. However, if $upbdPair \neq C.lub$, we will create a temporary class Ct (line 12), if the class Ct is not in the $C.tempset$, which contains all new classes that are generated from C and will be output in the main algorithm, we add it to $C.tempset$. If there exists a generated class C' which $Ct.lub=C'.lub$, we discarded Ct since Ct and C' are the same class in fact.

4 Experimental Results

The synthetic datasets we used were generated using a tool from IBM [1]. We will show our algorithm’s ability to handle very low support levels and a large number of distinct itmsets. The first dataset we used is T10I4.N10K. Figure 2 shows the memory requirements of *Apriori*, FP-tree and our algorithms (single_maintain and Batch_maintain) as the support threshold is varied from 0.1% to 1.0%. The number of tuples is 12 million. Because of high memory requirements, FP-tree could not be executed with support levels lower than 0.4%. This limitations of the FP-tree approach has been identified by other experimental studies also [3]. The important property of our algorithm is that the memory requirements do not increase significantly as the support level is decreased. Figure 3 compares the execution time. As expected both single_maintain and Batch_maintain algorithms have the lowest execution time among all other algorithms. Figure 4 examines the execution times as the dataset is increased;

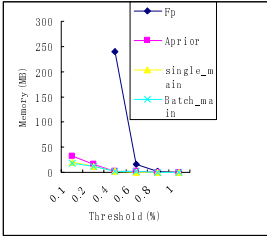


Fig. 2. Memory Requirements with changing Support level (T10.I4.N10K)

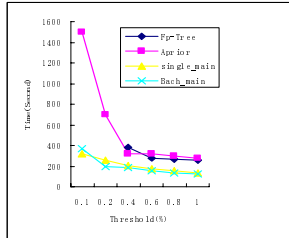


Fig. 3. Execution Time with changing Support Level (T10.I4.N10K)

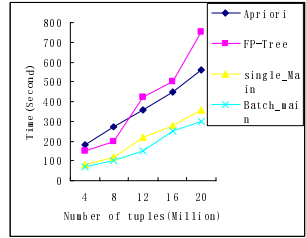


Fig. 4. Execution Time with increasing Dataset Size (threshold=0.4%, T10.I4.N10K)

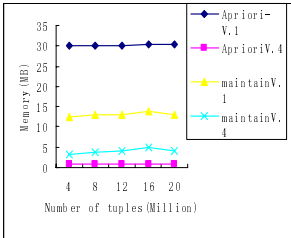


Fig. 5. Memory requirements with increasing data size (T10.I4.N10k)

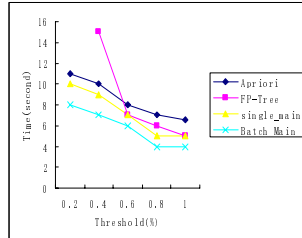


Fig. 6. Execution time with changing Support level (BMS-WebVew-1)

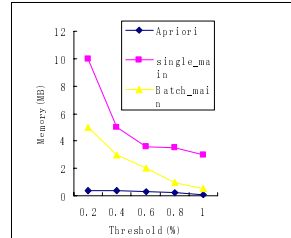


Fig. 7. Memory requirements with Changing support level (BMS-WebVew-1)

our algorithm is faster than *Apriori* and FP-tree when the data size is varied. More importantly, our algorithms can always give an accuracy of 100%. Figure 5 focuses on memory requirements with support level of 0.4% and 0.1%. Because of high memory requirements of FP-tree, our algorithm is only compared against *Apriori*.

The real dataset we use is the BMS-WEBView-1 dataset which contains several months of click stream data from an e-commerce website, and is also used by Jin [5]. In our experiments, we duplicated and randomized the original dataset to obtain 1 million tuples, as has been done in the In-Core [5]. Our algorithm is competitive in accurate and when the support level is very low, just as shown in Figure 6, when the support level is lower than 0.4%, our algorithm is faster than *Apriori* and FP-tree. In Figure 7, the memory cost of *Apriori* is very low, because of the number of frequent itemset is relatively small.

5 Conclusions

In this paper, we have developed a new approach for frequent itemset mining on data streams. We propose a memory efficient and accurate one-pass algorithm, which can deal with batch updates. In maintenance of frequent itemsets, an efficient in-memory data structure FIET is used to record all equivalence classes. In addition, we have developed an efficient algorithm to incrementally update the FIET when batch tuples arrive. Our detailed experimental evolution has shown our algorithm is very accurate and space efficient in theory and practice, and allow us to deal with large number of distinct items at low support levels. At last, the memory usage and time complexity of our algorithm are shown to be linear in the number of equivalence classes in FIET.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. Very Large Databases (VLDB'94), pages 487-499, Santiago, Chile, September 1994.
2. C. Borgelt. Apriori implementation. <http://fuzzy.cs.UniMagdeburg.de/borgelt/Software>.
3. J. Gehrke, F. Korn, and D. Srivastava. On computing correlated aggregates over continual data streams. In Proc. ACM SIGMOD Management of Data, pp 13-24. June 2001.
4. Bart Goethals. FP-tree Implementation. <http://www.cs.helsinki.fi/u/goethals/software/index.html>.
5. R. Jin and G. Agrawal. An algorithm for in-core frequent itemset mining on streaming data, International Conference on Data Mining 2005 (ICDM2005), pp 210-217.
6. M.J. Zaki and C. Hsiao. Charm, An efficient algorithm for closed itemset mining. In 2nd SIAM Data Mining, 2002
7. J. Han, J. Pei, and Y. Yin. Mining Frequent patterns without candidate generation. In Proceedings of the ACM SIGMOD Conference on Management of Data, 2000.
8. Cuiping Li, Gao Cong, Anthony K. H. Tung, Shan Wang, Incremental Maintenance of Quotient Cube for sum and Median, SIGKDD pp.226-235, 2004
9. G.S. Manku and R. Motwain. Approximate Frequency Counts Over Data Streams. In proceedings of Conference on Very Large Database (VLDB), pp 346-357, 2002
10. ZJ. Xie, H. Chen, and C. Li, An Efficient Algorithm for Frequent Itemset Mining on Data Streams. The 6th Industrial Conference of Data Mining 2006 (ICDM2006), LNAI 4065, pp. 474 – 491, 2006.July 2006, Leipzig ,Germany

Improving the Performance of Data Stream Classifiers by Mining Recurring Contexts*

Yong Wang¹, Zhanhuai Li², Yang Zhang^{3,**}, Longbo Zhang⁴, and Yun Jiang⁵

^{1,2,4,5}Dept. Computer Science & Software, Northwestern Polytechnical University, China
{wangyong, zhanglongbo, jiangyun}@mail.nwpu.edu.cn,
lizhanhuai@nwpu.edu.cn

³School of Information Engineering, Northwest A&F University, China
zhangyang@nwsuaf.edu.cn

Abstract. Traditional researches on data stream mining only put emphasis on building classifiers with high accuracy, which always results in classifiers with dramatic drop of accuracy when concept drifts. In this paper, we present our RTRC system that has good classification accuracy when concept drifts and enough samples are scanned in data stream. By using Markov chain and least-square method, the system is able to predict not only on which the next concept is but also on when the concept is to drift. Experimental results confirm the advantages of our system over Weighted Bagging and CVFDT, two representative systems in streaming data mining.

1 Introduction

Recently, mining data streams with concept drifting for actionable insights has become an important and challenging task for a wide range of applications, including credit card fraud detection, network intrusion detection, etc.

A substantial amount of recent work has focused on continuously mining of data streams [1-6, 12, 13]. However, a number of problems remain unsolved. Firstly, previous approaches are interested in predicting the class label of each specific sample. No significant effort has been devoted to foresee oncoming concept. Secondly, existing approaches may launch a new prediction strategy upon detecting a concept change. However, the concept drifting can only be detected after it has happened. Generally, this means a long delay after the concept drifting and hence the accuracy of classify is low during the delay. In other words, it still remains an open problem whether a concept drifting can be predicted.

In this paper, we propose a novel Recognizing and Treating Recurring Contexts (RTRC) system to solve the above problems. In our method, we propose to build a model of data streams history. This model represents compact and essential knowledge abstracted from raw data. Hence, one can easily keep a long history of concepts. When contexts recur, a mechanism for predicting oncoming concepts is used. Furthermore, this model can learn the duration pattern of the recurring concept.

* This work is supported by NSF 60373108.

** Corresponding author.

With the help of this duration pattern, the position where concept drifts could be predicted beforehand, and hence the current concept could be replaced by predicted concept in time. Experimental results on both synthetic and benchmark datasets demonstrate the advantage of our system.

This paper is organized as following. Section 2 reviews the related works; section 3 presents the method for building the model of data stream history; section 4 presents our algorithm for recognizing and treating recurring contexts; section 5 gives our empirical results; and section 6 concludes this paper.

2 Related Work

Much research has been performed on classifying data streams [4, 10, 11]. However, few classifiers are able to deal with recurring contexts. FLORA3 system [7] stores old concepts and reuses them when necessary. However, it is specially designed for small size data, not data streams. It represents concepts by conjunctions of attribute values and measures the conceptual equivalence by syntactical comparison, which is not applicable for data streams. Furthermore, FLORA does not explore the stored associations of old concepts and hence can not foresee the coming concept. PECS [8] uses lazy learning for handling concept drifting and stores samples for further reactivating. SPILICE [9] uses a meta-learner, which works in off-line and batch way, to extract hidden context, and to induce the local concepts. PrePro [10] can conduct a reactive prediction by detecting concept drifting and modifying the prediction model for oncoming samples, and a proactive prediction by foreseeing the coming concept given the current concept. The proactive mode predicts the forthcoming concept ahead of a concept drifting. Once a new trigger detected, the predicted concept immediately takes over the classification task. However, the trigger can be detected only after a concept drifting has happened. Generally speaking, this means a long delay after the concept drifting as PrePro doesn't try to foreseeing the oncoming concept drifting.

The Weighted Bagging approach [11] represents a big family of algorithms that use ensemble learners for prediction. This method uses a history of concepts (classifiers). However, the quality of this history is controlled by an arbitrary chunk size k , with no trigger detection nor conceptual equivalence. This algorithm tries to adapt to changes by assigning weights to classifiers proportional to their accuracy on the most recent data block.

The CVFDT approach [4] is one of the most well-known systems and represents another popular method that continuously adapts the prediction model to oncoming samples. In order to handle concept drifting, authors have chosen to retire old examples at a preset fixed rate. In cases where history repeats itself, CVFDT do not take advantage of previous experience and hence converge to new target concepts slowly when the concept drifts.

Our approach can effectively save data stream model in memory and explore association among different concepts. Furthermore, the patterns for concept drifting could be learned by making good use of the recurring contexts, and these patterns could be used for predicting oncoming concept drifting. In section 5, we conduct empirical comparisons among RTRC, Weighted Bagging and CVFDT to verify our approach.

3 Building the Model of Data Stream History

3.1 Model of Data Stream

Daily experience shows that due to cyclic phenomena, hidden contexts may be expected to recur. The seasons of a year could be looked as an example of regular phenomena; and inflation rates or market could be looked as an example of irregular phenomena. The core idea of our RTRC is that the contexts and its duration both can recur. For example, both the season, and the duration of the season, can recur. Therefore, we can prepare clothing just before wintertime, instead of changing clothing rules long after the winter begins. In other words, we can predict what next concept is, and how long the current concept will persist. The recurring contexts are helpful for foreseeing oncoming concepts, and the duration of the recurring contexts is helpful for predicting concept drifting. If these patterns can be learned from data streams, then the target, “the prediction error ideally should not be correlated to the amount of concept drifting” [1], could be achieved.

Suppose i is an integer, we write an integer, say, I_i , for the concept identifier; we write $Classifier_i$ for the corresponding concept, which is in the form of a predictive model; and we write $Duration_i$ for the corresponding duration of the hidden context. We define:

Definition 1. A model of data stream is defined as a sequence:

$$\dots(i-1, I_{i-1}, Classifier_{i-1}, Duration_{i-1})(i, I_i, Classifier_i, Duration_i)(i+1, I_{i+1}, Classifier_{i+1}, Duration_{i+1}) \dots$$

In this sequence, for a certain i , we say $(i, I_i, Classifier_i, Duration_i)$ is an element. Here, i means the position of this element in the sequence. The recurring contexts should have same concept identifier I_i . Each element retains essential information of the past data and can keep a long history of the streaming data. So, they can be reexamined and reused in the future when necessary. The possible associations among different concepts could be learned, and the concept duration can be recorded for learning more complicated patterns, which is very valuable because it could help us to prepare for the oncoming concept drifting.

3.2 Predictive Model and Duration of Prediction Model

Let’s write C_i for a category in the category set C_{set} ; V_j for an attribute in the attribute set V_{set} ; and X for a sample. The predictive model is constructed by an incremental version of naïve Bayes algorithm.

For discrete attributes, the algorithm stores the count for samples in each class C_i and the count for each V_j takes some certain value in C_i . In the learning phase, the algorithm uses these counts to compute $P(C_i)$ and $P(V_j | C_i)$. Then, under the Bayesian assumption that each attribute is conditionally independent on other attributes, the category of the sample X is predicted by:

$$P(C_i | X) = \arg \max_{C_i} P(C_i) \prod_j P(V_j | C_i) . \tag{1}$$

For continuous attributes, say, v_j , our algorithm stores the sum of its attribute values and the sum of its squared values for each C_i . In the learning phase, the algorithm uses the sample count and these sums to compute the mean (μ) and variance (δ^2) for each attributes. Then, assuming that the attribute values of v_j are normally distributed, it computes

$$P(V_j | C_i) = \Delta V_j \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(V_j - \mu)^2}{2\delta^2}}. \quad (2)$$

Here, ΔV_j is the size of interval in which the random variable for the attribute lies. (See [14] for details).

A sliding-window is used to detect concept drifting before enough information is gathered. Here, the window size and the error threshold are important parameters. Each incoming sample will be classified by the current prediction model. The first sample in the window is always a wrongly classified sample. When the window is full, if the error rate of the window exceeds the threshold, the first sample of the window is taken as a trigger; otherwise, the beginning of the window slides to the next wrongly classified sample, and the previous starting sample is dropped from the window. The number of samples between two adjacent trigger is the duration of current prediction model.

3.3 Similarity Between Predictive Models

When we add a predictive model C and its corresponding duration to the data stream model, we need to test the similarity between two target concepts. In reality we usually do not know the target concept, therefore the similarity between target concepts can be estimated by their optimal predictive models. If the similarity between a newly learned predictive model C and a previous predictive model C_j exceeds a pre-defined threshold, the system believes that C is a reappearing of C_j .

```

Input: Data set  $D = \{(X_1, y_1), \dots, (X_n, y_n)\}$ 
       Predictive model  $C$  newly learned from  $D$ 
       Previous predictive model  $C_j$ 
Output: Similarity
begin
FOR i FROM 1 TO n {
  ClassifyResult1 =  $C(X_i)$ ;
  ClassifyResult2 =  $C_j(X_i)$ ;
  IF (ClassifyResult1  $\neq$  ClassifyResult2) weight = -1;
  IF (ClassifyResult1 = ClassifyResult2) {
    IF (ClassifyResult1  $\neq$   $y_i$ ) weight = 0;
    ELSE weight = 1;
  }
  similarity = similarity + weight;
}

```

```

}
Return similarity;
end.

```

Algorithm 1. Measuring similarity between two predictive models

We illustrate algorithm 1 by a simple example. In figure 1, the two hyperplanes stand for two different target concepts, and the interpolated straight lines for both hyperplanes are decision tree optimal predictive model. We assume that hyperplane 1 is a newly target concept and hyperplane 2 is the previous target concept. A sample is positive (+) if it is above hyperplane 1; otherwise, it is negative (-). When we classify the samples in data set D by these predictive models, the samples in D could be in one of the following three categories: The samples which both predictive models make correct prediction, i.e. $C(X_i) = C_j(X_i) = y_i$, compose the first category. Obviously, these samples distribute in the top left and bottom right areas. The samples which both predictive models make wrong prediction, i.e. $(C(X_i) = C_j(X_i)) \neq y_i$, compose the second category. Due to concept drifting and the learning error, these samples can not be used to determine whether two predictive models are similar. The samples which satisfy $C(X_i) \neq C_j(X_i)$ compose the third category. These samples distribute in bottom left and top right areas. In algorithm 1, we estimate the similarity of two predictive models with the help of the count of samples in the first category and the third category.

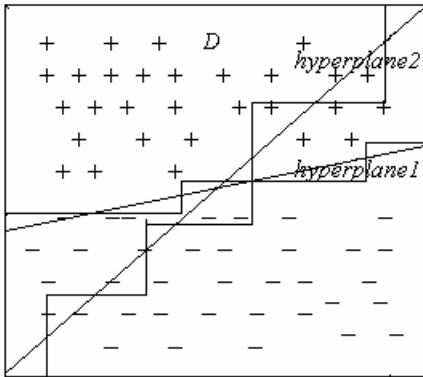


Fig. 1. Two different target concepts and their predictive models

Table 1. Transition matrix

Current state	Next state	
	1	2
1	0	2
2	1	0

3.4 Learning Concept Drifting Patterns

Given the model of streaming data and current concept, we can predict next concept by learning Markov chain. A Markov chain involves a sequence of states. For first-order Markov chain, the current state only depends on the immediate past state. For an example of first-order Markov chain, assuming the model of the streaming data is:

$(1,1,classifier_1,duration_1),(2,2,classifier_2,duration_2),(3,1,classifier_3,duration_3)(4,2,classifier_4,duration_4)$

A transition matrix can be built on this data model and be updated along the time. Table 1 shows the final status of this first-order transition matrix.

3.5 Learning Temporal Patterns

Here we represent our method for learning temporal patterns about concept drifting. The method of least squares can be used to estimating the quantitative trend between the independent variable and the dependent variable.

Suppose a data stream model is:

$(1,I_1,Classifier_1,Duration_1)(2,I_2,Classifier_2,Duration_2)\dots(i,I_i,Classifier_i,Duration_i)\dots(n,I_n,Classifier_n,Duration_n)$,

we write $\Omega = \{(i, I_i, Classifier_i, Duration_i) | i = 1, 2 \dots n\}$, $\mathfrak{R} = \{I | (i, I, Classifier_i, Duration_i) \in \Omega, 1 \leq i \leq n\}$,

then for $\forall I \in \mathfrak{R}$ we can define:

$$\Delta_I = \{(i, I) | 1 \leq i \leq n, I \in \mathfrak{R}, (i, I, Classifier_i, Duration_i) \in \Omega\} = \{(i_1, I), (i_2, I), \dots, (i_N, I)\},$$

where $N = |\Delta_I|$ and $i_1 < i_2 < \dots < i_N$.

Hence, we can get a sequence $(1, Duration_{i_1})(2, Duration_{i_2}) \dots (N, Duration_{i_N})$ from Δ_I , with the j -th element in this sequence $(j, Duration_{i_j})$ means the duration of the j -th time the concept I recurs in the data stream. After taking the basis function $(1, x, x^2, \dots, x^m), m < N$, we can get the normal equations:

$$\begin{bmatrix} \sum_{j=1}^N 1 & \sum_{j=1}^N j & \dots & \sum_{j=1}^N j^m \\ \sum_{j=1}^N j & \sum_{j=1}^N j^2 & \dots & \sum_{j=1}^N j^{m+1} \\ \dots & \dots & \dots & \dots \\ \sum_{j=1}^N j^m & \sum_{j=1}^N j^{m+1} & \dots & \sum_{j=1}^N j^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^N Duration_{i_j} \\ \sum_{j=1}^N j Duration_{i_j} \\ \vdots \\ \sum_{j=1}^N j^m Duration_{i_j} \end{bmatrix}. \tag{3}$$

The unknown coefficients a_0, a_1, \dots, a_m can be obtained by solving the above linear equations. Then, we can use

$$\phi_I^{Duration}(x) = a_0 + a_1x + \dots + a_mx^m. \tag{4}$$

to estimate the duration of the concept I .

By this method we can estimate the duration of a concept. When the duration of current concept will be over soon, we predict next concept by current concept. We classify oncoming sample using current concept and predicted next concept so as to find an approximate concept drifting point.

For example, suppose each sample of a data stream contains three attributes: $color \in \{green, blue, red\}$, $shape \in \{triangle, circle, rectangle\}$, and $size \in \{small, medium, large\}$ and the data stream consists of concept (A) “ $color = red \wedge size = small$ ” and concept (B) “ $color = green \vee shape = circle$ ”:

$$\frac{\dots(\text{green,triangle,medium},0.0),(\text{red,circle,small},1.0)|(\text{blue,triangle,large},0.0),(\text{blue,triangle,medium},0.0),(\text{green,circle,small},1.0)\dots}{(A):\text{color}=\text{red} \wedge \text{size}=\text{small} \qquad (B):\text{color}=\text{green} \vee \text{shape}=\text{cicle}}$$

Here, “|” stands for a sudden concept drift. In this example, the exact concept drifting point is very difficult to find out, because that the samples $(\text{blue,triangle,large},0.0)$ and $(\text{blue,triangle,medium},0.0)$, which come from concept B , satisfy concept A too, and that the sample $(\text{red,circle,small},1.0)$, which comes from the concept A , satisfies the concept B too.

Using concept A and B together to classify these samples, we can get a sequence: $\dots(1,0),(1,1),(1,1),(1,1),(0,1)\dots$. Here, the first element $(1,0)$ stands for that the sample $(\text{green,riangle,medium},0.0)$ satisfies concept A and does not satisfy concept B . What we can confirm is that the concept drifting point lies inside the sequence. So, we randomly select a point from the sequence as a concept drift point, and then an approximate concept drifting point could be found and saved to our data stream model.

Even an approximate concept drifting point can be found, we may observe a sudden drop in prediction accuracy unless we know the exact concept drifting point beforehand. Under this situation, we fit curves for each exact concept drifting point.

For a concept $I \in \mathfrak{R}$, and $\Delta_I = \{(i_1, I), (i_2, I), \dots, (i_N, I)\}$ where $N = |\Delta_I|$ and $i_1 < i_2 < \dots < i_N$, we can fit two curves by least squares to estimate the start time and end time for this concept.

The j -th element of sequence $(1, \sum_{i=1}^{i_1-1} \text{Duration}_i)(2, \sum_{i=1}^{i_2-1} \text{Duration}_i) \dots (N, \sum_{i=1}^{i_N-1} \text{Duration}_i)$ means the j -th start time of concept I in the data stream. And hence this sequence could be used to fit the curve ϕ_I^{start} , which is used for estimating the j' -th start time of concept I , $j' > N$.

The j -th element of sequence $(1, \sum_{i=1}^{i_1} \text{Duration}_i)(2, \sum_{i=1}^{i_2} \text{Duration}_i) \dots (N, \sum_{i=1}^{i_N} \text{Duration}_i)$ means the j -th end time of concept I in the data stream. And hence this sequence could be used to fit the curve ϕ_I^{end} , which is used for estimating the j' -th end time of concept I , $j' > N$.

In case the curve conflicts with the current concept drifting situation, we should fix the curve again by adding the concept drifting point, which cause the confliction, into the curve. In this situation, a simple heuristic method can result in a quick convergence: we prefer to use up-to-date elements in the data stream model to fit the curve. If the curve does not conflict with the fact, then the corresponding counters should be increased. The curve could be used for predicting concept drifting after all these counters exceed the predefined threshold.

4 Recognizing and Treating Recurring Contexts

Our method, which can process recurring contexts and their duration efficiently and effectively, is summarized in algorithm 2. At each time stamp, a classifier could be returned.

```

Input: WinSize is the threshold for a sliding window
       data stream stream=  $\{X_i, y_i\}_0^n$ 
       accuracy threshold  $\theta$  for estimate concept drifting
Output: a classifier
begin
Classifier=TrainAClassifier(stream.get(0));
HavePredictedResult=False;
PatternStable=False;
FOR i From 1 To n{
  Inst=stream.get(i);
  IF (HavePredictedResult==False) {
    SlidingWindow ← ClassifyAndSave (Classifier, Inst);
    Classifier=Update (Classifier, Inst);
    Counter++;
    IF ((SlidingWindow.Size==WinSize)&&
        (SlidingWindow.ErrorRate> $\theta$ )) {
      DataStreamModel ← SaveAnElement (Classifier,
        Counter-WinSize, SlidingWindow);
      PredictedResult=
        Predict (DataStreamModel, Classifier);
      Counter=WinSize;
      IF (PredictedResult.Size==0)
        Classifier=TrainClassifier (SlidingWindow);
      ELSE Classifier=PredictedResult.Classifier;
      HavePredictedResult=True;
    }
    ELSE IF (SlidingWindow.Size==WinSize)
      Moving SlidingWindow;
  }
  ELSE IF ((HavePredictedResult==True)&&
    (PatternStable==False)) {
    IF (Counter<=
      PredictedResult.ConceptDuration-WinSize) {
      Counter++;
      Classifier=Update (Classifier, Inst);
      IF (Counter==
        PredictedResult.ConceptDuration-WinSize) {
        TempPredictedResult=
          Predict (DataStreamModel, Classifier);
        IF (TempPredictedResult.Size==0)
          HavePredictedResult=False;
        Else{
          NextClassifier=
            TempPredictedResult.Classifier;
          BuildPatterns (DataStreamModel,

```

```

        Classifier,NextClassifier);
    }
}
ELSE{
    IF(Classify(Classifier,Inst)==False)&&
    (Classify(NextClassifier,Inst)==True){
        DataStreamModel ←
        SaveAnElement(Classifier,Counter);
        Counter=1;
        PredictedResult=TempPredictedResult;
        Classifier=NextClassifier;
        IF(TestPatterns(List)==True)
            PatternStable=True;
        ELSE
            FixPattern(List);
    }
}
ELSE IF(PatternStable==True){
    Classifier=UsePattern(i);
    IF(CheckPatterns(i,Inst,Classifier)=False)
        PatternStable=False;
}
Return Classifier;
}
end.

```

Algorithm 2. Recognizing and treating recurring contexts

RTRC has three stages. In the first stage, if there are no recurring contexts, we use sliding window approach to detect a concept drifting and learn new concept from scratch after a concept drifting, which often means a long delay after concept drifting and slow convergence to new concept. In the second stage, if there is enough information for predicting next concept and its duration, RTRC replaces current concept with the predicted concept at the approximate concept drifting point. In the third stage, as long as the curves are stable, we classify the oncoming samples according to these curves. It can be seen that RTRC can transform from one stage to another depending on the inputting data stream, and that RTRC can utilize experience from previous learning by reusing saved concepts, and induce stable patterns about concept drifting and concept duration.

Generally speaking, a data stream contains limited contexts. Under this condition, the time complexity of algorithm 2 is $O(n)$. For the extreme condition, when each sample of the data stream comes from different contexts, the time complexity is $O(n^2)$. The space complexity of our algorithm is a constant.

5 Empirical Study and Results

5.1 STAGGER Concepts Dataset

In this section, we present experimental results for weighted Bagging [11], VFDT [4] and our RTRC. We evaluated these systems on the *STAGGER Concepts* [3,7,10]

dataset, a standard benchmark for evaluating how learners cope with drifting concepts. The dataset can simulate the scenario of recurring context. Each sample consists of three attributes: $color \in \{green, blue, red\}$, $shape \in \{triangle, circle, rectangle\}$, and $size = \{small, medium, large\}$. There are three underlying concepts 1, $color = red \wedge size = small$; 2, $color = green \vee shape = circle$; and 3, $size = medium \vee size = large$.

Training samples were generated randomly according to the hidden concept, and the predictive performance was tested on 100 test samples which were also generated randomly, according to the same underlying concept. The underlying concept was made to change in the cyclic order, say, $(1,1,C_1,D_1)$, $(2,2,C_2,D_2)$, $(3,3,C_3,D_3)$, $(4,1,C_4,D_4)$, $(5,2,C_5,D_5)$, $(6,3,C_6,D_6) \dots$. Here $D_1 = D_4 = \dots = D_{3n-2}$, $D_2 = D_5 = \dots = D_{3n-1}$, $D_3 = D_6 = \dots = D_{3n}$, $n = 1, 2 \dots$ $D_1 \neq D_2$, $D_2 \neq D_3$, and $D_1 \neq D_3$. Thus, we created a situation of recurring concepts. This experiment was repeated many times, with training data generated randomly each time with different D_1, D_2 and D_3 .

Figure 2 displays the result of typical individual run with $D_1 = 99, D_2 = 100$ and $D_3 = 101$ on the *STAGGER Concepts* dataset. Note that at the beginning (Here, there is no enough information for predicting next concept and its duration), RTRC learns each concept and its accuracy increase slowly. When a concept drifting happens, a sudden drop appears in predictive accuracy, and there is a long delay before the concept drifting can be detected. However, after RTRC has scanned enough samples, it can predict next concept before concept drifting ever since time stamp 601. In figure 3, we present performance comparison between RTRC and Weighted Bagging at each data block (about 30 samples). This figure shows that almost for each concept drifting point Weighted Bagging performs a dramatic drop in accuracy, while RTRC can learn a good predictive model at the 19-th data block, so that after this data block its predictive accuracy is not affected by concept drifting any more.

In figure 4, we compare RTRC with CVFDT. CVFDT always learns a concept from scratch upon trigger detection no matter whether this concept is a reappearance of an old one. And CVFDT do not try to use the temporal information, so a periodical dramatic drop in accuracy still can be observed. It is shown in figure 4 that RTRC performs much less drops in accuracy than CVFDT does.

From this experiment result we can make conclusion that RTRC is much better than Weighted Bagging and CVFDT on *STAGGER Concept* dataset.

5.2 Moving Hyperplane Dataset

We create synthetic data with drifting concepts based on a moving hyperplane for experiment. This *Moving Hyperplane* dataset is also widely used for experiment [2,3,10]. A hyperplane in a d -dimensional space is denoted by equation: $\sum_{i=1}^d w_i x_i = w_0$. Here each vector of variables $\langle x_1, x_2, \dots, x_d \rangle$ is a randomly generated sample and is uniformly distributed in the multidimensional space $[0,1]^d$. All the samples which satisfy $\sum_{i=1}^d w_i x_i \geq w_0$ are labeled positive and otherwise negative. The value of each coefficient w_i is continuously changed, as illustrated in figure 5, so that the hyperplane is gradually drifting in the space. The value of w_0 is always set

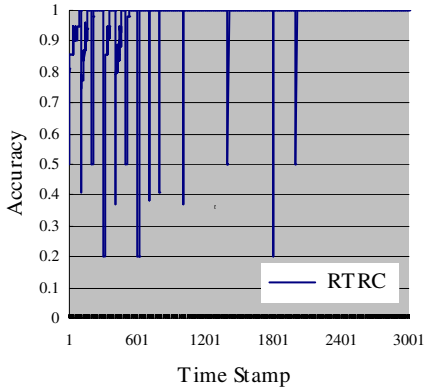


Fig. 2. Predictive accuracy for RTRC on *STAGGER Concepts* at each time stamp

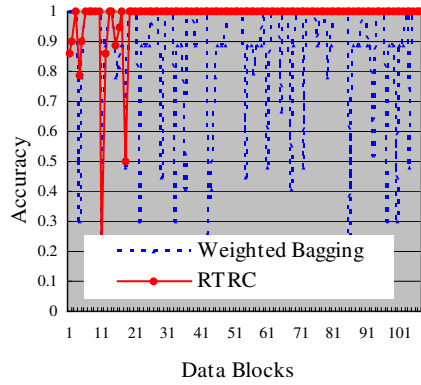


Fig. 3. Predictive accuracy for RTRC and Weighted Bagging on *STAGGER Concepts* at each data block

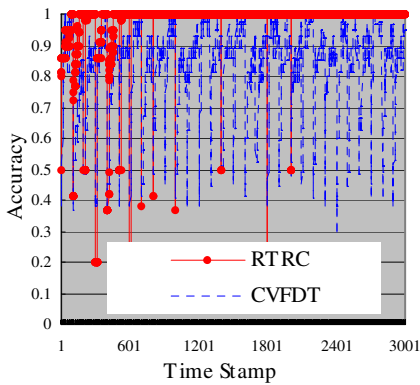


Fig. 4. Predictive accuracy for RTRC and CVFDT on *STAGGER Concepts* at each time stamp

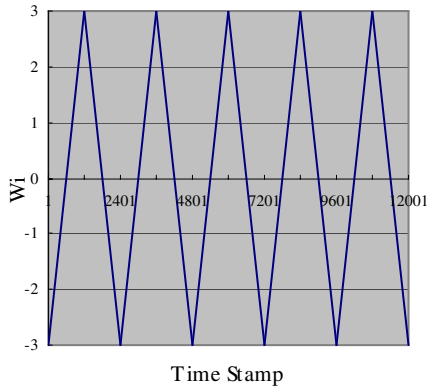


Fig. 5. To simulate concept drifting by a moving hyperplane

as $w_0 = \frac{1}{2} \sum_{i=1}^d w_i$, so that roughly half of the samples are positive, and the other half are negative. In this experiment, we set $w_i \in [-3, 3]$ and its changing rate is 0.005 per sample.

Because CVFDT does not work with continuous attributes, we do not compare RTRC and CVFDT on *Moving Hyperplane*. Figure 6 presents the result of RTRC compared with Weighted Bagging at each data block. Figure 7 shows the results for RTRC on *Moving Hyperplane*. This experiment reveals that algorithm 1 measures all hyperplane with $w_i \in [-3, 0]$ as equivalent and all hyperplane with $w_i \in (0, 3]$ as equivalent. As a result, a concept sequence of 1,2,1,2,1,2... is learned. Weighted

Bagging has a low predictive accuracy steadily, while RTRC gradually increase its accuracy. By the time stamp 8400, stable predictive model was discovered and used to predict concept drifting points. So, after this time stamp, the prediction accuracy is not affected by concept drifting any more.

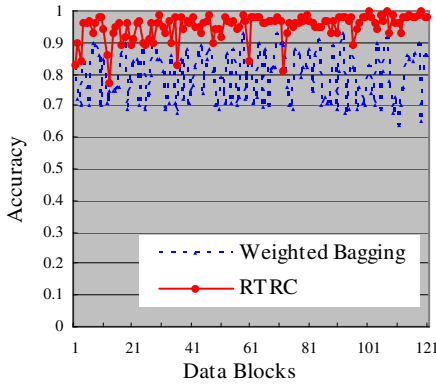


Fig. 6. Predictive accuracy for RTRC and Weighted Bagging on *Moving Hyperplane* at each data block

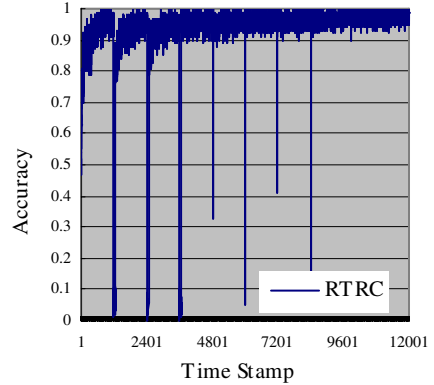


Fig. 7. Predictive accuracy for RTRC on the *Moving Hyperplane* at each time stamp

6 Conclusion

Tracking concept drifting is important for many applications. In this paper, we present our RTRC method, which can save old concepts and their durations so to learn prediction models. When there is recurring context, we can use Markov chain and least-square method to predict what the next concept is and when the concept will drift. Hence we can use appropriate concept at right time. It can be observed from our experiments that the proposed methods achieve higher predictive accuracies than Weighted Bagging and CVFDT. And furthermore, once scanned enough samples from data stream, RTRC can achieve high accuracy without any dramatic dropping of accuracy any more.

Traditional researches on data stream mining only put emphasis on building accuracy classifiers, while not enough focuses have been given to predict what the next concept is and when the concept will drift. By learning from this information, RTRC partially realize the target, “the prediction accuracy is not correlated to concept drifting” [1].

References

1. Wei Fan: Systematic Data Selection to Mine Concept-Drifting Data Streams. In proceeding of the conference KDD (2004) 128-137
2. F. Chu and C. Zaniolo: Fast and Light Boosting for Adaptive Mining of Data Streams. In proceeding of the conference PAKDD (2004)

3. Kolter J., Maloof M.: Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift. In proceeding of the conference ICDM (2003)
4. G.Hulten, L.Spencer, P. Domingos: Mining Time-Changing Data Streams. In proceeding of the conference ACM SIGKDD (2001)
5. W.Street, Y. Kim: A Streaming Ensemble Algorithm(sea) for Large-Scale Classification. In proceeding of the conference SIGKDD (2001)
6. Xingquan Zhu, Xindong Wu, Ying Yang: Effective Classification of Noisy Data Streams with Attribute-Oriented Dynamic Classifier Selection. In proceeding of the conference ICDM (2004)
7. G.Widmer and M. Kubat: Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning* (1996) 69-101
8. Salganicoff M.: Tolerating Concept and Sampling Shift in Lazy Learning Using Prediction Error Context Switching. *AI Review, Special Issue on Lazy Learning*, 11(1-5) (1997) 133-155
9. Harries M., Sammut C., Horn K.: Extracting Hidden Context, *Matching Learning* 32 (2) (1998) 101-126
10. Ying Yang, Xindong Wu, Xingquan Zhu: Combining Proactive and Reactive Predictions for Data Streams. In proceeding of the conference KDD (2005)
11. H.Wang, Wei Fan, P. Yu, J.Han: Mining Concept-Drifting Data Streams Using Ensemble Classifiers. In proceeding of the conference SIGKDD (2003)
12. Pedro Domingos, Geoff Hulten: Mining High-Speed Data Streams. In proceeding of the conference KDD (2000) 71-80
13. R. Jin, G. Agrawal: Efficient Decision Tree Construction on Streaming Data, In proceeding of the conference ACM SIGKDD (2003)
14. G. John, P. Langley: Estimating Continuous Distributions in Bayesian Classifiers. In proceeding of the conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA (1995) 338-345

Retraction Note to: Structural Analysis and Mathematical Methods for Destabilizing Terrorist Networks Using Investigative Data Mining

Nasrullah Memon and Henrik Legind Larsen

Software Intelligence Security Research Center
Department of Software, Electronics and Media Technology
Aalborg Universitet Esbjerg
Niels Bohrs Vej 8, 6700 Esbjerg Denmark
{nasrullah, legind}@cs.aau.dk

X. Li, O.R. Zaiane, and Z. Li (Eds.): ADMA 2006, LNAI 4093, pp. 1037 – 1048, 2006.
© Springer-Verlag Berlin Heidelberg 2006

DOI 10.1007/11811305_120

The publisher regrets to announce that the following chapter entitled “Structural Analysis and Mathematical Methods for Destabilizing Terrorist Networks Using Investigative Data Mining” by Nasrullah Memon and Henrik Legind Larsen, pp. 1037–1048, published in LNAI 4093 has been retracted. This chapter contains a large amount of reused and uncited material that was not published within quotation marks.

The original online version for this chapter can be found at
http://dx.doi.org/10.1007/11811305_113

Author Index

- Alfred, Rayner 889
Alhajj, Reda 64
Alonso, Rafael 1017
Argamon, Shlomo 485
- Baek, Jangsun 574
Bae, Sung Min 372
Bao, Shenghua 798
Bao, Yidan 1000
Bauer, Thomas 135
Begg, Rezaul 296
Bhumiratana, Sakarindr 827
- Cai, Zixing 72
Cao, Longbing 582
Cao, Yongcun 985
Cao, Yongfeng 157
Castelli, Eric 845
Castillo, Gladys 42
Cen, Haiyan 1000
Chaijaruwanich, Jeerayut 827, 835
Chang, Chia-Wen 693
Chang, Lei 761
Charoenkwan, Phasit 827
Chen, Arbee L.P. 1077
Chen, Changjia 934
Chen, Hong 1085
Chen, Huowang 864
Chen, Jianer 247
Chen, Leiting 626
Chen, Lihui 328
Chen, Ping 606
Chen, Yao 533
Cheng, Xiaochun 644
Chowdhury, Abdur 485
- Da Costa, David 416
Dai, Honghua 213
Dan, Hongwei 92
Degemmis, M. 661
Deng, Zhi-Hong 56
Dong, Xiangjun 100
Dong, ZhaoYang 769
Du, Nan 606
- Duan, Lei 239
Duan, Zhuohua 72
- Feng, Yi 110, 118
Fu, Ada Wai-chee 31
- Gama, João 42
Gan, Guojun 271
Gao, Wen 173, 636
Ge, Dingfei 143
Gonzalez, Hector 1
Goswami, Suchandra 80
Greene, Glenna 1017
Gu, Haijie 723
Gu, Ping 652
Gu, Zhi-Min 509
Guo, Gongde 165
Guo, Wenzhong 388
- Ha, Sung Ho 372
Hagenbuchner, M. 19
Han, Dingyi 790
Han, Jiawei 1
Han, Jie 790
Han, Xiqing 100
Hardoon, David R. 681
He, Tingting 1008
He, Xiping 652
He, Yong 525, 1000
He, Zhi 473
He, Zongyuan 741
Honavar, Vasant 465
Hong, ManPyo 1025
Hou, Jian-rong 782
Hou, Ruilian 100
Hsueh, Sue-Chen 693
Hu, Hesuan 380
Hu, Tianming 284
Huang, Chong 636
Huang, Houkuan 473
Huang, Min 1000
Huang, Pei 782
Huang, Ronghuai 644
Huang, Tiejun 173, 636

- Huang, Zhenhua 927
 Hui, Xiao-Feng 947

 Jeong, Kyeong Ja 1049
 Jia, Jinkang 934
 Jia, Sen 731
 Jiang, Maojin 485
 Jiang, Yun 1094
 Jiao, Licheng 967
 Jin, Yang 702
 Joo, Jinu 465

 Kasabov, Nikola 197
 Kaya, Mehmet 64
 Kazakov, Dimitar 889
 Keogh, Eamonn 31
 Khamphachua, Jamlong 835
 Kim, Jong-Min 911
 Klabjan, Diego 1
 Kong, Fansheng 308, 348
 Kumar, R. Pradeep 977

 Law, Rob 135
 Lee, Bum Ju 819
 Lee, Heon Gyu 819
 Lee, Jae Sik 959
 Lee, Jin Chun 959
 Lee, Keunjoon 465
 Lee, Sanghack 501
 Legind Larsen, Henrik 1037
 Leng, Ming 493
 Leung, Oscar Tat-Wing 31
 Lhee, Kyung-suk 1025
 Li, Anbo 263
 Li, Cuiping 1085
 Li, Gang 213, 594
 Li, Hong 247
 Li, Hua 1017
 Li, Jianyu 396
 Li, Jin 881
 Li, Qing 967
 Li, Rui 798
 Li, Shaozi 388
 Li, Siwei 1008
 Li, Wanqing 550
 Li, Xiaolei 1
 Li, Xiaoli 525
 Li, Xiao-Lin 448
 Li, Xin 56
 Li, Yixiao 308, 348

 Li, Zhanhuai 542, 1094
 Li, Zhiwu 380
 Li, Zhiyong 749
 Lin, Chenxi 790
 Lin, Jessica 31
 Lin, Ming-Yen 693
 Ling, Guangjie 731
 Liu, Dayou 126
 Liu, Dong 594
 Liu, Liping 284
 Liu, Peide 364
 Liu, Peng 457
 Liu, Qihe 626
 Liu, Wanquan 550
 Liu, Yanbing 919
 Liu, Yuanjie 798
 Lopes, Heitor S. 871
 Lops, P. 661
 Lu, Jingli 223
 Lu, Xicheng 992
 Luo, Huilan 308, 348
 Luo, Siwei 396
 Lv, Yan 72
 Lv, Yanping 388

 Ma, Tian Min 197
 Ma, Yutao 809
 Man, Yi 644
 Manorat, Aompilai 827
 Mao, Keming 711
 Memon, Nasrullah 1037
 Meng, Jun 741, 749
 Min, Fan 626
 Mo, Li 205
 Morales, Rafael 899

 Nagabhushan, P. 977
 Nagatomi, Ryoichi 1057
 Nasraoui, Olfa 80
 Neagu, Daniel 165
 Nguyen, Cong Phuong 845
 Noh, Kiyong 819
 Núñez, Marlon 899

 Oh, Sanghun 279
 Orgun, Mehmet A. 316
 Oyanagi, Shigeru 436

 Palittapongarpim, Prasit 835
 Pan, Ding 618

- Pan, Yantao 992
 Pan, Yunhe 92
 Park, Kyu-Sik 279
 Park, Sang Chan 566
 Park, Sungyong 501
 Patist, Jan Peter 517
 Peng, Wei 992
 Peng, Yonghong 809
 Pham, Ngoc Yen 845
 Prasitwattanaseree, Sukon 827, 835

 Qian, Yuntao 533, 731
 Qu, Chao 284
 Qu, Guozhong 1008
 Qu, Xiao 143
 Qu, Zhong 255
 Quang, Tran Minh 436

 Ren, Han 1008
 Roh, Tae Hyup 424
 Rong, Gang 723
 Ryu, Keun Ho 819

 Sami, Ashkan 856, 1057
 Saunders, Craig 681
 Schenkel, Peter 550
 Semeraro, G. 661
 Shawe-Taylor, John 681
 Shin, Miyoung 189
 Shin, Moon Sun 1049
 Shon, Ho-Sun 819
 Sidhu, Kush 485
 Son, Young Sook 574
 Song, Qun 197
 Sperduti, A. 19
 Suh, Jong Hwan 566
 Sun, Dan 741, 749
 Sun, Fengrong 100
 Sun, Jiancheng 150
 Sun, Jie 947
 Sun, Jun 340
 Sung, Sam Yuan 284
 Szedmak, Sandor 681

 Takahashi, Makoto 1057
 Tang, Changjie 239
 Tang, Guizhong 673
 Tang, Hai-Yan 448
 Tang, Shiwei 56, 761
 Tariq, Usman 1025

 Tian, Shengfeng 473
 Tian, Yonghong 173, 636
 Tjhi, William-Chandra 328
 Tokuyama, Takeshi 1057
 Tse, Tony 135
 Tsoi, A.C. 19
 Tu, Xinhui 1008
 Tu, Yiqing 213, 594

 Venturini, Gilles 416

 Wan, Li 606
 Wang, Aiping 263
 Wang, Anrong 380
 Wang, Change 881
 Wang, Chunshan 110, 118
 Wang, Fangshi 181
 Wang, Guoren 356, 711
 Wang, Hao 157
 Wang, Ji 864
 Wang, Jin 798
 Wang, Jinlong 92
 Wang, Lian 255
 Wang, Menghao 919
 Wang, Min 263
 Wang, Shanshan 165
 Wang, Shuang-Cheng 448
 Wang, Shuliang 1065
 Wang, Shulin 864
 Wang, Shuqin 126
 Wang, Tengjiao 761
 Wang, Wei 927
 Wang, Yong 1094
 Warit, Saradee 835
 Webb, Geoffrey I. 223
 Weber, Karin 135
 Wei, Dagang 239
 Wei, Jinmao 126
 Weinert, Wagner R. 871
 Wu, Bin 606
 Wu, Jianhong 271
 Wu, Naijun 457
 Wu, Tzu-Chiang 1077
 Wu, Wei 205
 Wu, Zhaohui 110, 118

 Xiang, Yao 247
 Xie, Zhi-jun 1085
 Xie, Zhipeng 558
 Xu, Ciwen 985

- Xu, Congfu 92
 Xu, De 181
 Xu, Guandong 296
 Xu, Guangyu 356
 Xu, Hongli 181
 Xu, Wenbo 340
 Xu, Zhao 769
 Xu, Zhengming 388
 Xue, Gui-Rong 790

 Yamazaki, Katsuhiko 436
 Yan, Guanghui 542
 Yang, Bo 404
 Yang, Dongqing 761
 Yang, Hwan-Seok 911
 Yang, Jihoon 465, 501
 Yang, Ming-Hua 509
 Yang, Shuzhong 396
 Yang, Wen 157
 Yang, Ying 223
 Yang, Zijiang 271
 Ye, Bin 340
 Ye, Qi 606
 Ye, Yangdong 594
 Yin, Junjie 457
 Yin, Linjun 711
 Yin, Ying 356
 Yoon, Won-Jung 279
 You, Junping 126
 Yu, Songnian 493
 Yu, Yong 790, 798
 Yuan, Haning 1065
 Yuan, Liu 542

 Zhang, Boyun 864
 Zhang, Haijian 157
 Zhang, Huan 239
 Zhang, Huaxiang 364
 Zhang, Jianzhong 626
 Zhang, Kang 316
 Zhang, Ke-Bing 316
 Zhang, Longbo 1094
 Zhang, Taiyi 150
 Zhang, Tianqing 239
 Zhang, Wei 457
 Zhang, Yanchun 296
 Zhang, Yang 1094
 Zhang, Yong 1008
 Zhao, Hui 782
 Zhao, Jun Hua 769
 Zhao, Mingying 881
 Zhao, Rui 247
 Zhao, Shuguang 881
 Zhao, Yaqin 673
 Zhao, Yue 985
 Zhao, Yuhai 356, 711
 Zheng, Yan 644
 Zhou, Changle 388
 Zhou, Jianzhong 205
 Zhou, Xianzhong 673
 Zhou, Yatong 150
 Zhou, Zhongmei 110, 118
 Zhu, Chengjun 205
 Zhu, Jiaxian 457
 Zhu, Qingsheng 652
 Zuo, Wanli 702