Zhi-Hua Zhou
Hang Li
Qiang Yang (Eds.)

LNAI 4426

Advances in Knowledge Discovery and Data Mining

11th Pacific-Asia Conference, PAKDD 2007
Nanjing, China, May 2007
Proceedings

Springer

Lecture Notes in Artificial Intelligence 4426

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Zhi-Hua Zhou Hang Li
Qiang Yang (Eds.)

Advances in Knowledge Discovery and Data Mining

11th Pacific-Asia Conference, PAKDD 2007
Nanjing, China, May 22-25, 2007
Proceedings

Springer

Series Editors

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

Volume Editors

Zhi-Hua Zhou
Nanjing University
National Lab for Novel Software Technology
Hankou Road 22, Nanjing 210093, China
E-mail: zhouzh@nju.edu.cn

Hang Li
Microsoft Research Asia
No. 49 Zhichun Road, Haidian District, Beijing, China 100080
E-mail: hangli@microsoft.com

Qiang Yang
Hong Kong University of Science and Technology
Department of Computer Science and Engineering
Clearwater Bay, Kowloon, Hong Kong, China
E-mail: qyang@cs.ust.hk

Library of Congress Control Number: 2007923867

CR Subject Classification (1998): I.2, H.2.8, H.3, H.5.1, G.3, J.1, K.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-71700-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-71700-3 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12042982 06/3180 5 4 3 2 1 0

Preface

The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) has been held every year since 1997. This year, the 11th in the series (PAKDD 2007), was held at Nanjing, China, May 22–25, 2007. PAKDD is a leading international conference in the area of data mining. It provides an international forum for researchers and industry practitioners to share their new ideas, original research results and practical development experiences from all KDD-related areas including data mining, machine learning, databases, statistics, data warehousing, data visualization, automatic scientific discovery, knowledge acquisition and knowledge-based systems.

This year we received a record number of submissions. We received 730 research papers from 29 countries and regions in Asia, Australia, North America, South America, Europe and Africa. The submitted papers went through a rigorous reviewing process. Every submission except very few was reviewed by three reviewers. Moreover, for the first time, PAKDD 2007 introduced a procedure of having an area chair supervise the review process of every submission. Thus, most submissions were reviewed by four experts. The Program Committee members were deeply involved in a highly engaging selection process with discussions among reviewers and area chairs. When necessary, additional expert reviews were sought. As a result, a highly selective few were chosen to be presented at the conference, including only 34 (4.66%) regular papers and 92 (12.6%) short papers in these proceedings.

The PAKDD 2007 program also included four workshops. They were a workshop on Data Mining for Biomedical Applications (BioDM 2007), a workshop on Data Mining for Business (DMBiz 2007), a workshop on High-Performance Data Mining and Applications (HPDMA 2007) and a workshop on Service, Security and Its Data Management Technologies in Ubi-Com (SSDU 2007). A data mining competition under the PAKDD flag was also organized for the second time after the first competition that was held in PAKDD 2006.

PAKDD 2007 would not have been successful without the support of many people and organizations. We wish to thank the members of the Steering Committee for their invaluable suggestions and support throughout the organization process. We are indebted to the area chairs, Program Committee members and external reviewers for their effort and engagement in providing a rich and rigorous scientific program for PAKDD 2007. We wish to express our gratitude to our General Workshop Chair Takashi Washio for selecting and coordinating the exciting workshops, to the Tutorial and PAKDD School Chair Graham Williams for coordinating the fruitful tutorials and school lecturers, to the Industrial Track Chair Joshua Z. Huang for handling industrial track papers, to the PAKDD Competition Chair Nathaniel Noriel for organizing the PAKDD Competition and to the distinguished keynote speakers and tutorial presenters for their

wonderful talks and lectures. We are also grateful to the Local Arrangement Chairs Yang Gao and Xianglin Fei as well as the Local Organizing Committee, whose great effort ensured the success of the conference.

We greatly appreciate the support from various institutions. The conference was organized by the LAMDA Group of Nanjing University, Nanjing, China, in cooperation with Nanjing University of Aeronautics and Astronautics, the Japanese Society for Artificial Intelligence, and the Singapore Institute of Statistics. It was sponsored by the National Natural Science Foundation of China (NSFC), Microsoft AdCenter Labs, NEC Labs China, Microsoft Research Asia (MSRA), Salford Systems and K.C. Wong Education Foundation.

We also want to thank all authors and all conference participants for their contribution and support. We hope all participants took this opportunity to share and exchange ideas with one another and enjoyed PAKDD 2007.

January 2007

Zhi-Hua Zhou
Hang Li
Qiang Yang

Organization

PAKDD 2007 Conference Committee

Honorary Chairs

Hiroshi Motoda	AFOSR/AOARD and Osaka University, Japan (Life-long member)
Ruqian Lu	Chinese Academy of Sciences, China

General Chairs

David W.-L. Cheung	University of Hong Kong, China
Jian Lu	Nanjing University, China

Program Committee Chairs

Zhi-Hua Zhou	Nanjing University, China
Hang Li	Microsoft Research Asia, China
Qiang Yang	HKUST, China

Local Arrangements Chairs

Yang Gao	Nanjing University, China
Xianglin Fei	Nanjing University, China

Workshop Chair

Takashi Washio	Osaka University, Japan

Tutorial Chair and PAKDD School Chair

Graham Williams	University of Canberra, Australia

Industrial Chairs

Joshua Z. Huang	University of Hong Kong, China
Yunming Ye	Harbin Institute of Technology, China

PAKDD Competition Chair

Nathaniel Noriel	Singapore Institute of Statistics, Singapore

Publicity Chair

Kay Chen Tan	National University of Singapore, Singapore

Publication Chair

Yuan Jiang	Nanjing University, China

Web Chair

Yang Yu	Nanjing University, China

Publication and Registration Secretary

Xu-Ying Liu	Nanjing University, China

PAKDD 2007 Conference Steering Committee

Chairs

David W.-L. Cheung	University of Hong Kong, China
Rao Kotagiri	University of Melbourne, Australia

Treasurer

Graham Williams	University of Canberra, Australia

Members

Arbee L. P. Chen	National Chengchi University, Taiwan
Ming-Syan Chen	National Taiwan University, Taiwan
Tu Bao Ho	Japan Advanced Institute of Science and Technology, Japan
Masaru Kitsuregawa	Tokyo University, Japan
Huan Liu	Arizona State University, USA
Ee-Peng Lim	Nanyang Technological University, Singapore
Hiroshi Motoda	AFOSR/AOARD and Osaka University, Japan (Life-long member)
Jaideep Srivastava	University of Minnesota, USA
Takao Terano	Tokyo Institute of Technology, Japan
Kyu-Young Whang	Korea Advanced Institute of Science and Technology, Korea
Chengqi Zhang	University of Technology Sydney, Australia
Ning Zhong	Maebashi Institute of Technology, Japan

PAKDD 2007 Program Committee

Chairs

Zhi-Hua Zhou	Nanjing University, China
Hang Li	Microsoft Research Asia, China
Qiang Yang	HKUST, China

Area Chairs

Naoki Abe	IBM T.J. Watson Research Center, USA
Phoebe Chen	Deakin University, Australia

Zheng Chen	Microsoft Research Asia, China
Lee-Feng Chien	Academia Sinica, Taiwan
Eibe Frank	University of Waikato, New Zealand
João Gama	LIACC-University Porto, Portugal
Achim Hoffmann	The University of New South Wales, Australia
James Kwok	HKUST, China
Jinyan Li	Institute for Infocomm Research, Singapore
Charles X. Ling	University of Western Ontario, Canada
Huan Liu	Arizona State University, USA
Wee Keong Ng	Nanyang Technological University, Singapore
Jian Pei	Simon Fraser University, Canada
Fabio Roli	University of Cagliari, Italy
Takao Terano	Tokyo Institute of Technology, Japan
Kai Ming Ting	Monash University, Australia
Wei Wang	University of North Carolina at Chapel Hill, USA
Shichao Zhang	Guangxi Normal University, China
Zhongfei (Mark) Zhang	SUNY Binghamton, USA
Zijian Zheng	Microsoft, USA

Members

Gagan Agrawal	Vic Ciesielski
David Albrecht	Diane Cook
Aijun An	Alfredo Cuzzocrea
Vo Ngoc Anh	Dao-Qing Dai
Chid Apte	Honghua Dai
Hiroki Arimura	Gautam Das
Michael W. Berry	Tamraparni Dasu
Michael Berthold	Ian Davidson
Steffen Bickel	Luc De Raedt
Hendrik Blockeel	Xiaoyong Du
Jean-Francois Boulicaut	Tina Eliassi-Rad
Ulf Brefeld	Tapio Elomaa
Rui Camacho	Andries Engelbrecht
Longbing Cao	Floriana Esposito
Tru Hoang Cao	Johannes Fürnkranz
Sanjay Chawla	Wei Fan
Arbee Chen	Ada Waichee Fu
Ming-Syan Chen	Dragan Gamberger
Shu-Ching Chen	Junbin Gao
Songcan Chen	Rayid Ghani
Yixin Chen	Fosca Giannotti
William K. Cheung	Aristides Gionis
Yiu-ming Cheung	Bart Goethals
Sungzoon Cho	Dimitrios Gunopulos

Shyam Kumar Gupta
Jiawei Han
Hermann Helbig
Tu Bao Ho
Thu Hoang
Wynne Hsu
Xiaohua Hu
Jimmy Huang
Jin Huang
San-Yih Hwang
Sanjay Jain
Szymon Jaroszewicz
Daxin Jiang
Licheng Jiao
Huidong Jin
Rong Jin
Ruoming Jin
Alipio M. Jorge
Hillol Kargupta
George Karypis
Hiroyuki Kawano
Eamonn Keogh
Boonserm Kijsirikul
Myung Won Kim
Masaru Kitsuregawa
Rao Kotagiri
Marzena Kryszkiewicz
Ravi Kumar
Vipin Kumar
Wai Lam
Nada Lavrac
Jonathan Lawry
Sang Ho Lee
Vincent C S Lee
Wee Sun Lee
Yoon-Joon Lee
Tze Yun Leong
Chun-hung Li
Gang Li
Jianzhong Li
Tao Li
Xiao-Lin Li
Xue Li
Xuelong Li
Andrew Lim
Em-Peng Lim
Chih-Jen Lin
Xuemin Lin
Tie-Yan Liu
Xiaohui Liu
Woong-Kee Loh
Chang-Tien Lu
Jixin Ma
Marco Maggini
Yutaka Matsuo
Michael Mayo
Sameep Mehta
Wagner Meira Jr.
Xiaofeng Meng
Rosa Meo
Toshiro Minami
Pabitra Mitra
Yang-Sae Moon
Shinichi Morishita
Hiroshi Motoda
S. Muthu Muthukrishnan
Atsuyoshi Nakamura
Richi Nayak
Wilfred Ng
Hung Son Nguyen
Ngoc Thanh Nguyen
Zaiqing Nie
Kamal Nigam
Tadashi Nomoto
Zoran Obradovic
Takashi Okada
Salvatore Orlando
Matthew Otey
Satoshi Oyama
Sankar K. Pal
Yi Pan
Dino Pedreschi
Wen-Chih Peng
Yonghong Peng
Vincenzo Piuri
Joel Quinqueton
Naren Ramakrishnan
Sanjay Ranka
Patricia Riddle
Asim Roy

P. S. Sastry
Kenji Satou
Joern Schneidewind
Dou Shen
Yi-Dong Shen
Shengli Sheng
Daming Shi
Zhongzhi Shi
Akira Shimazu
Masashi Shimbo
Arno Siebes
Andrzej Skowron
Myra Spiliopoulou
Ashok N. Srivastava
Jaideep Srivastava
Aixin Sun
Einoshin Suzuki
Ah-Hwee Tan
Chew Lim Tan
Pang-Ning Tan
Zhaohui Tang
David Taniar
Theeramunkong Thanaruk
Hannu Toivonen
Luis Torgo
Ivor W. Tsang
Ah Chung Tsoi
Shusaku Tsumoto
Tomoyuki Uchida
Jeffrey D. Ullman
Benjamin W. Wah
Guoyin Wang
Haixun Wang
Hui Wang
Jason T. L. Wang
Lipo Wang
Wenjia Wang
Xufa Wang
Zhihai Wang
Graham Williams
Limsoon Wong
Rebecca Wright
Xindong Wu
Xintao Wu
Zhaohui Wu
Hui Xiong
Zhuoming Xu
Takehisa Yairi
Seiji Yamada
Chunsheng Yang
Hui Yang
Min Yao
Yiyu Yao
Jieping Ye
Wai Kiang Yeap
Dit-Yan Yeung
Jian Yin
Tetsuya Yoshida
Clement Yu
Hwanjo Yu
Jeffrey Xu Yu
Jian Yu
Philip S. Yu
Bianca Zadrozny
Mohammed Zaki
Bo Zhang
Changshui Zhang
Chengqi Zhang
Daoqiang Zhang
Du Zhang
Harry Zhang
Junping Zhang
Kang Zhang
Mengjie Zhang
Weixiong Zhang
Xuegong Zhang
Zili Zhang
Ning Zhong
Sheng Zhong
Aoying Zhou
Shuigeng Zhou
Yan Zhou
Xiaojin Zhu
Xingquan Zhu
Djamel Abdelakder Zighed

PAKDD 2007 External Reviewers

Osman Abul
Gulsah Altun
Bill Andreopoulos
Mafruz Zaman Ashrafi
Keven Ates
Francesca Barrientos
Jeremy Besson
Shyam Boriah
Toon Calders
Nicolas Cebron
Vineet Chaoji
Bernard Chen
Minmin Chen
Yun Chi
Massimo Coppola
Dayong Deng
Kevin Deronne
Nguyen Luong Dong
Jianfeng Du
Nuno Escudeiro
Mohamed Gaber
Feng Gao
Jing Gao
Ling Guo
Songtao Guo
Zhen Guo
Rohit Gupta
Sam Han
Jie Hao
Mohammad Al Hasan
Hongxing He
Chi-Wei Hsu
Haejin Hu
Ruoyun Huang
Tao Jiang
Chris Kauffman
Sung Jin Kim
Pushpa Kumar
Man Lan
Sangjun Lee
Sute Lei
Guoliang Li
Ming Li
Yong Li
Yuanxiang Li
Hong-Cheu Liu
Jiang Liu
Nianjun Liu
Xu-Ying Liu
Yang Liu
Ying Liu
Shijian Lu
Claudio Lucchese
Feifei Ma
Yuval Marom
Rodney Martin
Alessio Micheli
Ieva Mitasiunaite
Mirco Nanni
Minh Le Nguyen
Ann Nicholson
Zhengyu Niu
Masayuki Okabe
Takashi Onoda
Matt Otey
Nikunj Oza
Gaurav Pandey
Steve Pellicer
Raffaele Perego
Benjarath Phoophakdee
Marcin Pluciński
Huzefa Rangwala
Pedro Rodrigues
Saeed Salem
Ron Van Schyndel
Jouni Seppanen
Claudio Silvestri
Fabrizio Silvestri
Gyorgy Simon
Lay Ki Soon
John Stutz
Yasufumi Takama
Peter Tischer
Nikil Wale
Qian Wan
Raymond Wan

Dong Wang
Richard Watson
William Webber
Jun Xu
You Xu
Ghim Eng Yap
Hang Yu
Yang Yu
Huaifeng Zhang
Wei Zhang
Chunying Zhao
Min Zhao
Yanchang Zhao

Organized by

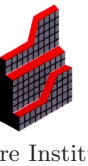
Nanjing University

LAMDA Group

In cooperation with

Nanjing University of Aeronautics and Astronautics

The Japanese Society for Artificial Intelligence



Singapore Institute of Statistics

Sponsored by

National Natural Science Foundation of China

Microsoft adCenter Labs

NEC Labs China

Microsoft Research Asia

Salford Systems

王宽诚
教育基金会

K.C. Wong Education Foundation

Table of Contents

Keynote Speeches

Regular Papers

Short Papers

Research Frontiers in Advanced Data Mining Technologies and Applications

Jiawei Han

Department of Computer Science, University of Illinois at Urbana-Champaign

Abstract. Research in data mining has two general directions: *theoretical foundations* and *advanced technologies and applications*. In this talk, we will focus on the research issues for advanced technologies and applications in data mining and discuss some recent progress in this direction, including (1) pattern mining, usage, and understanding, (2) information network analysis, (3) stream data mining, (4) mining moving object data, RFID data, and data from sensor networks, (5) spatiotemporal and multimedia data mining, (6) biological data mining, (7) text and Web mining, (8) data mining for software engineering and computer system analysis, and (9) data cube-oriented multidimensional online analytical processing.

Data mining, as the confluence of multiple intertwined disciplines, including *statistics, machine learning, pattern recognition, database systems, information retrieval*, *World-Wide Web*, and *many application domains*, has achieved great progress in the past decade [1]. Similar to many research fields, data mining has two general directions: *theoretical foundations* and *advanced technologies and applications*. Here we focus on *advanced technologies and applications in data mining* and discuss some recent progress in this direction. Notice that some popular research topics, such as privacy-preserving data mining, are not covered in the discussion for lack of space/time. Our discussion is organized into nine themes, and we briefly outline the current status and research problems in each theme.

1 Pattern Mining, Pattern Usage, and Pattern Understanding

Frequent pattern mining has been a focused theme in data mining research for over a decade. Abundant literature has been dedicated to this research and tremendous progress has been made, ranging from efficient and scalable algorithms for frequent itemset mining in transaction databases to numerous research frontiers, such as sequential pattern mining, structural pattern mining, correlation mining, associative classification, and frequent-pattern-based clustering, as well as their broad applications.

Recently, studies have proceeded to scalable methods for mining colossal patterns where the size of the patterns could be rather large so that the step-by-step growth using an Apriori-like approach does not work, methods for pattern compression, extraction of high-quality top-k patterns, and understanding patterns by context analysis and generation of semantic annotations. Moreover, frequent patterns have been used for effective

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1–5, 2007.

classification by top-k rule generation for long patterns and discriminative frequent pattern analysis. Frequent patterns have also been used for clustering of high-dimensional biological data. Scalable methods for mining long, approximate, compressed, and sophisticated patterns for advanced applications, such as biological sequences and networks, and the exploration of mined patterns for classification, clustering, correlation analysis, and pattern understanding will still be interesting topics in research.

2 Information Network Analysis

Google's PageRank algorithm has started a revolution on Internet search. However, since information network analysis covers many additional aspects and needs scalable and effective methods, the systematic study of this domain has just started, with many interesting issues to be explored. Information network analysis has broad applications, covering social and biological network analysis, computer network intrusion detection, software program analysis, terrorist network discovery, and Web analysis.

One interesting direction is to treat information network as graphs and further develop graph mining methods. Recent progress on graph mining and its associated structural pattern-based classification and clustering, graph indexing, and similarity search will play an important role in information network analysis. Moreover, since information networks often form huge, multidimensional heterogeneous graphs, mining noisy, approximate, and heterogeneous subgraphs based on different applications for the construction of application-specific networks with sophisticated structures will help information network analysis substantially. The discovery of the power law distribution of information networks and the rules on density evolution of information networks will help develop effective algorithms for network analysis. Finally, the study of link analysis, heterogeneous data integration, user-guided clustering, user-based network construction, will provide essential methodology for the in-depth study in this direction.

3 Stream Data Mining

Stream data refers to the data that flows into the system in vast volume, changing dynamically, possibly infinite, and containing multi-dimensional features. Such data cannot be stored in traditional database systems, and moreover, most systems may only be able to read the stream once in sequential order. This poses great challenges on effective mining of stream data.

With substantial research, progress has bee made on efficient methods for mining frequent patterns in data streams, multidimensional analysis of stream data (such as construction of stream cubes), stream data classification, stream clustering, stream outlier analysis, rare event detection, and so on. The general philosophy is to develop single-scan algorithms to collective information about stream data in tilted time windows, exploring micro-clustering, limited aggregation, and approximation. It is important to

explore new applications of stream data mining, *e.g.*, real-time detection of anomaly in computer networks, power-grid flow, and other stream data.

4 Mining Moving Object Data, RFID Data, and Data from Sensor Networks

With the popularity of sensor networks, GPS, cellular phones, other mobile devices, and RFID technology, tremendous amount of moving object data has been collected, calling for effective analysis. There are many new research issues on mining moving object data, RFID data, and data from sensor networks. For example, how to explore correlation and regularity to clean noisy sensor network and RFID data, how to integrate and construct data warehouses for such data, how to perform scalable mining for petabyte RFID data, how to find strange moving objects, how to cluster trajectory data, and so on. With time, location, moving direction, speed, as well as multidimensional semantics of moving object data, likely multi-dimensional data mining will play an essential role in this study.

5 Spatiotemporal and Multimedia Data Mining

The real world data is usually related to space, time, and in multimedia modes (*e.g.*, containing color, image, audio, and video). With the popularity of digital photos, audio DVDs, videos, YouTube, Internet-based map services, weather services, satellite images, digital earth, and many other forms of multimedia and spatiotemporal data, mining spatial, temporal, spatiotemporal, and multimedia data will become increasingly popular, with far-reaching implications. For example, mining satellite images may help detect forest fire, find unusual phenomena on earth, and predict hurricanes, weather patterns, and global warming trends.

Research in this domain needs the confluence of multiple disciplines including image processing, pattern recognition, parallel processing, and data mining. Automatic categorization of images and videos, classification of spatiotemporal data, finding frequent/sequential patterns and outliers, spatial collocation analysis, and many other tasks have been studied popularly. With the mounting in many applications, scalable analysis of spatiotemporal and multimedia data will be an important research frontier for a long time.

6 Biological Data Mining

With the fast progress of biological research and the accumulation of vast amount of biological data (especially, a great deal of it has been made available on the Web), biological data mining has become a very active field, including comparative genomics, evolution and phylogeny, biological databases and data integration, biological sequence analysis, biological network analysis, biological image analysis, biological literature analysis (*e.g.*, PubMed), and systems biology. This domain is largely overlapped with

bioinformatics but data mining researchers has been emphasizing on integrating biological databases with biological data integration, constructing biological data warehouses, analyzing biological networks, and developing various kinds of scalable bio-data mining algorithms.

Advances in biology, medicine, and bioinformatics provide data miners with abundant real data sets and a broad spectrum of challenging research problems. It is expected that an increasing number of data miners will devoted themselves to this domain and make contributions to the advances in both bioinformatics and data mining.

7 Text and Web Mining

The Web has become the ultimate information access and processing platform, housing not only billions of link-accessed "pages", containing textual data, multimedia data, and linkages, on the surface Web, but also query-accessed "databases" on the deep Web. With the advent of Web 2.0, there is an increasing amount of dynamic "workflow" emerging. With its penetrating deeply into our daily life and evolving into unlimited dynamic applications, the Web is central in our information infrastructure. Its virtually unlimited scope and scale render immense opportunities for data mining.

Text mining and information extraction have been applied not only to Web mining but also to the analysis of other kinds of semi-structured and unstructured information, such as digital libraries, biological information systems, business intelligence and customer relationship management, computer-aided instructions, and office automation systems.

There are lots of research issues in this domain, which takes the collaborative efforts of multiple disciplines, including information retrieval, databases, data mining, natural language processing, and machine learning. Some promising research topics include heterogeneous information integration, information extraction, personalized information agents, application-specific partial Web construction and mining, in-depth Web semantics analysis, and turning Web into relatively structured information-base.

8 Data Mining for Software Engineering and Computer System Analysis

Software program executions and computer system/network operations potentially generate huge amounts of data. Data mining can be performed on such data to monitor system status, improve system performance, isolate software bugs, detect software plagiarism, analyze computer system faults, uncover network intrusions, and recognize system malfunctions.

Data mining for software and system engineering can be partitioned into static analysis and dynamic/stream analysis, based on whether the system can collect traces beforehand for post-analysis or it must react at real time to handle online data. Different methods have been developed in this domain by integration and extension of the methods developed in machine learning, data mining, pattern recognition, and statistics. However, this is still a rich domain for data miners with further development of sophisticated, scalable, and real-time data mining methods.

9 Data Cube-Oriented Multidimensional Online Analytical Processing

Viewing and mining data in multidimensional space will substantially increase the power and flexibility of data analysis. Data cube computation and OLAP (online analytical processing) technologies developed in data warehouse have substantially increased the power of multidimensional analysis of large datasets. Besides traditional data cubes, there are recent studies on construction of regression cubes, prediction cubes, and other sophisticated statistics-oriented data cubes. Such multi-dimensional, especially high-dimensional, analysis tools will ensure data can be analyzed in hierarchical, multidimensional structures efficiently and flexibly at user's finger tips. This leads to the integration of online analytical processing with data mining, called OLAP mining.

We believe that OLAP mining will substantially enhance the power and flexibility of data analysis and bring the analysis methods derived from the research in machine learning, pattern recognition, and statistics into convenient analysis of massive data with hierarchical structures in multidimensional space. It is a promising research field that may lead to the popular adoption of data mining in information industry.

Reference

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques (2nd ed.). Morgan Kaufmann (2006)

Finding the Real Patterns

Geoffrey Webb

Clayton School of Information Technology,
P.O. Box 75 Monash University,
Victoria 3800, Australia

Abstract. Pattern discovery is one of the fundamental tasks in data mining. Pattern discovery typically explores a massive space of potential patterns to identify those that satisfy some user-specified criteria. This process entails a huge risk (in many cases a near certainty) that many patterns will be false discoveries. These are patterns that satisfy the specified criteria with respect to the sample data but do not satisfy those criteria with respect to the population from which those data are drawn. This talk discusses the problem of false discoveries, and presents techniques for avoiding them.

References

Webb, G.I.: Discovering significant rules. In Ungar, L., Craven, M., Gunopulos, D., Eliassi-Rad, T., eds.: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2006, New York, The Association for Computing Machinery (2006) 434 – 443.

Webb, G.I.: Discovering significant patterns. Machine Learning (in-press).

Biography

Geoff Webb holds a research chair in the Faculty of Information Technology at Monash University. Prior to Monash he held appointments at Griffith University and then Deakin University where he received a personal chair. His primary research areas are machine learning, data mining, and user modelling. He is widely known for his contribution to the debate about the application of Occam's razor in machine learning and for the development of numerous algorithms and techniques for machine learning, data mining and user modelling. His commercial data mining software, Magnum Opus, is marketed internationally by Rulequest Research. He is editor-in-chief of the highest impact data mining journal, Data Mining and Knowledge Discovery and a member of the editorial boards of Machine Learning, ACM Transactions on Knowledge Discovery in Data, and User Modeling and User-Adapted Interaction.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, p. 6, 2007.

Class Noise vs Attribute Noise: Their Impacts, Detection and Cleansing

Xindong Wu

Department of Computer Science,
University of Vermont,
33 Colchester Avenue, Burlington,
Vermont 05405, USA

Abstract. Noise handling is an essential task in data mining research and applications. There are three issues in dealing with noisy information sources: noise identification, noise profiling, and noise tolerant mining. During noise identification, erroneous data records are identified and ranked according to their impact or some predefined measures. Class noise and attribute noise can be distinguished at this stage. This identification allows the users to process their noisy data with different priorities based on the data properties. Noise profiling discovers patterns from previously identified errors that can be used to summarize and monitor these data errors. In noise tolerant mining, we integrate the noise profile information into data mining algorithms and boost their performances from the original noisy data. In this talk, I will present our existing and ongoing research efforts on these three issues.

Biography

Xindong Wu is a Professor and the Chair of the Department of Computer Science at the University of Vermont. He holds a PhD in Artificial Intelligence from the University of Edinburgh, Britain. His research interests include data mining, knowledge-based systems, and Web information exploration. He has published extensively in these areas in various journals and conferences, including IEEE TKDE, TPAMI, ACM TOIS, DMKD, KAIS, IJCAI, AAAI, ICML, KDD, ICDM, and WWW, as well as 12 books and conference proceedings.

Dr. Wu is the Editor-in-Chief of the IEEE Transactions on Knowledge and Data Engineering (by the IEEE Computer Society), the founder and current Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), an Honorary Editor-in-Chief of Knowledge and Information Systems (by Springer), and a Series Editor of the Springer Book Series on Advanced Information and Knowledge Processing (AI&KP). He was Program Committee Chair for ICDM '03 (the 2003 IEEE International Conference on Data Mining) and is Program Committee Co-Chair for KDD-07 (the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining). He is the 2004 ACM SIGKDD Service Award winner, the 2006 IEEE ICDM Outstanding Service Award winner, and a 2005 Chaired Professor in the Cheung Kong (or

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 7–8, 2007.

Yangtze River) Scholars Programme at the Hefei University of Technology sponsored by the Ministry of Education of China and the Li Ka Shing Foundation. He has been an invited/keynote speaker at numerous international conferences including IEEE EDOC'06, IEEE ICTAI'04, IEEE/WIC/ACM WI'04/IAT'04, SEKE 2002, and PADD-97.

Multi-modal and Multi-granular Learning*

Bo Zhang[1] and Ling Zhang[2]

[1] Dept. of Computer Science & Technology, Tsinghua University, Beijing, China
dcszb@tsinghua.edu.cn
[2] Institute of Artificial Intelligence, Anhui University, Hefei, Anhui, China
zling@ahu.edu.cn

Introduction

Under large scale data, machine learning becomes inefficient. In order to enhance its performances, new methodologies should be adopted. Taking video retrieval as an example, we discuss its two basic problems: representation and classification problems.

In spite of the form (text, image, speech, or video) of information there always exists a big semantic gap between its low-level feature based machine representations and the corresponding high-level concepts used by users, so the traditional single feature machine representation is not available under large scale data. To deal with the problem, the multi-modal and multi-granular representation is introduced. The reasons are the following. On the one hand, the representations from different modalities of the same video such as speech, image, and text may complement each other. On the other hand, a coarse representation of one modality, for example the global feature of an image such as color moment, color correlagram and global texture has a good robustness but a poor expressiveness. Contrarily, its fine representation, a representation with small grain-size, such as a pixel-based representation of an image has a good expressiveness but a poor robustness. Both expressiveness and robustness are needed in machine representation. Therefore, the multi-granular representation in one modality may solve the contradiction among them. We present a set of experimental results in image (and video) retrieval to show how the multi-modal and multi-granular representation improves the performances of machine learning.

By machine representation, a video (text, speech or image) will be translated into a vector (point) in a high dimensional feature space generally. Then information processing becomes a set of operations on a point set of the space. And the supervised machine learning becomes the classification of a set of points. By using multi-modal and multi-granular representation it means that the number of dimensionality of the feature space increases. It improves the learning performance but increases the computational complexity as well. This is so called dimensionality curse in machine learning. When the size of data increases, the problem becomes more serious. The general methodology used is the multi-classifier strategy. In the multi-classifier system, each classifier has its own classification criterion and input feature set. Firstly, the strategy is used to optimize the combination of the results from different

* Supported by the National Natural Science Foundation of China (Grant No. 60321002), the National Key Foundation R&D Project (Grant No. 2003CB317007, 2004CB318108).

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 9–10, 2007.

classifiers, that is, the information fusion problem. There have been many different information fusion approaches so far. Secondly, the multi-classifier is used in a hierarchical way, that is, multi-level classifiers. In the multi-level classifiers, a set of data is divided into a collection of subsets first and then each subset is further divided until the final result is obtained. By properly organizing the classifiers, the computational complexity can be reduced greatly. We will show some experimental results to verify the above statement. Thirdly, new efficient learning algorithms should be invented. Although there have been many learning algorithms recently the performance of most of them worsen when facing large scale data. We will present some learning algorithms that have rather good performances when dealing with the large scale data.

In conclusion, multi-modal and multi-granular learning is a new methodology inspired by human intelligence. The cognitive power in human learning consists of a set of resourceful strategies such as multi-modal, multi-granular representation, multi-feature fusion, and hierarchical structure, etc. In order to improve the machine learning, it should integrate both the cognitive power of human learning and the computational power of computers.

Biography

Mr. Bo Zhang graduated from Dept. of Automatic Control, Tsinghua University in 1958. He is now a professor of Computer Science and Technology Department, Tsinghua University, Beijing, China, the member of Chinese Academy of Sciences. His main research interests include artificial intelligence, robotics, intelligent control and pattern recognition. He has published about 150 papers and 3 monographs in these fields.

Mr. Ling Zhang graduated from Dept. of Mathematics, Nanjing University, Nanjing, China in 1960. He is now a professor of Dept. of Computer Science, Anhui University, Hefei, China and the director of Artificial Intelligence Institute, Anhui University. His main interests are artificial intelligence, machine learning, neural networks, genetic algorithms and computational intelligence. He has published more than 100 papers and 4 monographs in these fields.

Hierarchical Density-Based Clustering of Categorical Data and a Simplification

Bill Andreopoulos, Aijun An, and Xiaogang Wang

York University, Dept. of Computer Science and Engineering,
Toronto Ontario, M3J 1P3, Canada
{billa, aan}@cse.yorku.ca, stevenw@mathstat.yorku.ca

Abstract. A challenge involved in applying density-based clustering to categorical datasets is that the 'cube' of attribute values has no ordering defined. We propose the HIERDENC algorithm for *hierarchical density-based clustering of categorical data*. HIERDENC offers a basis for designing simpler clustering algorithms that balance the tradeoff of accuracy and speed. The characteristics of HIERDENC include: *(i)* it builds a hierarchy representing the underlying cluster structure of the categorical dataset, *(ii)* it minimizes the user-specified input parameters, *(iii)* it is insensitive to the order of object input, *(iv)* it can handle outliers. We evaluate HIERDENC on small-dimensional standard categorical datasets, on which it produces more accurate results than other algorithms. We present a faster simplification of HIERDENC called the MULIC algorithm. MULIC performs better than subspace clustering algorithms in terms of finding the multi-layered structure of special datasets.

1 Introduction

A growing number of clustering algorithms for categorical data have been proposed in recent years, along with interesting applications, such as partitioning large software systems and protein interaction data [6,13,29]. In the past, polynomial time approximation algorithms have been designed for NP-hard partitioning algorithms [9]. Moreover, it has recently been shown that the "curse of dimensionality" involving efficient searches for approximate nearest neighbors in a metric space can be dealt with, if and only if, we assume a bounded dimensionality [12,21]. Clearly, there are tradeoffs of efficiency and approximation involved in the design of categorical clustering algorithms. Ideally, a set of probabilistically justified goals for categorical clustering would serve as a framework for approximation algorithms [20,25]. This would allow designing and comparing categorical clustering algorithms on a more formal basis.

Our work is motivated by density-based clustering algorithms, such as CLIQUE [1], CLICKS [28], CACTUS [10], COOLCAT [5], DBSCAN [8], OPTICS [4], Chameleon [19], ROCK [14], DENCLUE [15], and others. Although most of these approaches are efficient and relatively accurate, we go beyond them and approach the problem from a different viewpoint. Many of these algorithms require the user to specify input parameters (with wrong parameter

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 11–22, 2007.

values resulting in a bad clustering), may return too many clusters or too many outliers, often have difficulty finding clusters within clusters or subspace clusters, or are sensitive to the order of object input [6,12,13,28]. We propose a categorical clustering algorithm that builds a hierarchy representing a dataset's entire underlying cluster structure, minimizes user-specified parameters, and is insensitive to object ordering. This offers to a user a dataset's cluster structure as a hierarchy, which is built independently of user-specified parameters or object ordering. A user can cut its branches and study the cluster structure at different levels of granularity, detect subclusters within clusters, and know the central densest area of each cluster. Although such an algorithm is slow, it inspires faster simplifications that are useful for finding the rich cluster structure of a dataset.

A categorical dataset with m attributes is viewed as an m-dimensional 'cube', offering a spatial density basis for clustering. A cell of the cube is mapped to the number of objects having values equal to its coordinates. Clusters in such a cube are regarded as *subspaces* of high object density and are separated by subspaces of low object density. Clustering the cube poses several challenges:

(i) Since there is no ordering of attribute values, the cube cells have no ordering either. The search for dense subspaces could have to consider several orderings of each dimension of the cube to identify the best clustering (unless all attributes have binary values).

(ii) The density of a subspace is often defined relative to a user-specified value, such as a radius. However, different radii are preferable for different subspaces of the cube [4]. In dense subspaces where no information should be missed, the search is more accurately done 'cell by cell' with a low radius of 1. In sparse subspaces a higher radius may be preferable to aggregate information. The cube search could start from a low radius and gradually move to higher radii. Although the term 'radius' is borrowed from geometrical analogies that assume circular constructs, we use the term in a looser way and it is not a Euclidean distance.

We present the *HIERDENC* algorithm for hierarchical density-based clustering of categorical data, that addresses the above challenges. HIERDENC clusters the m-dimensional *cube* representing the spatial density of a set of objects with m categorical attributes. To find its dense subspaces, HIERDENC considers an object's neighbors to be all objects that are within a *radius* of *maximum dissimilarity*. Object neighborhoods are insensitive to attribute or value ordering. Clusters start from the densest subspaces of the cube. Clusters expand outwards from a dense subspace, by connecting nearby dense subspaces. Figure 1 shows examples of creating and expanding clusters in a 3-d dataset. The radius is the maximum number of dimensions by which neighbors can differ.

We present the *MULIC algorithm*, which is a faster simplification of HIERDENC. MULIC is motivated by clustering of categorical datasets that have a *multi-layered* structure. For instance, in protein interaction data a cluster often has a center of proteins with similar interaction sets surrounded by peripheries of

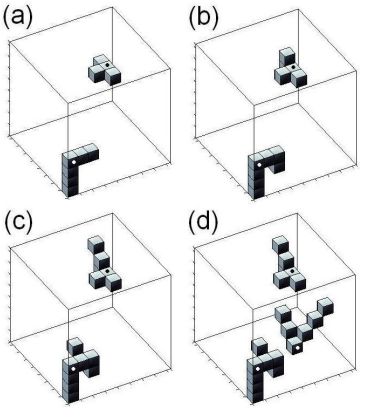


Fig. 1. A cluster is a dense subspace with a 'central' cell marked with a dot. (*a*) radius=1, two new clusters. (*b*) radius=1, clusters expand. (*c*) radius=2, clusters expand. (*d*) radius=2, one new cluster.

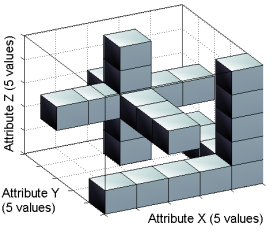


Fig. 2. Two HIERDENC 'hyper-cubes' in a 3D cube, for r=1

proteins with less similar interaction sets [7]. On such data, MULIC outperforms other algorithms that create a flat clustering.

This paper is organized as follows. Section 2 presents the HIERDENC algorithm. Section 3 describes the MULIC clustering algorithm and its relation to HIERDENC. Section 4 discusses the experiments. Section 5 concludes the paper.

2 HIERDENC Clustering

Basics. We are given a dataset of objects S (which might contain duplicates) with m categorical attributes, $X_1, \cdots, X_m$. Each attribute X_i has a domain D_i with a finite number of d_i possible values. The space S^m includes the collection of possibilities defined by the cross-product (or cartesian product) of the domains, $D_1 \times \cdots \times D_m$. This can also be viewed as an m-dimensional 'cube' with $\prod_{i=1}^{m} d_i$ cells (positions). A cell of the cube represents the unique logical intersection in a cube of one member from every dimension in the cube. The function λ maps a cell $\mathbf{x} = (x_1, \cdots, x_m) \in S^m$ to the nonnegative number of objects in S with all m attribute values equal to $(x_1, \cdots, x_m)$:

$$\lambda : \{(x_1, \cdots, x_m) \in S^m\} \rightarrow N.$$

We define the HIERDENC *hyper-cube* $C(\mathbf{x_0}, r) \subset S^m$, centered at cell $\mathbf{x_0}$ with radius r, as follows:

$$C(\mathbf{x_0}, r) = \{\mathbf{x} : \mathbf{x} \in S^m \; and \; dist(\mathbf{x}, \mathbf{x_0}) \leq r \; and \; \lambda(\mathbf{x}) > 0\}.$$

The $dist(\cdot)$ is a distance function. The *Hamming* distance is defined as follows:

$$HD(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m} \delta(x_i, y_i) \;\; where \;\; \delta(x_i, y_i) = \begin{cases} 1, & \text{if } x_i \neq y_i \\ 0, & \text{if } x_i = y_i \end{cases}$$

HD is viewed as the most natural way to represent distance in a categorical space. People have looked for other distance measures but HD has been widely accepted for categorical data and is commonly used in coding theory.

Figure 2 illustrates two HIERDENC hyper-cubes in a 3-dimensional cube. Since r=1, the hyper-cubes are visualized as 'crosses' in 3D and are not shown as actually having a cubic shape. A hyper-cube excludes cells for which λ returns 0. Normally, a hyper-cube will equal a subspace of S^m. A hyper-cube can not equal S^m, unless $r = m$ and $\forall \mathbf{x} \in S^m \; \lambda(\mathbf{x}) > 0$.

The *density* of a subspace $X \subset S^m$, where X could equal a hyper-cube $C(\mathbf{x_0}, r) \subset S^m$, involves the sum of λ evaluated over all cells of X:

$$density(X) = \sum_{\mathbf{c} \in X} \frac{\lambda(\mathbf{c})}{|S|}.$$

This density can also be viewed as the likelihood that a hyper-cube contains a random object from S, where $|S|$ is the size of S. HIERDENC seeks the densest hyper-cube $C(\mathbf{x_0}, r) \subset S^m$. This is the hyper-cube centered at $\mathbf{x_0}$ that has the maximum likelihood of containing a random object from S. The cell $\mathbf{x_0}$ is a member of the set $\{\mathbf{x} \in S^m : Max(P(\Omega \in C(\mathbf{x}, r)))\}$, where Ω is a discrete random variable that assumes a value from set S.

The *distance* between two clusters G_i and G_j is the distance between the nearest pair of their objects, defined as:

$$D(G_i, G_j) = \min\{dist(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in G_i \; and \; \mathbf{y} \in G_j\}.$$

Clusters G_i and G_j are *directly connected relative to* r if $D(G_i, G_j) \leq r$. Clusters A and B are *connected relative to* r if: A and B are directly connected relative to r, or if: there is a chain of clusters $C_1, \cdots, C_n$, $A = C_1$ and $B = C_n$, such that C_i and C_{i+1} are directly connected relative to r for all i such that $1 \leq i < n$.

HIERDENC Algorithm and Discussion. Figure 3 shows the HIERDENC clustering algorithm. The default initial value of radius r is 1. G_k represents the kth cluster formed. The remainder set, $R = \{\mathbf{x} : \mathbf{x} \in S^m \; and \; \mathbf{x} \notin G_i, i = 1, \cdots, k\}$, is the collection of unclustered cells after the formation of k clusters.

Step 1 retrieves the *densest* hyper-cube $C \subset S^m$ of radius r. Step 1 checks that the densest hyper-cube represents more than one object ($density(C(\mathbf{x_0}, r)) > \frac{1}{|S|}$), since otherwise the cluster will not expand, ending up with one object. If the hyper-cube represents zero or one object, then r is incremented. *Step 2* creates a new *leaf* cluster at level $r \geq 1$. Starting from an existing leaf cluster, *step 3* tries to move to the densest hyper-cube of radius r nearby. If a dense hyper-cube is found near the cluster, then in *step 4* the cluster expands by collecting the hyper-cube's cells. This is repeated for a cluster until no such connection

Input: space S^m.
Output: a hierarchy of clusters.
Method:

$r = 1$. //radius of hyper-cubes
$R = S^m$. //set of unclustered cells
$k = 0$. //number of leaf clusters
$k_r = 0$. //number of clusters at level r
$G_k = null$. //kth cluster
$U = null$. //set of hyper-cube centers

Step 1: Find $\mathbf{x_0} \in R$ such that $\max_{\mathbf{x_0}} density(C(\mathbf{x_0}, r))$.

If $density(C(\mathbf{x_0}, r)) \leq \frac{1}{|S|}$, then:
(1) $r = r + 1$.
(2) If $k_{r-1} > 1$, then:
(3) Merge clusters that are connected relative to r.
(4) $k_r = \#merged + \#unmerged\ clusters$.
(5) Repeat Step 1.

Step 2: Set $\mathbf{x_c} = \mathbf{x_0}$, $k = k + 1$, $G_k = C(\mathbf{x_c}, r)$, $R = R - C(\mathbf{x_c}, r)$ and $U = U \cup \{\mathbf{x_c}\}$.

Step 3: Find $\mathbf{x}^* \in C(\mathbf{x_c}, r)$ such that $\mathbf{x}^* \notin U$ and $\max_{\mathbf{x}^*} density(C(\mathbf{x}^*, r))$.

Step 4: If $density(C(\mathbf{x}^*, r)) > \frac{1}{|S|}$, then:
Update current cluster G_k: $G_k = G_k \cup C(\mathbf{x}^*, r)$.
Update R: $R = R - C(\mathbf{x}^*, r)$.
Update U: $U = U \cup \{\mathbf{x}^*\}$.
Re-set the new center: $\mathbf{x_c} = \mathbf{x}^*$.
Go to Step 3.
Otherwise, move to the next step.

Step 5: Set $k_r = k_r + 1$.
If $k_r > 1$, then execute lines $(3) - (4)$.
If $r < m$ and $density(R) > 1\%$, then go to Step 1.

Step 6: While $r < m$, execute lines $(1) - (4)$.

Fig. 3. The HIERDENC algorithm

can be made. New objects are clustered until $r = m$, or $density(R) \leq 1\%$ and the unclustered cells are identified as outliers (*step 5*). For many datasets, most objects are likely to be clustered long before $r = m$.

Initially $r = 1$ by default, since most datasets contain subsets of similar objects. Such subsets are used to initially identify dense hyper-cubes. When r is incremented, a special process *merges* clusters that are connected relative to r. Although the initial $r = 1$ value may result in many clusters, similar clusters are

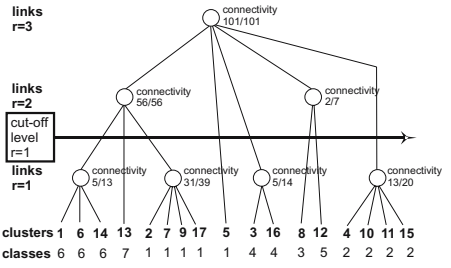


Fig. 4. The HIERDENC tree resulting from clustering the *zoo* dataset. A link (circle) represents two or more merged clusters.

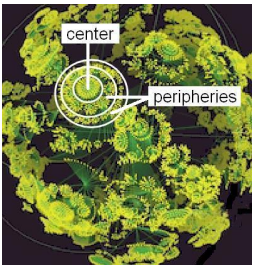


Fig. 5. A cluster has a center surrounded by peripheral areas (CAIDA)

merged gradually. As Figure 4 shows, a merge is represented as a *link* between two or more links or *leaf* clusters, created at a level $r \geq 1$. A link represents a group of merged clusters. This process gradually constructs one or more cluster tree structures, resembling hierarchical clustering [18,24]. The user specifies a cut-off level (e.g. $r = 3$) to cut tree branches; links at the cut-off level are extracted as merged clusters. *Step 5* checks if a newly formed cluster is connected to another cluster relative to r and if so links them at level r. *Step 6* continues linking existing clusters into a tree, until $r = m$. By allowing r to reach m, an entire tree is built. At the top of the tree, there is a single cluster containing all objects of the dataset.

In [3] we propose and evaluate several methods for setting the HIERDENC tree cut-off level. One method involves cutting the HIERDENC tree at level r such that the average connectivity of the resulting merged clusters is minimized. The $connectivity_r$ of a merged cluster (a set of connected leaf clusters) relative to r is the fraction of its objects that have another object within distance r in a different leaf cluster in the same connected set. Another method useful for finding clusters within clusters is to set the cut-off(s) for a branch of links from leafs to root at the *level(s)* $r \geq 1$ such that the resulting merged cluster has $0.0 < connectivity_r < 1.0$. Another method is to balance the number of clusters with the entropy of the partition [22]. This involves setting the cut-off at level r such that the *Akaike's Information Criterion (AIC)* is minimized [2]. The AIC of a partition is $entropy + 2k$, where k is the number of clusters.

Although HIERDENC has similarities to CLIQUE [1], the two have significant differences. HIERDENC is intended for categorical data while CLIQUE for numerical data. HIERDENC minimizes input parameters, while CLIQUE takes as input parameters the grid size and a global density threshold for clusters. HIERDENC retrieves the densest hyper-cube relative to the radius. The radius relaxes gradually, implying that HIERDENC can find clusters of different densities. HIERDENC can often distinguish the central hyper-cube of a cluster from the rest of the cluster, because of its higher density. HIERDENC creates a tree representing the entire dataset structure, including subclusters within clusters.

3 MULIC as a Simplification of HIERDENC

MULIC stands for *multiple layer clustering* of categorical data. MULIC is a faster simplification of HIERDENC. MULIC balances clustering accuracy with time efficiency. The MULIC algorithm is motivated by datasets the cluster structure of which can be visualized as shown in Figure 5. In such datasets a cluster often has a center of objects that are similar to one another, along with peripheral objects that are less similar to the central objects. Such datasets include protein interaction data, large software systems and others [7].

MULIC does not store the cube in memory and makes simplifications to decrease the runtime. A MULIC cluster starts from a dense area and expands outwards via a radius represented by the ϕ variable. When MULIC expands a cluster it does not search all member objects as HIERDENC does. Instead, it uses a mode that summarizes the content of a cluster. The mode of cluster c is a vector $\mu_c = \{\mu_{c1}, \cdots, \mu_{cm}\}$ where μ_{ci} is the most frequent value for the ith attribute in the given cluster c [16]. The MULIC clustering algorithm ensures that when an object o is clustered it is inserted into the cluster c with the least dissimilar mode μ_c. The default dissimilarity metric between o and μ_c is the Hamming distance presented in Section 2.1, although any metric could be used. A MULIC cluster consists of *layers* formed gradually, by relaxing the maximum dissimilarity criterion ϕ for inserting objects into existing clusters. MULIC does not require the user to specify the number of clusters and can identify outliers. Figure 6 shows the main part of the MULIC clustering algorithm. An optional final step merges similar clusters to reduce the number of clusters and find more interesting structures.

Merging of Clusters. Sometimes the dissimilarity of the top layers of two clusters is less than the dissimilarity of the top and bottom layers of one of the two clusters. To avoid this, after the clustering process MULIC can merge pairs of clusters whose top layer modes' dissimilarity is less than the maximum layer depth of the two clusters. For this purpose, MULIC preserves the modes of the top layers of all clusters. The default merging process, detailed in [3], merges clusters in a non-hierarchical manner such that clusters have a clear separation. However, a hierarchical cluster merging process is also proposed [3].

MULIC Discussion. MULIC is a simplification of HIERDENC. The tradeoffs between accuracy and time efficiency are as follows:

(i) When creating a cluster, HIERDENC searches the cube to retrieve the densest hyper-cube relative to r representing two or more objects, which is costly. MULIC creates a cluster if *two* or more objects are found within a dissimilarity distance of ϕ from each other, likely indicating a dense subspace. Clusters of size one are filtered out. MULIC's ϕ variable is motivated by HIERDENC's radius r. The initial objects clustered with MULIC affect the modes and the clustering. For this issue we propose in [3] an optional preprocessing step that orders the objects by decreasing aggregated frequency of their attribute values, such that objects with more frequent values are clustered first and the modes will likely

Input: a set S of objects.
Parameters: (1) $\delta\phi$: the increment for ϕ.
(2) *threshold* for ϕ : the maximum number of values that can differ between an object and the mode of its cluster.
Default parameter values: (1) $\delta\phi = 1$.
(2) *threshold* = the number of categorical attributes m.
Output: a set of clusters.
Method:

1. Order objects by decreasing aggregated frequency of their attribute values.
2. Insert the first object into a new cluster, use the object as the mode of the cluster, and remove the object from S.
3. Initialize ϕ to 1.
4. Loop through the following until S is empty or $\phi > threshold$
  a. For each object o in S
    i. Find o's nearest cluster c by using the dissimilarity metric to compare o with the modes of all existing cluster(s).
    ii. If the number of different values between o and c's mode is larger than ϕ, insert o into a new cluster
    iii. Otherwise, insert o into c and update c's mode.
    iv. Remove object o from S.
  b. For each cluster c, if there is only one object in c, remove c and put the object back in S.
  c. If in this iteration no objects were inserted in a cluster with $size > 1$, increment ϕ by $\delta\phi$.

Fig. 6. The MULIC clustering algorithm

contain the most frequent values. This object ordering process has been evaluated in [3], which showed that it is better than a random ordering of objects; we do not include the same results here.

(ii) When expanding a cluster HIERDENC searches the member cells to find dense hyper-cubes relative to r, which is costly. MULIC instead uses a 'mode' as a summary of a cluster's content and only clusters objects within a distance of ϕ from the mode. MULIC increases ϕ by $\delta\phi$ when no new objects can be clustered, which is motivated by HIERDENC's increasing r. MULIC can create new clusters at any value of ϕ, just as HIERDENC can create new clusters at any value of r. Although MULIC can find clusters of arbitrary shapes by increasing ϕ, it loses some of HIERDENC's ability in this realm.

(iii) MULIC's cluster merging is motivated by HIERDENC's merging. The MULIC cluster merging process can organize clusters into a tree structure as HIERDENC does. For MULIC applications, such as the one on protein interaction data discussed in [3], we do not construct a tree since we prefer the clusters to have a clear separation and not to specify a cut-off.

MULIC has several differences from traditional hierarchical clustering, which stores all distances in an upper square matrix and updates the distances after

each merge [18,24]. MULIC clusters have a clear separation. MULIC does not require a cut-off to extract the clusters, as in hierarchical clustering; this is of benefit for some MULIC applications, such as the one on protein interaction data discussed in [3]. One of the drawbacks of hierarchical clustering is that the sequence of cluster mergings will affect the result and 'bad' mergings can not be undone later on in the process. Moreover, if several large clusters are merged then interesting local cluster structure is likely to be lost. MULIC, on the other hand, does not merge clusters during the object clustering. Instead, any cluster mergings that may be desirable for the particular application are done after object clustering has finished. MULIC aims not to lose cluster structure caused by several large clusters being merged during the clustering process.

Computational Complexity. The best-case complexity of MULIC has a lower bound of $\Omega(mNk)$ and its worst-case complexity has an upper bound of $O(mN^2 \frac{threshold}{\delta\phi})$. The cost is related to the number of clusters k and the number of objects N. Often $k \ll N$, $m \ll N$, and all objects are clustered in the initial iterations, thus N often dominates the cost. The worst-case runtime would occur for the rather uncommon dataset where all objects were extremely dissimilar to one another, such that the algorithm had to go through all m iterations and all N objects were clustered in the last iteration when $\phi = m$. The MULIC complexity is comparable to that of k-Modes of $O(mNkt)$, where t is the number of iterations [16].

4 Performance Evaluation

To evaluate the applicability of HIERDENC and MULIC to the clustering problem, we first use the *zoo* and *soybean-data* categorical datasets. These datasets were obtained from the UCI Repository [23]. Objects have class labels defined based on some domain knowledge. We ignore class labels during clustering. We compare the HIERDENC and MULIC results to those of several other density-based algorithms, ROCK [14], CLICKS [28], k-Modes [16], and AutoClass [26]. CLICKS was shown to outperform STIRR [11] and CACTUS [10]. To evaluate the clustering quality we use *HA Indexes* [17] and *Akaike's Information Criterion (AIC)* [2]. HA Indexes is a class-label-based evaluation, which penalizes clustering results with more or fewer clusters than the defined number of classes. Since the class labels may or may not be consistent with the clustering structure and dissimilarity measure used, we also estimate the AIC of each clustering. AIC penalizes non-uniformity of attribute values in each cluster and too many clusters. In [3] we discuss MULIC with non-hierarchical and hierarchical merging of clusters applied to protein interaction data and large software systems.

For MULIC we set $\delta\phi = 1$, $threshold = m$, and we order the objects as described in [3]. We applied the other algorithms (except HIERDENC) on more than 10 random orderings of the objects. For k-Modes and ROCK we set the number of clusters k to the number of classes, as well as larger numbers. AutoClass considers varying numbers of clusters from a minimum of 2.

Table 1. HA Indexes (higher is better), Entropy, and AIC measures (lower is better)

	zoo (7 classes)					soybean-data (19 classes)				
Tool	*HAI.*	*Entr.*	*AIC*	*k*	*sec*	*HAI.*	*Entr.*	*AIC*	*k*	*sec*
HIERDENC (leaf clusters)	85.5%	2.15	36.15	17	0.04	92.2%	4.4	182.4	89	2.1
HIERDENC (after tree cut)	94%	2.3	18.3	8	0.04	95%	7.5	47.5	20	2.1
MULIC (no merging)	84%	2.5	40.5	19	0	92%	4.6	182.6	89	0.05
MULIC (after merging)	91.5%	2.8	22.8	10	0.03	93%	11.5	61.5	25	0.08
k-Modes	90%	3.5	23.5	10	0.005	80%	16.12	66.12	25	0.03
ROCK	73%	3.7	23.7	10	0.008	69.2%	19.5	69.5	25	0.04
AutoClass	79.5%	4.8	16.8	6	0.04	77.6%	25	39	7	0.13
CLICKS	91.5%	2.5	20.5	9	0.01	70%	10	90	40	1
Chameleon (wCluto part.)	72%	3.8	23.8	10	0	79%	16.5	66.5	25	0.1

HIERDENC Results. Table 1 shows the HIERDENC results for these datasets before and after cutting the tree. After cutting the HIERDENC tree for *zoo*, its HA Indexes, Entropy, and AIC are slightly better than CLICKS. The HIERDENC results for *soybean-data* are significantly better than CLICKS. The Entropy is naturally lower (better) in results with many clusters; by comparing results of algorithms with similar numbers of clusters, the HIERDENC Entropy is often lower. The drawback we notice is that the HIERDENC runtime is significantly higher on *soybean-data* than on *zoo*.

Figure 4 illustrates the HIERDENC tree for *zoo*. There are 17 leaf clusters in total in the HIERDENC tree. Except for the last 3 created leaf clusters, all other leaf clusters are homogeneous with regards to the class labels of member objects. The last 3 leaf clusters were created for high r values of 7, 6, and 4. The rest of the leaf clusters were created for lower r values. For *zoo* we cut off the HIERDENC tree at level $r = 1$; *zoo* is a rather dense cube with many nonzero cells and we do not want to aggregate information in the cube. The $r = 1$ cut-off minimizes the connectivity relative to r of the resulting clusters. By cutting the HIERDENC *zoo* tree at $r = 1$, there are 8 resulting clusters. There are a few cases of incorrectly clustered objects by cutting at $r = 1$. However, the lower number of clusters results in improved HA Indexes.

For the *soybean-data* set, there are 89 leaf clusters in total in the HIERDENC tree. The leaf clusters created for $r \leq 9$ are homogeneous with regards to the class labels of member objects. For leaf clusters created for $r > 9$, the homogeneity of the class labels decreases. Only 23 objects are clustered for $r > 9$, so these could be labeled as outliers. For *soybean-data* we cut off the HIERDENC tree at $r = 4$; *soybean-data* is a sparse cube of mostly '0' cells, since the dataset has 35 dimensions but only 307 objects. The $r = 4$ cut-off minimizes the connectivity relative to r of the resulting clusters. By cutting the HIERDENC *soybean-data* tree at $r = 4$, there are 20 resulting merged clusters.

MULIC Results. Table 1 shows the MULIC results for these datasets with and without merging of clusters. MULIC has good Entropy measures and HA Indexes, because the attribute values are quite uniform in clusters. It is interesting

how MULIC finds subclusters of similar animals; for example, the animals 'porpoise', 'dolphin', 'sealion', and 'seal' are clustered together in one MULIC cluster. MULIC with non-hierarchical merging of clusters has as a result that the number of clusters decreases, which often improves the quality of the result according to the HA Indexes. After merging the MULIC clusters, the number of clusters for *zoo* and *soybean-data* is close to the class-label-based number of classes. After merging the MULIC clusters for *zoo*, the HA Indexes, Entropy, and AIC are as good as CLICKS. The MULIC results for *soybean-data* are better than CLICKS. The Entropy is naturally lower (better) in results with many clusters; by comparing results of algorithms with similar numbers of clusters, the MULIC Entropy is often lower. MULIC runtimes are lower than HIERDENC.

5 Conclusion

We have presented the HIERDENC algorithm for categorical clustering. In HIERDENC a central subspace often has a higher density and the radius relaxes gradually. HIERDENC produces good clustering quality on small-dimensional datasets. HIERDENC motivates developing faster clustering algorithms.

MULIC balances clustering accuracy with time efficiency. MULIC provides a good solution for domains where clustering primarily supports long-term strategic planning and decision making, such as analyzing protein-protein interaction networks or large software systems [3]. The tradeoffs involved in simplifying HIERDENC with MULIC point us to the challenge of designing categorical clustering algorithms that are accurate and efficient.

References

1. R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. SIGMOD 1998
2. H. Akaike. A new look at the statistical model identification. IEEE TAC, 19, 716-23, 1974
3. B. Andreopoulos. Clustering Algorithms for Categorical Data. PhD Thesis, Dept of Computer Science & Engineering, York University, Toronto, Canada, 2006
4. M. Ankerst, M. Breunig, H.P. Kriegel, J. Sander. OPTICS: Ordering Points to Identify the Clustering Structure. SIGMOD 1999
5. D. Barbara, Y. Li, J. Couto. COOLCAT:an entropy-based algorithm for categorical clustering. CIKM 2002
6. P. Berkhin. Survey of Clustering Data Mining Techniques. Accrue Software, Inc. TR, San Jose, USA, 2002
7. Z. Dezso, Z.N. Oltvai, A.L. Barabasi. Bioinformatics analysis of experimentally determined protein complexes in the yeast Saccharomyces cerevisiae. Genome Res. 13, 2450-4, 2003
8. M. Ester, H.P. Kriegel, J. Sander, X. Xu. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD 1996
9. G. Even, J. Naor, S. Rao, B. Schieber. Fast Approximate Graph Partitioning Algorithms. SIAM Journal on Computing, 28(6):2187-2214, 1999

10. V. Ganti, J. Gehrke, R. Ramakrishnan. CACTUS-clustering categorical data using summaries. KDD 1999
11. D. Gibson, J. Kleiberg, P. Raghavan. Clustering Categorical Data: an Approach based on Dynamical Systems. VLDB 1998
12. A. Gionis, A. Hinneburg, S. Papadimitriou, P. Tsaparas. Dimension Induced Clustering. KDD 2005
13. J. Grambeier, A. Rudolph. Techniques of Cluster Algorithms in Data Mining. Data Mining and Knowledge Discovery 6: 303-360, 2002
14. S. Guha, R. Rastogi, K. Shim. ROCK: A Robust Clustering Algorithm for Categorical Attributes. Information Systems 25(5): 345-366, 2000
15. A. Hinneburg, D.A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD 1998
16. Z. Huang. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining & Knowledge Disc. 2(3): 283-304, 1998
17. L.Hubert & P.Arabie. Comparing partitions. J.Classification 193-218, 1985
18. D. Jiang, J. Pei, A. Zhang. DHC: a density-based hierarchical clustering method for time series gene expression data. IEEE Symp. on Bioinf. and Bioeng., 2003
19. G. Karypis, E.H. Han, V. Kumar. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Computer 32(8): 68-75, 1999
20. J. Kleinberg, C. Papadimitriou, P. Raghavan. Segmentation Problems. STOC 1998
21. R. Krauthgamer, J.R. Lee. The black-box complexity of nearest neighbor search. ICALP 2004
22. T. Li, S. Ma, M. Ogihara. Entropy-Based Criterion in Categorical Clustering. ICML 2004
23. C.J. Mertz, P. Merphy. UCI Repository of Machine Learning Databases, 1996
24. R. Mojena, Hierarchical grouping methods and stopped rules: An evaluation. The Computer Journal, 20(4), 359-63, 1977
25. C.Papadimitriou. Algorithms, Games, and the Internet. STOC 2001
26. J. Stutz and P. Cheeseman. Bayesian Classification (AutoClass): Theory and results. Advances in Knowledge Discovery & Data Mining, 153-180, 1995
27. Y. Yang, S. Guan, J. You. CLOPE: a fast and effective clustering algorithm for transactional data. KDD 2002
28. M. Zaki, M. Peters. CLICK: Clustering Categorical Data using K-partite Maximal Cliques. TR04-11, Rensselaer Polytechnic Institute, 2004
29. Y. Zhang, A.W. Fu, C.H. Cai, P.A. Heng. Clustering Categorical Data. ICDE 2000

Multi-represented Classification Based on Confidence Estimation

Johannes Aßfalg, Hans-Peter Kriegel, Alexey Pryakhin, and Matthias Schubert

Institute for Informatics, Ludwig-Maximilians-University of Munich, Germany
{assfalg,kriegel,pryakhin,schubert}@dbs.ifi.lmu.de

Abstract. Complex objects are often described by multiple representations modeling various aspects and using various feature transformations. To integrate all information into classification, the common way is to train a classifier on each representation and combine the results based on the local class probabilities. In this paper, we derive so-called confidence estimates for each of the classifiers reflecting the correctness of the local class prediction and use the prediction having the maximum confidence value. The confidence estimates are based on the distance to the class border and can be derived for various types of classifiers like support vector machines, k-nearest neighbor classifiers, Bayes classifiers, and decision trees. In our experimental results, we report encouraging results demonstrating a performance advantage of our new multi-represented classifier compared to standard methods based on confidence vectors.

1 Introduction

In many application areas such as multimedia, biology, and medicine, objects can be described in multiple ways. For example, images can be described by color histograms or texture features or proteins can be described by text annotations, sequence descriptions, and 3D shapes. To classify these multi-represented objects, it is often useful to integrate as much information as possible because the representation providing the best suitable object description might vary from object to object. A simple way to combine multiple representations would be to span a feature space with respect to all features occurring in some representation. However, this approach induces treating sparse dimensions such as word occurrences in the same way as color distributions or texture descriptions. Therefore, established methods for classifier combination, which is also called classifier fusion, train a classifier on each representation and derive a global class prediction based on the class probabilities of each of these classifiers.

In this paper, we introduce a new method for combining local class predictions based on so-called confidence estimates. A confidence estimate reflects the degree of reliability for the class prediction of a given classifier. In contrast, the probability distributions used in the established methods for building combined classifiers represent the likelihood that an object o belongs to any of the possible classes. The difference becomes clear when considering the following two-class case. A classic probability estimate for class c_1 of 40 % implies that it would be better to predict class c_2 which must have correspondingly a probability estimate of 60 %. On the other hand, a confidence estimate of the class decision for class c_1 of 40 % only implies that the result of the

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 23–34, 2007.

classifier is rather unreliable. In multi-class problems, the difference can be seen more easily because there is only 1 confidence estimate for the class decision and not a separated probability estimation for each of the classes. In this paper, we argue that reliable confidence estimates are easier to derive and that a confidence estimate yields enough information for learning powerful multi-represented classifiers.

A second advantage of the proposed method is that confidence estimates can be derived from multiple types of classifiers such as Support Vector Machines (SVMs), Bayes classifiers, k-nearest-neighbor (kNN) classifiers and decision trees. Since the principle idea of deriving confidence estimates is the same for each of these classifiers, the confidence estimates are directly comparable even for different types of classifiers trained on varying feature spaces. This is not necessarily the case for probability estimation because the idea behind approximating the class probabilities is often quite different for various classifiers. Thus, the semantic of the probabilities is not necessarily comparable leading to suboptimal classification results.

To derive confidence estimates, we introduce the concept of the confidence range of a decision. The confidence range is the smallest range inside which the classified object could be moved in order to be assigned to a different class. In other words, the confidence range corresponds to the distance of an object to the closest class border. Therefore, deriving confidence estimates from SVMs can be done in a straightforward way. However, there still exist major differences between the probability estimation for a SVM as proposed in [1] and the confidence estimate employed in this paper. First of all, it is possible to derive a confidence estimate of less than 50 % for the predicted class, if it is quite uncertain that the prediction is correct. Additionally, the method proposed in [1] yields a solution for probability estimation in two class problems while our method using confidence estimates can be applied to an arbitrary number of classes. For employing other classifiers than SVMs, we will provide algorithms for several well-established classification methods like Bayes classifiers, decision trees, and kNN classifiers for deriving confidence ranges. Afterwards the confidence ranges are used to calculate confidence estimates which are finally used to find the global class decision. The main contributions of this paper are:

- A new method for combining classifiers based on the confidence estimates instead of complete distribution vectors.
- Methods for deriving confidence ranges for various classifiers such as decision trees, Bayes classifiers, or kNN classifiers.
- Methods for learning a function that derives confidence estimates from the derived confidence ranges.

Our experimental evaluation illustrate the capability of our new approach to improve the classification accuracy compared to combined classifiers that employ distribution vectors.

The rest of the paper is organized as follows. Section 2 surveys related work. In section 3, we introduce the general idea for our method of classifier combination. Afterwards, section 4 describes methods to derive confidence ranges for various classifiers and explains their use for deriving confidence estimates. The results of our experimental evaluation are shown in section 5. Section 6 concludes the paper with a summary and ideas for future work.

2 Related Work

In general, methods that employ multiple learners to solve a common classification problem are known as ensemble learning. An overview over ensemble learning techniques can be found in [2]. Within the area of ensemble learning our work deals with the subarea of classifier combination. The aim of classifier combination is to use multiple independently trained classifiers and combine their results to increase the classification accuracy in comparison to the accuracy of a single classifier. Combining classifiers to learn from objects given by multiple representations has recently drawn some attention in the pattern recognition community [3,4,5]. The authors of [3] developed a theoretical framework for combining classifiers which use multiple pattern representations. Furthermore, the authors propose several combination strategies like max, min, and, product rule. [4] describes so-called decision templates for combining multiple classifiers. The decision templates employ the similarity between classifier output matrices. In [5] the author proposes a method of classifier fusion to combine the results from multiple classifiers for one and the same object. Furthermore, [5] surveys the four basic combination methods and introduces a combined learner to derive combination rules increasing classification accuracy. All methods mentioned above assume that a classifier provides reliable values of the posteriori probabilities for all classes. Techniques for deriving probability estimates from various classifiers can be found in [1,6]. Learning reliable probability estimates and measuring their quality is a rather difficult task, because the training sets are labeled with classes and not with class probability vectors. In contrast to these solutions, we propose a method that calculates a single confidence estimate reflecting the correctness of each particular class decision. A related subarea of ensemble learning is co-training or co-learning which assumes a semi-supervised setting. The classification step of co-training employs multiple independent learners in order to annotate unlabeled data. [7] and [8] were the first publications that reported an increase of classification accuracy by employing multiple representations. The most important difference of co-learning approaches to our new approach of multi-represented classification is that we do not consider a semi-supervised setting. Additionally, co-training retrains its classifiers within several iterations whereas the classifiers in our approach are only trained once. Recently, methods of hyper kernel learning [9] were introduced that are also capable of employing several representation in order to learn a classifier. In contrast to our method the hyper kernel learners optimize the use of several kernels that can be based on multiple representations within one complex optimization problem which is usually quite difficult to solve.

3 Confidence Based Multi-represented Classification

In this section we will specify the given task of multi-represented classification and describe our new approach of using a single confidence value for each representation to derive global class decisions.

A multi-represented object o is given by an n-tuple $(o_1, \ldots, o_n) \in R_1 \times \ldots \times R_n$ of feature vectors drawn from various feature spaces $R_i = F_i \cup \{-\}$. F_i denotes the corresponding feature space of representation i and "$-$" denotes that there is no object

description for this representation. Missing representations are a quite common problem in many application areas and thus, should be considered when building a method. Obviously, for each object there has to be at least one $o_j \neq$ "$-$". For a given set of classes $C = c_i, \ldots, c_k$, our classification task can be described in the following way. Given a training set of multi-represented objects $TR \subset R_1 \times \ldots \times R_n$, we first of all train a classifier $CL_i : R_i \rightarrow C$ for each representation. Afterwards, we train a confidence estimator $CE_{CL_i} : R_i \rightarrow [0..1]$ based on a second training set TR_{conf} for each classifier which predicts the confidence of the class decision $CL_i(o_i)$ for each object o_i. Let us note that we employed cross validation for this second training since the number of training objects is usually limited. To combine these results, we employ the following combination method:

$$CL_{global}(o) = CL_{\underset{0 \leq j \leq n}{\operatorname{argmax}} \{CE_{CL_{j(o)}}\}}(o)$$

where o is an unknown data object.

In other words, we employ each classifier $CL_j(o)$ for deriving a class and afterwards determine the confidence of this class decision $CE_{CL_j}(o)$. As a global result, we predict the class c_i which was predicted by the classifier having the highest confidence estimate $CE_{CL_i}(o)$. To handle unknown representations, we define $CE_{CL_j}(-) = 0$. Thus a missing representation cannot have the highest confidence value.

4 Deriving Confidence Estimates

After describing our general pattern for multi-represented classification, we now turn to describing the method for deriving the confidence estimates. The main idea of our proposed confidence estimation is that the larger the area around an object for which the class prediction does not change, the larger is the confidence for the class decision. In other words, the more we can alter the characteristics of an object without changing its class, the more typical is this object for the given class in the given feature space. The confidence range can be determined by calculating the distance of the classified object o to the closest class border. Let us note that we can apply this idea regardless of the used classification method. To formalize the area around each object for which the class prediction remains unchanged, we define the confidence range of an object as follows:

Definition 1. *Let $o \in F$ be a feature vector and let $CL : F \rightarrow C$ be a classifier w.r.t. the class set C. Then the confidence range $CRange(o)$ is defined as follows:*

$$CRange(o) = \min \{\|v\| \, | v \in F \wedge CL(o) \neq CL(o+v)\}$$

The methods for deriving the confidence range are varying between the classification methods. For SVMs the method to extract a confidence range is straightforward. For the two-class case, we can use the output of the SVM as distance to the separating hyperplane. In the case of multi-class SVMs, the minimum distance to all of the used hyperplanes is considered. For other classification paradigms, the calculation is less straightforward. In general, the confidence range of an object o can be determined by

taking the minimum distance for which the class prediction changes from class c_{pred} to some other class c_{other}. Thus, we can determine $CRange(o)$ by:

$$CRange(o) = \min_{c_{other} \in C \setminus c_{\{pred\}}} CRange_{c_{pred}, c_{other}}(o)$$

where $CRange_{c_{pred}, c_{other}}(o)$ is the distance of o to the class border between the predicted class c_{pred} and the class c_{other}. In the following, we will provide methods for approximating $CRange(o)$ for three well-established classification methods.

4.1 Bayes Classification

For a Bayes classifier over a feature space $F \subseteq \mathbb{R}^d$, each class c is determined by a prior $p(c)$ and a density function corresponding to $p(x|c)$ where $x \in F$. The class having the highest value for $p(c) \cdot p(x|c)$ is predicted. Thus, for each class c_{other}, we need to consider the following equation describing the class border between two classes:

$$p(c_{pred}) \cdot p(\boldsymbol{x}|c_{pred}) = p(c_{pred}) \cdot p(\boldsymbol{x}|c_{other})$$
$$\Rightarrow p(c_{pred}) \cdot p(\boldsymbol{x}|c_{pred}) - p(c_{pred}) \cdot p(\boldsymbol{x}|c_{other}) = 0$$

To determine the distance of an object o to this class border, we need to solve the following optimization problem:

$$\min_{x \in \mathbb{R}^d} d(o, x)$$
$$s.t.\ p(c_{pred}) \cdot p(\boldsymbol{x}|c_{pred}) - p(c_{pred}) \cdot p(\boldsymbol{x}|c_{other}) = 0$$

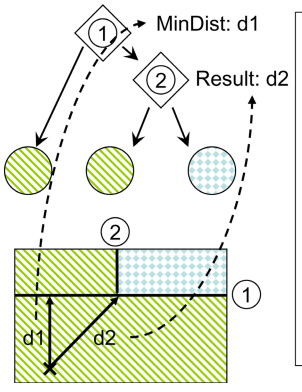
For example, the optimization problem for a general Gaussian distribution can be formulated as follows:

$$\min_{x \in \mathbb{R}^d} d(o, x)$$
$$s.t. (x - \mu_1)^T \times (\Sigma_1)^{-1} \times (x - \mu_1)$$
$$-(x - \mu_2)^T \times (\Sigma_2)^{-1} \times (x - \mu_2) - \ln \frac{p(c_1) \cdot \Sigma_2}{p(c_2) \cdot \Sigma_1} = 0$$

To solve this problem, we employed a gradient descent approach which is an iterative method for solving non linear optimization problems. Beginning at an initialized point, the direction of the steepest descent is determined. Then, a step in this direction is made whereas the step size is calculated by applying the Cauchy principle. The steps are repeated until the minimum is reached which usually occurs after a small number of iterations.

4.2 Decision Trees

For most decision trees, each node in the tree belongs to some discriminative function separating the training instances with respect to a single dimension of the feature space. Therefore, each leaf of a decision tree corresponds to a hyper rectangle. However, to




```
calculateCRange(Object o, Tree decisionTree)
  queue=getNewPriorityQueue(orderAscending);
  /*classify o, store omitted subtrees in queue with their MinDist to o */
  classID:=classify(o,decisionTree, queue);
  LOOP
    current=queue.removeFirst();
    if (current.isLeaf() AND current.getClass()!= classID) then
      return current.getMinDist();
    else
      foreach subtree ∈ current.getSubtrees()
        queue.insert(subtree, subtree.getMinDistTo(o);
    end if;
  end LOOP;
```

Fig. 1. Example and pseudo code for determining CRange for decision trees

determine $CRange(o)$, it is not sufficient to calculate the minimum distance of o to the border of this hyper rectangle. If the neighboring leaf does correspond to the same class, the border of the leaf does not provide a class border. Therefore, for determining $CRange(o)$, we need to find the distance of object o to the closest leaf belonging to any other class than c_{pred}. To find this leaf node while traversing the tree for classification, we collect all subtrees that do not contain o, i.e. the subtrees that are omitted during the classification of o. Each of these subtrees is stored in a priority queue which is ordered by the minimum distance of o to any node of the subtree. After the classification algorithm reaches a leaf node (i.e. o is classified), we can process the priority queue containing the collected subtrees. If the first object in the queue is a leaf node, we determine whether the leaf corresponds to a class which is different to the class prediction. In this case, we can calculate $CRange(o)$ as the distance of o to the boundaries of this leaf. Since the priority queue is ordered by the minimum distance to o, there cannot exist any other leaf with a boundary closer to o than the calculated $CRange(o)$. If the top element of the queue is a subtree, we remove the tree from the queue and insert its descendants. Figure 1 illustrates an example for a small decision tree on the left side. The right side of Figure 1 describes the proposed algorithm in pseudo code. Let us note that calculating the minimum distance of o to any leaf node in the tree can be done by only considering the attributes which are encountered while traversing the tree. For each other attribute, the feature value of o must be contained within the range of the tree node.

4.3 *k*NN Classification

Though it is sufficient for finding an accurate class decision, determining the k nearest neighbors for an object o is not enough to determine $CRange(o)$. Since the k nearest neighbors do not necessarily contain representative objects of each of the classes, finding the class border for a kNN classifier needs to consider additional objects belonging to each of the classes. If the number of considered neighbors is one, the class borders are described by Voronoi cells around the training objects. In this case, we can easily

calculate $CRange(o)$ on the basis of the particular $CRange_{c_{pred},c_{other}}(o)$. Thus, we only need to compare the distances to the class border which is determined by the nearest neighbor u_c of the predicted class c to any nearest neighbor $u_{\hat{c}}$ of the other classes $\hat{c}$. This distance can be calculated using the following lemma.

Lemma 1. *Let o be an object, let u_c be the nearest neighbor belonging to class $CL(o) = c$ and let u_{other} be the nearest object belonging to some other class $other \in C \setminus c$. Furthermore, let $d(x_1, x_2)$ be the Euclidian distance in $\mathbb{R}^d$. Then, $CRange_{c,other}(o)$ for a nearest neighbor classifier can be calculated as follows:*

$$CRange_{c,other}(o) = \frac{d(u_c, u_{other})^2 + d(u_c, o)^2 - d(u_{other}, o)^2}{2d(u_c, u_{other})} - \frac{d(u_c, u_{other})}{2}$$

A proof for this lemma can be found in [10].

Unfortunately, $CRange_{c,other}(o)$ is much more complicated to calculate for $k > 1$ because this would require to calculate Voronoi cells of the order k. Since this would cause a very time consuming calculations, we propose a heuristic method to approximate $CRange(o)$ for the case of $k > 1$. The idea of our heuristic is to determine the set U_c^k consisting of the k closest objects for each class c. Note that the union of these sets obviously contains the k nearest neighbors as well. For each class, we now build the weighted centroid. The weights for determining the centroid are derived by the inverse squared distance to the classified object o, in order to mirror the well-known weighted decision rule for kNN classifiers. Formally, these class representatives are defined as follows:

$$Rep_c^k(o) = \sum_{u_i \in U_c^k} \frac{1}{d(o, u_i)^2} \cdot u_i \cdot \frac{1}{\sum_{u_i \in U_c^k} \frac{1}{d(o,u_i)^2}}$$

After having calculated a representative for each class, we can proceed as in the case for $k = 1$. Let us note that using this heuristic, an object might have a negative distance to the class if it is placed on the wrong side of the estimated class border. However, this only occurs if the distance to the border is rather small and thus, the class decision is more or less unreliable.

4.4 From Ranges to Confidence Estimates

Our goal is to compare the results of various classifiers trained on varying feature spaces and employing various classification paradigms. However, the derived confidence ranges do not represent a comparable and accurate confidence estimate so far. To transform the confidence ranges into usable confidence estimates, we must cope with the following two problems. First the confidence ranges are distances in different feature spaces and thus, a comparably small distance in R_1 might induce a much higher confidence than a larger confidence range in representation R_2. Obviously, we have to learn which confidence range induces a high likelihood for a correct class prediction. A second problem we have to cope with is noise. In a noisy representation, an object having a comparably high confidence range might still be classified with comparably low confidence. Therefore, the confidence estimates should mirror the global reliability of the classifier as well as the confidence range of the individual object.

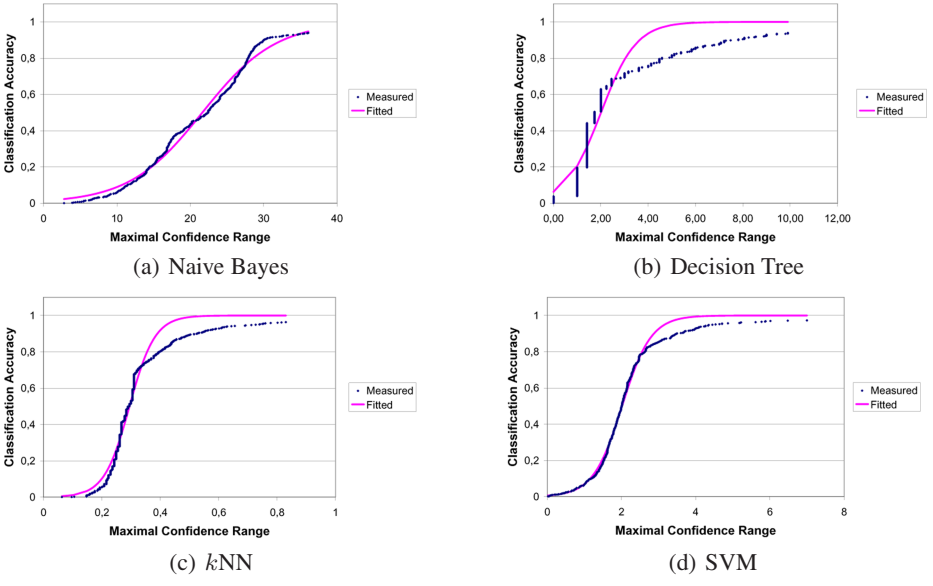


Fig. 2. Relationship between confidence range and classification accuracy

In order to understand the relationship between confidence ranges and the reliability of the class prediction, we performed several experiments. First, we partitioned the training data of each representation into three folds. We used two folds to train the classifier and used the remaining fold, the test fold, for calculating the confidence ranges. Afterwards, we formed a curve expressing the accuracy of the class predictions for all objects having a smaller $CRange(o)$ then the given x-value in the graph. More precisely, for each test object o a set of objects was determined containing all objects u in the test fold for which $CRange(u) \leq CRange(o)$. Then, the classification accuracy for each of these subsets was calculated providing the y-value for the x-value $CRange(o)$. We observed small classification accuracies for subsets having a small maximal confidence range. The accuracy is improved with increasing maximal confidence range values and finally reaches the accuracy observed on the complete test set. Furthermore, the graph displayed a sigmoidal pattern, i.e. there is a range of values where a small increase of the confidence ranges results in a high increase of classification accuracy. The results of the above described experiments are presented in Figure 2. The curve labeled with 'Measured' corresponds to these observed accuracy values while the curve labeled with 'Fitted' displays the function we will introduce in the following to model this behavior. As can be seen in Figure 2, the measured values form a sigmoidal shape for all examined classification techniques.

Based on these observations we developed a technique to calculate confidence estimates for each classifier. These confidence estimates range from 0 to 1 and thus, unlike confidence ranges, the confidence estimates are directly comparable to each other. Since the confidence estimates cannot become larger than the classification accuracy on the

Table 1. Description of the data sets

	DS1	DS2	DS3	DS4	DS5	DS6	DS7
Name	Oxidore-ductase	Trans-ferase	Transporter Activity	Protein Binding	Enzyme Regularization	Sonar	Wave 5000
No. of Classes	20	37	113	63	19	2	3
No. of Objects	1051	2466	2218	4264	1046	208	5000

complete test set, the noise level in each representation is considered as well. In the following, the calculation of the confidence estimates is described in detail.

1. For a given classifier CL_j, perform a 3-fold cross validation with the training data in order to yield confidence range/accuracy pairs as described above.
2. A suitable optimization algorithm (e.g. Levenberg-Marquardt algorithm [11]) is used to determine the parameters α_j and β_j that minimize the least squares error for the sigmoid target function $accuracy_j(o) = \frac{1}{1+exp(\alpha_j \times CRange_j(o)+\beta_j)}$ given the observed pairs of confidence ranges and classification accuracy.
3. For classifier CL_j and object o the confidence estimate $CE_{CL_j}(o)$ can finally be calculated as:

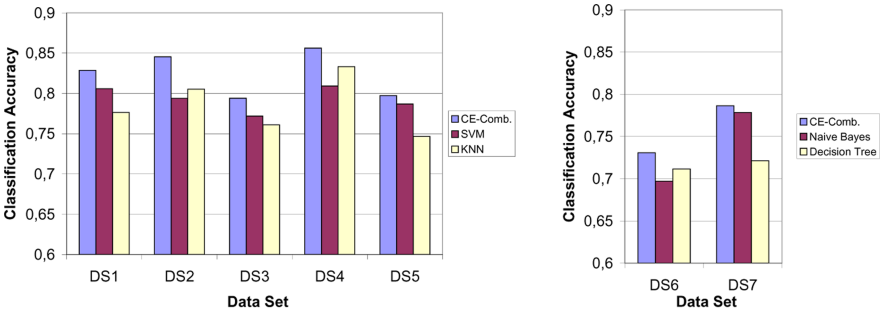
$$CE_{CL_j}(o) = \frac{1}{1 + exp(\alpha_j \times CRange_j(o) + \beta_j)}$$

The derived confidence estimates are now used for classifier combination as described in section 3, i.e. the classification result based on the representation yielding the highest confidence estimate is used as the global prediction of the combined classifier.

5 Experimental Evaluation

For our experimental evaluation, we implemented our new method for classifier combination and confidence estimation in Java 1.5. All experiments were performed on a workstation featuring two Opteron processors and 8 GB main memory. For comparing our method to the established methods for classifier combination as described in [5], we used the J48 (decision tree), Naive Bayes, SMO (SVM), and IBK (kNN classifier) classifiers provided by the WEKA machine learning package [12]. The WEKA implementation provides probabilities for each of the classes which were combined using the minimum, the maximum, the product, and average. For example, a joined confidence vector v is constructed by taking the average class probability for each class c_i over all representation R_j as ith component v_i. For our confidence estimates we used the same classifiers and additionally implemented our new method. For fitting the sigmoid functions we used the method introduced in [11].

We tested our new ensemble method on 5 multi-represented data sets describing proteins (DS1-DS5). The 5 test beds consist of 19 to 113 Gene Ontology [13] classes. The corresponding objects were taken from the SWISS-PROT [14] protein database and consist of a text annotation and the amino acid sequence of the proteins. In order to



(a) Protein data sets (b) UCI data sets

Fig. 3. Classification accuracy of CE-Comb. compared to separated classification

obtain a flat class-system with sufficient training objects per class, the original data was adapted. We employed the approach described in [15] to extract features from the amino acid sequences. The basic idea is to use local (20 amino acids) and global (6 exchange groups) characteristics of a protein sequence. To construct a meaningful feature space, we formed all possible 2-grams for each kind of characteristic, which yielded the 436 dimensions of our sequence feature space. For text descriptions, we employed a TFIDF [16] vector for each description that was built of 100 extracted terms. Since these data sets could only be tackled by the more flexible SVMs and kNN classifiers, we could not test the Naive Bayes and the J48 decision tree classifiers on these problems. Therefore, by splitting the wave-5000 (DS7) and the sonar data set (DS6) from the well-known UCI machine learning repository [17] vertically, we generated two additional multi-represented. Thus, we derived for each data set two representations containing only half of the attributes of the original data set. An overview of our 7 test beds is given in Table 1.

For our experiments, we first of all classified each representation separately using several classifiers. Afterwards, we combined the best classification method for each representation using our new method based on confidence estimates (CE-Comb.) and the 4 standard methods mentioned above which are based on probability vectors. For the UCI data sets, we tested only Naive Bayes and J48 because these data sets were chosen to provide an example for these two types of classifiers.

Our first result compares the classification accuracy of our new combination method with the classification accuracy achieved in the separated representations. Figure 3 displays the achieved classification accuracies. In all 7 data sets our new combination method achieved a higher classification accuracy than both corresponding classifiers which rely on only a single representation. Thus, using the additional information of both representations was always beneficial.

In Table 2 we compare our new method to the established methods of classifier combination. The accuracies which where achieved using our confidence estimates are shown in the first row. For all data sets our new method for classifier combination outperforms the established approaches. The improvement by using our new combination method was up to 4 % in data sets DS3 and DS6. Let us note that the classifiers in

Table 2. Accuracies for different combination methods

	DS 1	**DS 2**	**DS 3**	**DS 4**	**DS 5**	**DS 6**	**DS 7**
CE-Comb.	**0.8287**	**0.8454**	**0.7939**	**0.8562**	**0.7973**	**0.7692**	**0.7954**
Product	0.7761	0.8092	0.7633	0.8302	0.7514	0.7307	0.7794
Sum	0.8072	0.8051	0.7768	0.8142	0.7877	0.7307	0.7808
Min	0.7761	0.8053	0.7610	0.8332	0.7466	0.7307	0.7786
Max	0.8058	0.7939	0.7718	0.8090	0.7868	0.7307	0.7806

Table 3. Comparing various combinations

	DS 1	**DS 2**	**DS 3**	**DS 4**	**DS 5**
kNN+SVM	0.8116	0.8215	0.7831	0.8485	0.7782
SVM+kNN	**0.8287**	**0.8454**	**0.7939**	**0.8562**	**0.79732**
SVM+SVM	0.8097	0.836	0.7921	0.8410	0.7906
kNN+kNN	0.789	0.8215	0.7889	0.8499	0.7667

each representation for all types of ensembles were always exactly the same. Thus, the improvement is achieved by the different combination method only.

Our final result illustrates that the capability to combine classifiers of different types proves to be beneficial on real world data sets. We tested all possible combinations of SVMs and kNN classifiers for the 5 protein data sets. The results are displayed in Table 3. For all data sets, the combination of using a linear SVM for text classification and a nearest neighbor classifier for sequence classification proved to yield the best accuracy. Thus, our new method is capable of exploiting different types of classifiers which often yields a better ensemble than using only classifiers of one and the same type.

6 Conclusions

In this paper we describe a new method for classifier combination. The two main aspects of our new approach are the following. First of all, the global class decision is not dependent on complete probability distributions over all classes but depends only on a confidence estimate that the given classification result is indeed correct. The second aspect is that the used confidence estimates are not limited to a particular type of classifier. By introducing the general concept of confidence ranges, it is possible to generate comparable confidence estimates for different types of classifiers and varying feature spaces. To derive these confidence estimates, we provide algorithms to calculate confidence ranges for kNN classifiers, decision trees, and Bayes classifiers. The confidence ranges are transformed into meaningful confidence estimates using a trained sigmoid function. Our experimental evaluation shows that our new method is capable of outperforming established methods based on probability vectors. Additionally, we observed that it is sometimes indeed useful to use different types of classifiers for classifying different representations.

In our future work, we are going to extend our new idea for classifier combination to other methods of ensemble learning like co-training. Measuring the agreement

between the used classifiers by means of our new confidence estimates might yield an improvement compared to established methods using probability vectors.

References

1. Platt, J.: "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods". In: Advances in Large Margin Classifiers, MIT Press. (1999) 61–74
2. Valentini, G., Masulli, F.: "Ensembles of learning machines". Neural Nets WIRN (2002)
3. Kittler, J., Hatef, M., Duin, R., Matas, J.: "On Combining Classifiers". IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 226–239
4. Kuncheva, L., Bezdek, J., Duin, R.: "Decision Templates for Multiple Classifier Fusion: an Experimental Comparison". Pattern Recognition **34** (2001) 299–314
5. Duin, R.: "The Combining Classifier: To Train Or Not To Train?". In: Proc. 16th Int. Conf. on Pattern Recognition (ICPR'02), Quebec City, Canada). (2002) 765–770
6. Zadrozny, B., Elkan, C.: "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers". In: Proc. 18th Int. Conf. on Machine Learning, San Francisco, CA. (2001) 609–616
7. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: ACL. (1995) 189–196
8. Blum, A., Mitchell, T.: "Combining labeled and unlabeled data with co-training". In: Proc. of the eleventh annual conference on Computational learning theory (COLT '98), New York, NY, USA (1998) 92–100
9. Ong, C.S., Smalo, A.: "Machine Learning with Hyperkernels". In: Proc. of the 20th Int. Conf. (ICML 2003), Washington, DC, USA. (2003) 576–583
10. Kriegel, H.P., Schubert, M.: "Advanced Prototype Machines: Exploring Prototypes for classification". In: in Proc. 6th SIAM Conf. on Data Mining, Bethesda, MD. USA. (2006) 176–188
11. Levenberg, K.: "A method for the solution of certain problems in least squares". Quart. Appl. Math. **2** (1944) 164–168
12. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Jave Implementations. Morgan Kaufmann (1999)
13. The Gene Ontology Consortium: "Gene Ontology: Tool for the Unification of Biology". Nature Genetics **25** (2000) 25–29
14. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: "The SWISS-PROT Protein Knowledgebase and its Supplement TrEMBL in 2003". Nucleic Acid Research **31** (2003) 365–370
15. Deshpande, M., Karypis, G.: "Evaluation of Techniques for Classifying Biological Sequences". In: Proc. of the 6th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining (PAKDD '02), London, UK (2002) 417–431
16. Salton, G.: Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley (1989)
17. University of Irvine. http://www.ics.uci.edu/ mlearn/MLRepository.html, UCI Machine Learning Repository. (2005)

Selecting a Reduced Set for Building Sparse Support Vector Regression in the Primal

Liefeng Bo, Ling Wang, and Licheng Jiao

Institute of Intelligent Information Processing
Xidian University, Xi'an 710071, China
{blf0218, wliiip}@163.com
http://see.xidian.edu.cn/graduate/lfbo

Abstract. Recent work shows that Support vector machines (SVMs) can be solved efficiently in the primal. This paper follows this line of research and shows how to build sparse support vector regression (SVR) in the primal, thus providing for us scalable, sparse support vector regression algorithm, named SSVR-SRS. Empirical comparisons show that the number of basis functions required by the proposed algorithm to achieve the accuracy close to that of SVR is far less than the number of support vectors of SVR.

1 Introduction

Support vector machines (SVMs) [1] are powerful tools for classification and regression. Though very successful, SVMs are not preferred in application requiring high test speed since the number of support vectors typically grows linearly with the size of the training set [2]. For example in on-line classification and regression, in addition to good generalization performance, high test speed is also desirable. Reduced set (RS) methods [3-4] have been proposed for reducing the number of support vectors. Since these methods operate as a post-processing step, they do not directly approximate the quantity we are interested in. Another alternative is the reduced support vector machines (RSVM) [5], where the decision function is expressed as a weighted sum of kernel functions centered on a random subset of the training set. Though simple and efficient, RSVM may result in a lower accuracy than the reduced set methods when their number of support vectors is kept in the same level.

Traditionally, SVMs are trained by using decomposition techniques such as SVMlight [6] and SMO [7], which solve the dual problem by optimizing a small subset of the variables each iteration. Recently, some researchers show that both linear and non-linear SVMs can be solved efficiently in the primal. As for linear SVMs, finite Newton algorithm [8-9] has proven to be more efficient than SMO. As for non-linear SVM, recursive finite Newton algorithm [10-11] is as efficient as the dual domain method. Intuitively, when our purpose is to compute an approximate solution, the primal optimization is preferable to the dual optimization because it directly minimizes the quantity we are interested in. On the contrary, introducing approximation in the dual may not be wise since there is indeed no guarantee that an approximate dual solution yields a good approximate primal solution. Chapelle

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 35–46, 2007.

[10] compares the approximation efficiency in the primal and dual domain and validates this intuition.

In this paper, we develop a novel algorithm, named SSVR-SRS for building the reduced support vector regression. Unlike our previous work [11] where recursive finite Newton algorithm is suggested to solve SVR accurately, SSVR-SRS aims to find a sparse approximation solution, which is closely related to SpSVM-2 [12] and kernel matching pursuit (KMP) [13], and can be regarded as extension of the key idea of matching pursuit to SVR. SSVR-SRS iteratively builds a set of basis functions to decrease the primal objective function by adding one basis function at one time. This process is repeated until the number of basis functions has reached some specified value. SSVR-SRS can find the approximate solution at a rather low cost, i.e. $O(nm^2)$ where n is the number of training samples and m the number of all picked basis functions. Our experimental results demonstrate the efficiency and effectiveness of the proposed algorithms.

The paper is organized as follows. In Section 2, support vector regression in the primal is introduced. SSVR-SRS is discussed in Section 3. Comparisons with RSVM, LIBSVM 2.82 [14] and the reduced set method are reported in Section 4. Some conclusions and remarks are given in Section 5.

2 Support Vector Regression in the Primal

Consider a regression problem with training samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$ where $\mathbf{x}_i$ is the input sample and y_i is the corresponding target. To obtain a linear predictor, SVR solves the following optimization problem

$$\begin{aligned} &\min_{\mathbf{w},b}\left(\frac{\|\mathbf{w}\|^2}{2}+C\sum_{i=1}^{n}\left(\xi_i^p+\overline{\xi}_i^p\right)\right) \\ &s.t.\ \mathbf{w}\cdot\mathbf{x}_i+b-y_i\le\varepsilon+\xi_i \\ &\quad y_i-\mathbf{w}\cdot\mathbf{x}_i+b\le\varepsilon+\overline{\xi}_i \\ &\quad \xi_i,\overline{\xi}_i\ge 0,\ i=1,2,\cdots n \end{aligned} \quad . \tag{1}$$

Eliminating the slack variables $\{\xi_i,\overline{\xi}_i\}_{i=1}^n$ and dividing (1) by the factor C, we get the unconstrained optimization problem

$$\min_{\mathbf{w},b}\left(L_\varepsilon(\mathbf{w},b)=\sum_{i=1}^{n}l_\varepsilon\left(\mathbf{w}\cdot\mathbf{x}_i+b-y_i\right)+\lambda\|\mathbf{w}\|^2\right), \tag{2}$$

where $\lambda=\dfrac{1}{2C}$ and $l_\varepsilon(r)=\max\left(|r|-\varepsilon,0\right)^p$. The most popular selections for p are 1 and 2. For convenience of expression, the loss function with p=1 is referred to as insensitive linear loss function (ILLF) and that with p=2 insensitive quadratic loss function (IQLF).

Non-linear SVR can be obtained by using the map $\phi(\bullet)$ which is determined implicitly by a kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \bullet \phi(\mathbf{x}_j)$. The resulting optimization is

$$\min_{\mathbf{w},b} \left(L_\varepsilon(\mathbf{w},b) = \sum_{i=1}^{n} l_\varepsilon(\mathbf{w} \cdot \phi(\mathbf{x}_i) - y_i) + \lambda \|\mathbf{w}\|^2 \right), \tag{3}$$

where we have dropped b for the sake of simplicity. Our experience shows that the generalization performance of SVR is not affected by this drop. According to the representer theory [15], the weight vector $\mathbf{w}$ can be expressed in terms of training samples,

$$\mathbf{w} = \sum_{i=1}^{n} \beta_i \phi(\mathbf{x}_i). \tag{4}$$

Substituting (4) into (3), we have

$$\min_{\boldsymbol{\beta}} \left(L_\varepsilon(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_\varepsilon \left(\sum_{j=1}^{n} \beta_i k(\mathbf{x}_i \mathbf{x}_j) - y_i \right) + \lambda \sum_{i=1}^{n} \beta_i \beta_j k(\mathbf{x}_i \mathbf{x}_j) \right). \tag{5}$$

Introducing the kernel matrix $\mathbf{K}$ with $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{K}_i$ the i-th row of $\mathbf{K}$, (5) can be rewritten as

$$\min_{\boldsymbol{\beta}} \left(L_\varepsilon(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_\varepsilon(\mathbf{K}_i \boldsymbol{\beta} - y_i) + \lambda \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta} \right). \tag{6}$$

A gradient descent algorithm is straightforward for IQLF; however, it is not applicable to ILLF since it is not differentiable. Inspired by the Huber loss function [16], we propose an insensitive Huber loss function (IHLF)

$$l_{\varepsilon,\Delta}(z) = \begin{cases} 0 & if\ |z| \le \varepsilon \\ (|z| - \varepsilon)^2 & if\ \varepsilon < |z| < \Delta \\ (\Delta - \varepsilon)(2|z| - \Delta - \varepsilon) & if\ |z| \ge \Delta \end{cases}, \tag{7}$$

to approximate ILLF. We emphasize that Δ is strictly greater than ε, ensuring that IHLF is differentiable.

The properties of IHLF are controlled by two parameters: ε and Δ. With certain ε and Δ values, we can obtain some familiar loss functions: (1) for $\varepsilon = 0$ and an appropriate Δ, IHLF becomes the Huber loss function; (2) for $\varepsilon = 0$ and $\Delta = \infty$, IHLF becomes the quadratic (Gaussian) loss function; (3) for $\varepsilon = 0$ and $\Delta \to \varepsilon$, IHLF approaches the linear (Laplace) loss function; (4) for $0 < \varepsilon < \infty$ and $\Delta = \infty$, IHLF becomes the insensitive quadratic loss function; and, (5) for $0 < \varepsilon < \infty$ and $\Delta \to \varepsilon$, IHLF approaches the insensitive linear loss function.

Introducing IHLF into the optimization problem (6), we have the following primal objective function:

$$\min_{\boldsymbol{\beta}} \left(L_{\varepsilon,\Delta}(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_{\varepsilon,\Delta}(\mathbf{K}_i \boldsymbol{\beta} - y_i) + \lambda \boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta} \right). \tag{8}$$

3 Selecting a Reduced Set in the Primal

In a reduced SVR, it is desirable to decrease the primal objective function as much as possible with as few basis functions as possible. The canonical form of this problem is given by

$$\begin{aligned} &\min\left(L_{\varepsilon,\Delta}(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_{\varepsilon,\Delta}(\mathbf{K}_i\boldsymbol{\beta} - y_i) + \lambda\boldsymbol{\beta}^T\mathbf{K}\boldsymbol{\beta} \right), \\ &s.t.\ \|\boldsymbol{\beta}\|_0 \le m \end{aligned} \tag{9}$$

where $\|\bullet\|_0$ is the l^0 norm, counting the nonzero entries of a vector and m is the specified maximum size of basis functions. However, there are several difficulties in solving (9). First, the constraint is not differentiable, so gradient descent algorithms can not be used. Second, the optimization algorithms can become trapped in a shallow local minimum because there are many minima to (9). Finally, an exhaustive search over all possible choices ($\|\boldsymbol{\beta}\|_0 \le m$) is computational prohibitive since the number of possible combinations is $\sum_{i=1}^{m}\binom{n}{m}$, too large for current computers.

Table 1. Flowchart of SSVR-SRS

Algorithm 3.1 SSVR-SRS
1. Set $P = \varnothing$, $Q = \{1, 2, \cdots, n\}$, $\boldsymbol{\beta} = \mathbf{0}$;
2. Select a new basis function from Q; let s be its index and set $P = P \cup \{s\}$ and $Q = Q - \{s\}$;
3. Solve the sub-problem with respect to $\boldsymbol{\beta}_P$ and the remaining variables are fixed at zero.
4. Check whether the number of basis functions is equal to m, if so, stop; otherwise go to step 2.

In this paper, we will compute an approximate solution using a matching pursuit-like method, named SSVR-SRS, to avoid optimizing (9) directly. SSVR-SRS starts with an empty set of basis functions and selects one basis function at one time to decrease the primal objective function until the number of basis functions has reached a specified value. Flowchart of SSVR-SRS is shown in Table 1. The final decision function takes the form

$$f(\mathbf{x}) = \sum_{i \in P} \beta_i k(\mathbf{x}, \mathbf{x}_i). \tag{10}$$

The set of the samples associated with the non-zero weights is called reduced set. Because here the reduced set is restricted to be a subset of training set, we consider this method as “selecting a reduced set”.

3.1 Selecting Basis Function

Let $\mathbf{K}_P$ the sub-matrix of $\mathbf{K}$ made of the columns indexed by P, $\mathbf{K}_{XY}$ the sub-matrix of $\mathbf{K}$ made of the rows indexed by X and the columns indexed by Y and $\boldsymbol{\beta}_P$ the sub-vector indexed by P.

How do we select a new basis function from Q? A natural idea is to optimize the primal objective function with respect to the variables $\boldsymbol{\beta}_P$ and β_j for each $j \in Q$ and select the basis function giving the least objective function value. The selection process can be described as a two-layer optimization problem,

$$s = \arg\min_{j \in Q} \left(\min_{\boldsymbol{\beta}_P, \beta_j} \left(L_{\varepsilon,\Delta}(\boldsymbol{\beta}) = \sum_{i=1}^{n} l_{\varepsilon,\Delta}\left(\mathbf{K}_{iP}\boldsymbol{\beta}_P + \mathbf{K}_{ij}\beta_j - y_i\right) + \lambda \begin{bmatrix} \boldsymbol{\beta}_P \\ \beta_j \end{bmatrix}^T \begin{bmatrix} \mathbf{K}_{PP} & \mathbf{K}_{Pj} \\ \mathbf{K}_{jP} & \mathbf{K}_{jj} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_P \\ \beta_j \end{bmatrix} \right) \right). \quad (11)$$

This basis function selection method, called pre-fitting, has appeared in kernel matching pursuit for least squares problem. Unfortunately, pre-fitting needs to solve the $|P|+1$ dimensional optimization problem $|Q|$ times, the cost of which is obviously higher than that of optimizing the sub-problem.

A cheaper method is to select the basis function that best fits the current residual vector in terms of a specified loss function. This method originated from matching pursuit [17] for least squares problem and was extended to an arbitrary differentiable loss function in gradient boosting [18]. However, our case is more complicated due to the occurrence of the regularization term, and thus we would like to select the basis function that fits the current residual vector and the regularization term as well as possible. Let the current residual vector be

$$\begin{cases} \mathbf{r}\left(\boldsymbol{\beta}_P^{opt}\right) = \mathbf{K}_P\boldsymbol{\beta}_P^{opt} - \mathbf{y} \\ r_i\left(\boldsymbol{\beta}_P^{opt}\right) = \mathbf{K}_{iP}\boldsymbol{\beta}_P^{opt} - y_i \end{cases}, \quad (12)$$

where $\boldsymbol{\beta}_P^{opt}$ is the optimal solution obtained by solving the sub-problem, and the index of basis function can be obtained by solving the following two-layer optimization problem,

$$s = \arg\min_{j \in Q} \left(\min_{\beta_j} \left(L_{\varepsilon,\Delta}(\beta_j) = \sum_{i=1}^{n} l_{\varepsilon,\Delta}\left(r_i\left(\boldsymbol{\beta}_P^{opt}\right) + \mathbf{K}_{ij}\beta_j\right) + \lambda \begin{bmatrix} \boldsymbol{\beta}_P^{opt} \\ \beta_j \end{bmatrix}^T \begin{bmatrix} \mathbf{K}_{PP} & \mathbf{K}_{Pj} \\ \mathbf{K}_{jP} & \mathbf{K}_{jj} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta}_P^{opt} \\ \beta_j \end{bmatrix} \right) \right). \quad (13)$$

Note that unlike pre-fitting, here $\boldsymbol{\beta}_P^{opt}$ is fixed.

$L_{\varepsilon,\Delta}(\beta_j)$ is one dimensional, piecewise quadratic function and can be minimized exactly. However, in practice, it is not necessary to solve it precisely. A simpler method is to compare the square of the gradient of $L_{\varepsilon,\Delta}(\beta_j)$ at $\beta_j = 0$ for all $j \in Q$,

$$\left(\nabla L_{\varepsilon,\Delta}(0)\right)^2 = \left(\mathbf{g}^T\mathbf{K}_j + 2\lambda\boldsymbol{\beta}_P^{opt}\mathbf{K}_{Pj}\right)^2, \quad (14)$$

where

$$g_i = \begin{cases} 0 & if \ \left| r_i\left(\boldsymbol{\beta}_P^{opt}\right)\right| \le \varepsilon \\ 2sign\left(r_i\left(\boldsymbol{\beta}_P^{opt}\right)\right)\left(\left| r_i\left(\boldsymbol{\beta}_P^{opt}\right)\right| - \varepsilon\right) & if \ \varepsilon < \left| r_i\left(\boldsymbol{\beta}_P^{opt}\right)\right| < \Delta, \\ 2sign\left(r_i\left(\boldsymbol{\beta}_P^{opt}\right)\right)(\Delta - \varepsilon) & if \ \left| r_i\left(\boldsymbol{\beta}_P^{opt}\right)\right| \ge \Delta \end{cases} \tag{15}$$

where $sign(z)$ is 1 if $z \ge 0$; otherwise $sign(z)$ is -1. To be fair, the square of the gradient should be normalized to

$$\frac{\left(\bar{\mathbf{g}}^T \bar{\mathbf{K}}_j\right)^2}{\left\|\bar{\mathbf{g}}\right\|_2^2 \left\|\bar{\mathbf{K}}_j\right\|_2^2}, \tag{16}$$

where $\bar{\mathbf{g}} = \begin{bmatrix} \mathbf{g} \\ 2\lambda \boldsymbol{\beta}_P^{opt} \end{bmatrix}$ and $\bar{\mathbf{K}}_j = \begin{bmatrix} \mathbf{K}_j \\ \mathbf{K}_{Pj} \end{bmatrix}$. This is an effective criterion because the gradient measures how well the j-th basis function fits the current residual vector and the regularization term. If set $\varepsilon = 0$, $\Delta = \infty$ and $\lambda = 0$, this criterion is exactly the one in the back-fitting version of KMP.

If each $j \in Q$ is tried, then the total cost of selecting a new basis function is $O\left(n^2\right)$, which is still more than what we want to accept. This cost can be reduced to $O(n)$ by only considering a random subset O of Q and selecting the next basis function only from O rather than performing an exhaustive search over Q,

$$s = \arg\min_{j \in O \subset Q} \left(\frac{-\left(\bar{\mathbf{g}}^T \bar{\mathbf{K}}_j\right)^2}{\left\|\bar{\mathbf{g}}\right\|_2^2 \left\|\bar{\mathbf{K}}_j\right\|_2^2} \right). \tag{17}$$

In the paper, we set $|O| = 100$.

3.2 Optimizing the Sub-problem

After a new basis function is included, the weights of basis functions, β_P are no longer optimal in terms of the primal objective function. This can be corrected by the so-called back-fitting method, which solves the sub-problem containing a new basis function and all previously picked basis functions. Thus, the sub-problem is a $|P|$ dimensional minimization problem expressed as

$$\min_{\boldsymbol{\beta}_P} \left(L_{\varepsilon,\Delta}\left(\boldsymbol{\beta}_P\right) = \sum_{i=1}^{n} l_{\varepsilon,\Delta}\left(\mathbf{K}_{iP}\boldsymbol{\beta}_P - y_i\right) + \lambda \boldsymbol{\beta}_P{}^T \mathbf{K}_{PP} \boldsymbol{\beta}_P \right). \tag{18}$$

$L_{\varepsilon,\Delta}\left(\boldsymbol{\beta}_P\right)$ is a piecewise quadratic convex function and continuously differentiable with respect to $\boldsymbol{\beta}_P$. Although $L_{\varepsilon,\Delta}\left(\boldsymbol{\beta}_P\right)$ is not twice differentiable, we still can use the finite Newton algorithm by defining the generalized Hessian matrix [11].

Define the sign vector $\mathbf{s}(\boldsymbol{\beta}_P) = \left[s_1(\boldsymbol{\beta}_P), \cdots, s_n(\boldsymbol{\beta}_P) \right]^T$ by

$$s_i(\boldsymbol{\beta}_P) = \begin{cases} 1 & if \ \ \varepsilon < r_i(\boldsymbol{\beta}_P) < \Delta \\ -1 & if \ -\Delta < r_i(\boldsymbol{\beta}_P) < -\varepsilon \\ 0 & otherwise \end{cases}, \tag{19}$$

the sign vector $\overline{\mathbf{s}}(\boldsymbol{\beta}_P) = \left[\overline{s}_1(\boldsymbol{\beta}_P), \cdots, \overline{s}_n(\boldsymbol{\beta}_P) \right]^T$ by

$$\overline{s}_i(\boldsymbol{\beta}_P) = \begin{cases} 1 & if \ \ r_i(\boldsymbol{\beta}_P) \geq \Delta \\ -1 & if \ r_i(\boldsymbol{\beta}_P) \leq -\Delta \\ 0 & otherwise \end{cases}, \tag{20}$$

and the active matrix

$$\mathbf{W}(\boldsymbol{\beta}_P) = diag\left\{ w_1(\boldsymbol{\beta}_P), \cdots, w_n(\boldsymbol{\beta}_P) \right\} \tag{21}$$

by $w_i(\boldsymbol{\beta}_P) = s_i^2(\boldsymbol{\beta}_P)$. The gradient of $L_{\varepsilon,\Delta}(\boldsymbol{\beta}_P)$ with respect to $\boldsymbol{\beta}_P$ is

$$\nabla L_{\varepsilon,\Delta}(\boldsymbol{\beta}_P) = 2\mathbf{K}_P^T \mathbf{W}(\boldsymbol{\beta}_P) \mathbf{r}(\boldsymbol{\beta}_P) - 2\varepsilon \mathbf{K}_P^T \mathbf{s}(\boldsymbol{\beta}_P) + 2(\Delta - \varepsilon) \mathbf{K}_P^T \overline{\mathbf{s}}(\boldsymbol{\beta}_P) + 2\lambda \mathbf{K}_{PP} \boldsymbol{\beta}_P . \tag{22}$$

The generalized Hessian is

$$\nabla^2 L_{\varepsilon,\Delta}(\boldsymbol{\beta}_P) = 2\mathbf{K}_P^T \mathbf{W}(\boldsymbol{\beta}_P) \mathbf{K}_P + 2\lambda \mathbf{K}_{PP} . \tag{23}$$

The Newton step at the *k*-th iteration is given by

$$\boldsymbol{\beta}_P^{k+1} = \boldsymbol{\beta}_P^k - t\left(\nabla^2 L_{\varepsilon,\Delta}(\boldsymbol{\beta}_P^k) \right)^{-1} \nabla L_{\varepsilon,\Delta}(\boldsymbol{\beta}_P^k) . \tag{24}$$

The step size t can be found by a line search procedure that minimizes the one dimensional piecewise-smooth, convex quadratic function. Since the Newton step is much more expensive, the line search does not add to the complexity of the algorithm.

3.3 Computational Complexity

In SSVR-SRS, the most time-consuming operation is computing the Newton step (24). When a new basis function is added, it involves three main steps: computing the column $\mathbf{K}_s$, which is $O(n)$, computing the new elements of the generalized Hessian, which is $O(nm)$ and inverting the generalized Hessian that can be computed in an incremental manner [12], which is $O(m^2)$. When the active matrix $\mathbf{W}(\boldsymbol{\beta}_P)$ is changed, the inversion of the generalized Hessian needs to be updated again, which is $O(cm^2)$. In most cases, c is a small constant, so it is reasonable to consider $O(nm)$ as an expensive cost since $n \gg m$. Adding up these costs till m basis functions are chosen, we get an overall complexity of $O(nm^2)$.

4 Experiments

In this section, we evaluate the performance of SSVR-SRS on five benchmark data sets and compare them with SVM, the reduced set method and reduced SVM.

4.1 Experimental Details

SVR is constructed based on LIBSVM 2.82 where the second order information is used to select the working set. RSVM is implemented by our own Matlab code. The reduced set method determines the reduced vectors $\{\mathbf{z}_i\}_{i=1}^m$ and the corresponding expansion coefficients by minimizing

$$\left\| \mathbf{w} - \sum_{j=1}^{m} \alpha_j \phi(\mathbf{z}_j) \right\|^2, \tag{25}$$

where $\mathbf{w} = \sum_{i \in S} \beta_i \phi(\mathbf{x}_i)$ is the weight vector obtained by optimizing (5) and S is the index set of support vectors. Reduced set selection (RSS) is parallel to SSVR-SRS and determines a new basis function by

$$s = \arg\min_{j \in O \subset Q} \left(\frac{-\left(\left[\boldsymbol{\beta}_S^T, -\boldsymbol{\alpha}_P^T \right] \begin{bmatrix} \mathbf{K}_{Sj} \\ \mathbf{K}_{Pj} \end{bmatrix} \right)^2}{\left\| \left[\boldsymbol{\beta}_S^T, -\boldsymbol{\alpha}_P^T \right] \right\|_2^2 k(\mathbf{x}_j, \mathbf{x}_j)} \right). \tag{26}$$

Five benchmark data sets: Abalone, Bank8fh, Bank32fh, House8l and Friedman3 are used in our empirical study. Information on these benchmark data sets is summarized in Table 2. These data sets have been extensively used in testing the performance of diversified kinds of learning algorithms. The first four data sets are available from Torgo's homepage: http://www.liacc.up.pt/~ltorgo/Regression/DataSets.html. Friedman3 is from [19]. The noise is adjusted for a 3:1 signal-to-noise ratio.

All the experiments were run on a personal computer with 2.4 GHz P4 processors, 2 GB memory and Windows XP operation system. Gaussian kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right)$ is used to construct non-linear SVR. The free parameters in the algorithms are determined by 10-fold cross validation except that Δ in the

Table 2. Information on benchmark data sets

Problem	Training	Test	Attribute	m
Abalone	3000	1177	8	50
Bank8fh	5000	4192	8	50
Bank32h	5000	4192	32	150
House8l	15000	7784	8	300
Friedman3	30000	20000	4	240

insensitive Huber loss function is fixed to 0.3. For each training-test pair, the training samples are scaled into the interval [-1, 1], and the test samples are adjusted using the same linear transformation. For SSVR-SRS, RSS and RSVM, the final results are averaged over five random implementations.

4.2 Comparisons with LIBSVM 2.82

Table 3-4 reports the generalization performance and the number of basis functions of SVR and SSVR-SRS. As we can see, compared with SVR, SSVR-SRS achieves the impressive reduction in the number of basis functions almost without sacrificing the generalization performance.

Table 3. Test error and number of basis functions of SVR, SSVR-SRS on benchmark data sets. Error denotes root-mean-square test error, Std denotes the standard deviation of test error and NBF denotes the number of basis functions. For SSVR-SRS, λ is set to be 1e-2 on the first four data sets and 1e-3 on Friedman3 data set.

Error	SVR		SSVR-SRS		
2.107	Error	NBF	Error	Std	NBF
Abalone	2.106	1152	2.107	0.006783	18
Bank8fh	0.071	2540	0.071	0.000165	40
Bank32nh	0.082	2323	0.083	0.000488	83
House8l	30575	2866	30796	126.106452	289
Friedman3	0.115	9540	0.115	0.000211	203

Table 4. Test error and number of basis functions of SVR, SSVR-SRS on benchmark data sets. For SSVR-SRS, λ is set to be 1e-5.

Problem	SVR		SSVR-SRS		
	Error	NBF	Error	Std	NBF
Abalone	2.106	1152	2.106	0.012109	17
Bank8fh	0.071	2540	0.071	0.000259	44
Bank32nh	0.082	2323	0.083	0.000183	119
House8l	30575	2866	30967	219.680790	282
Friedman3	0.115	9540	0.115	0.000318	190

4.3 Comparisons with RSVM and RSS

Figure 1-5 compare SSVR-SRS, RSVM and RSS on the five data sets. Overall, SSVR-SRS beats its competitors and achieves the best performance in terms of the decrease of test error with the number of basis functions. In most cases, RSVM is inferior to RSS, especially in the early stage. An exception is House8l data set where RSVM gives smaller test error than RSS when the number of basis functions is beyond some threshold value. SSVR-SRS significantly outperforms RSS on Bak32nh, House8l and Friedman3 data sets, but the difference between them becomes very small on the remaining data sets. SSVR-SRS is significantly superior to RSVM on

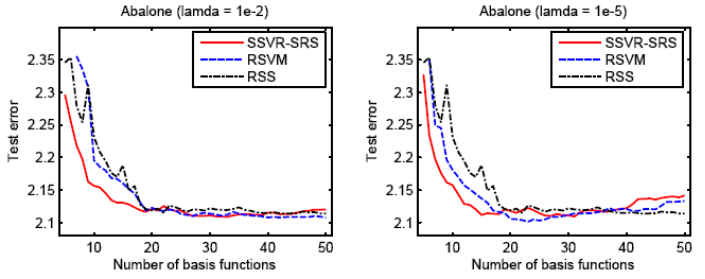


Fig. 1. Comparisons of SSVR-SRS, RSVM and RSS on Abalone

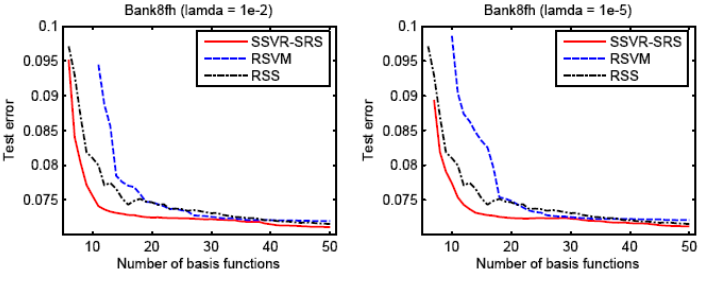


Fig. 2. Comparisons of SSVR-SRS, RSVM and RSS on Bank8fh

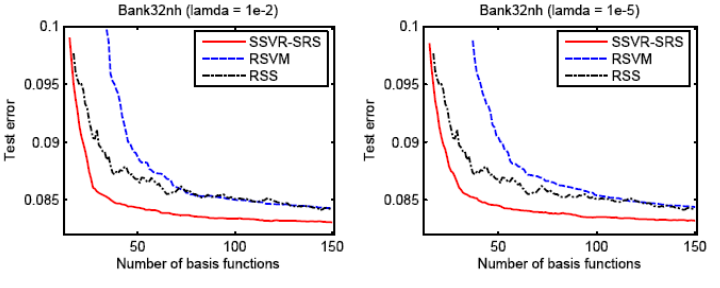


Fig. 3. Comparisons of SSVR-SRS, RSVM and RSS on Bank32nh

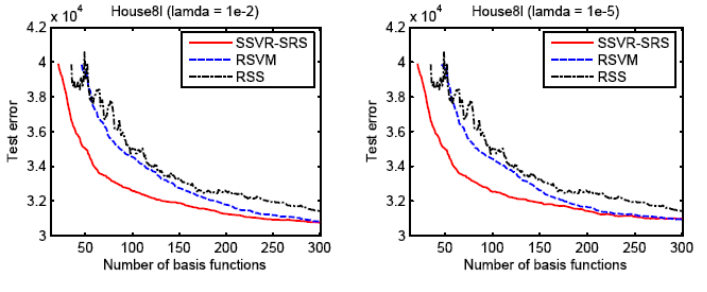


Fig. 4. Comparisons of SSVR-SRS, RSVM and RSS on House81

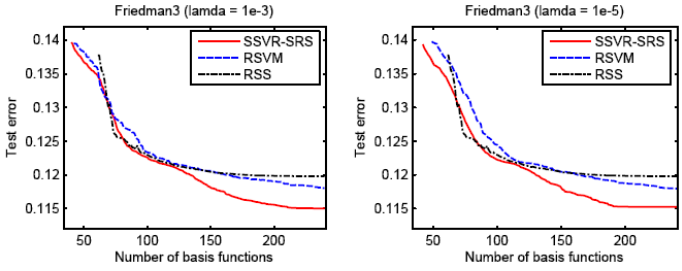


Fig. 5. Comparisons of SSVR-SRS, RSVM and RSS on Friedman3

four of the five data sets and comparable on the remaining data set. Another observation from Figure 1-5 is that SSVR-SRS with small regularization parameter starts over-fitting earlier than that with large regularization parameter, e.g. Abalone data set.

One phenomenon to note is that the reduced set selection has a large fluctuation in the generalization performance in the early stage. This is because the fact that, the different components of the weight vector $\mathbf{W}$ usually have a different impact on the generalization performance and therefore the better approximation to $\mathbf{W}$ does not necessarily leads to the better generalization performance. The fluctuation is alleviated with the increasing number of basis functions because the large number of basis functions can guarantee that each component of $\mathbf{W}$ is approximated well.

4.4 Training Time of SSVR-SRS

We do not claim that SSVR-SRS is more efficient than some state-of-the-art training decomposition algorithms such as SMO. Our main motivation is to point out that there is a way that can efficiently build a highly sparse SVR with the guaranteed generalization performance. In practice, depending on the number of basis functions, SSVR-SRS can be faster or slower than the decomposition algorithms. It is not fair to directly compare the training time of our algorithm with that of LIBSVM 2.82 since our algorithm is implemented by Matlab and however LIBSVM 2.82 by C++. But, we still list the training time in Table 5 as a rough reference.

Table 5. Training time of four algorithms on benchmark data sets

Problem	SSVR-SRS	RSVM	LIBSVM2.82	RSS
Abalone	5.73	2.59	1.70	2.85
Bank8fh	7.39	4.61	8.03	9.65
Bank32h	47.63	31.03	17.55	24.76
House8l	416.92	391.47	98.38	118.79
Fiedman3	565.59	462.57	1237.19	1276.42

5 Concluding Remarks

We have presented SSVR-SRS for building sparse support vector regression. Our method has three key advantages: (1) it directly approximates the primal objective

function and is more reasonable than the post-processing methods; (2) it scales well with the number of training samples and can be applied to large scale problems; (3) it simultaneously considers the sparseness and generalization performance of the resulting learner.

This work was supported by the Graduate Innovation Fund of Xidian University (No. 05004).

References

1. Vapnik, V: Statistical Learning Theory. New York Wiley-Interscience Publication (1998)
2. Steinwart, I. Sparseness of support vector machines. Journal of Machine Learning Research 4 (2003) 1071–1105
3. Burges, C. J. C. and Schölkopf, B. Improving the accuracy and speed of support vector learning machines. Advances in Neural Information Processing System 9 (1997) 375-381
4. Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., K. Muller, K. R., Raetsch, G., and Smola, A. J. Input space vs. feature space in kernel-based methods. IEEE Transactions on Neural Networks 10 (1999) 1000-1017
5. Lee, Y. J. and Mangasarian, O. L. RSVM: Reduced support vector machines. In Proceedings of the SIAM International Conference on Data Mining. SIAM, Philadelphia (2001)
6. Joachims, T. Making large-scale SVM learning practical. In Advances in Kernel Methods - Support Vector Learning, MIT Press, Cambridge, Massachussetts (1999)
7. Platt, J. Sequential minimal optimization: a fast algorithm for training support vector machines. In Advance in Kernel Methods - Support Vector Learning, MIT Press, Cambridge, Massachussetts (1999)
8. Mangasarian, O. L. A finite Newton method for classification. Optimization Methods & Software 17(5) (2002) 913-929
9. Keerthi, S. S. and Decoste D. M. A modified finite Newton method for fast solution of large scale linear svms. Journal of Machine Learning Research 6 (2005) 341-361
10. Chapelle, O. Training a Support Vector Machine in the Primal. Neural Computation (2006) (Accepted)
11. Bo, L. F., Wang, L. and Jiao L. C. Recursive finite Newton algorithm for support vector regression in the primal. Neural Computation (2007), in press.
12. Keerthi, S. S., Chapelle, O., and Decoste D. Building Support Vector Machines with Reduced Classifier Complexity. Journal of Machine Learning Research 7 (2006) 1493-1515
13. Vincent, P. and Bengio, Y. Kernel matching pursuit. Machine Learning 48 (2002) 165-187
14. Fan, R. E., Chen P. H., and Lin C. J. Working Set Selection Using Second Order Information for Training Support Vector Machines. Journal of Machine Learning Research 6 (2005) 1889-1918
15. Kimeldorf, G. S. and Wahba G. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. Annals of Mathematical Statistics 41 (1970) 495-502
16. Huber, P. Robust Statistics. John Wiley, New York (1981)
17. Mallat, S. and Zhang, Z. Matching pursuit with time-frequency dictionaries. IEEE Transactions on Signal Processing 41(12) (1993) 3397-3415
18. Friedman, J. Greedy Function Approximation: a Gradient Boosting Machine. Annals of Statistics 29 (2001) 1189-1232
19. Friedman, J. Multivariate adaptive regression splines. Annals of Statistics 19(1) (1991) 1-141

Mining Frequent Itemsets from Uncertain Data

Chun-Kit Chui[1], Ben Kao[1], and Edward Hung[2]

[1] Department of Computer Science, The University of Hong Kong, Pokfulam, Hong Kong
{ckchui,kao}@cs.hku.hk
[2] Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong
csehung@comp.polyu.edu.hk

Abstract. We study the problem of mining frequent itemsets from *uncertain data* under a *probabilistic framework*. We consider transactions whose items are associated with existential probabilities and give a formal definition of frequent patterns under such an uncertain data model. We show that traditional algorithms for mining frequent itemsets are either inapplicable or computationally inefficient under such a model. A *data trimming* framework is proposed to improve mining efficiency. Through extensive experiments, we show that the data trimming technique can achieve significant savings in both CPU cost and I/O cost.

1 Introduction

Association analysis is one of the most important data-mining model. As an example, in market-basket analysis, a dataset consists of a number of tuples, each contains the items that a customer has purchased in a transaction. The dataset is analyzed to discover associations among different items. An important step in the mining process is the extraction of frequent itemsets, or sets of items that co-occur in a major fraction of the transactions. Besides market-basket analysis, frequent itemsets mining is also a core component in other variations of association analysis, such as association-rule mining [1] and sequential-pattern mining [2].

All previous studies on association analysis assume a data model under which *transactions* capture doubtless facts about the items that are contained in each transaction. In many applications, however, the existence of an item in a transaction is best captured by a likelihood measure or a probability. As an example, a medical dataset may contain a table of patient records (tuples), each of which contains a set of symptoms and/or illnesses that a patient suffers (items). Applying association analysis on such a dataset allows us to discover any potential correlations among the symptoms and illnesses. In many cases, symptoms, being subjective observations, would best be represented by probabilities that indicate

This research is supported by Hong Kong Research Grants Council Grant HKU 7134/06E.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 47–58, 2007.

Table 1. A diagnosis dataset

Patient ID	Depression	Eating Disorder
1	90%	80%
2	40%	70%

their presence in the patients' tuples. Table 1 shows an example patient dataset. A probability value in such a dataset might be obtained by a personal assessment conducted by a physician, or it could be derived based on historical data statistics. (For example, a patient who shows positive reaction to Test A has a 70% probability of suffering from illness B.) Another example of uncertain datasets is pattern recognition applications. Given a satellite picture, image processing techniques can be applied to extract features that indicate the presence or absence of certain target objects (such as bunkers). Due to noises and limited resolution, the presence of a feature in a spatial area is often uncertain and expressed as a probability [3]. Here, we can model a spatial region as an object, and the features (that have non-zero probabilities of being present in a region) as the items of that object. The dataset can thus be considered as a collection of tuples/transactions, each contains a set of items (features) that are associated with the probabilities of being present. Applying association analysis on such a dataset allows us to identify closely-related features. Such knowledge is very useful in pattern classification [4] and image texture analysis [5].

In this paper we consider datasets that are collections of transactional records. Each record contains a set of items that are associated with *existential* probabilities. As we have mentioned, a core step in many association analysis techniques is the extraction of frequent itemsets. An itemset is considered *frequent* if it appears in a large-enough portion of the dataset. The occurrence frequency is often expressed in terms of a support count. For datasets that contain uncertain items, however, the definition of support needs to be redefined. As we will discuss later, due to the probabilistic nature of the datasets, the occurrence frequency of an itemset should be captured by an *expected* support instead of a traditional support count. We will explain the *Possible Worlds* interpretation of an uncertain dataset [6] and we will discuss how expected supports can be computed by a simple modification of the well-known *Apriori* algorithm [1].

Since the existence of an item in a transaction is indicated by a probability, an advantage of the existential uncertain data model is that it allows more information to be captured by the dataset. Consider again the example patient dataset. If we adopt a binary data model, then each symptom/illness can either be present (1) or absent (0) in a patient record. Under the binary model, data analysts will be forced to set a threshold value for each symptom/illness to quantize the probabilities into either 1 or 0. In other words, information about those (marginally) low values is discarded. The uncertain data model, however, allows such information be retained and be available for analysis. The disadvantage of retaining such information is that the size of the dataset would be much larger

than that under the quantized binary model. This is particularly true if most of the existential probabilities are very small. Consequently, mining algorithms will run a lot slower on such large datasets. In this paper we propose an efficient technique for mining existential uncertain datasets, which exploit the statistical properties of low-valued items. Through experiments, we will show that the proposed technique is very efficient in terms of both CPU cost and I/O cost.

The rest of this paper is organized as follows. Section 2 describes the Possible Worlds interpretation of existential uncertain data and defines the *expected support* measure. Section 3 discusses a simple modification of the *Apriori* algorithm to mine uncertain data and explains why such a modification does not lead to an efficient algorithm. Section 4 presents a *data trimming* technique to improve mining efficiency. Section 5 presents some experimental results and discusses some observations. We conclude the study in Section 6.

2 Problem Definition

In our data model, an uncertain dataset D consists of d transactions $t_1, \ldots, t_d$. A transaction t_i contains a number of items. Each item x in t_i is associated with a *non-zero* probability $P_{t_i}(x)$, which indicates the likelihood that item x is present in transaction t_i. There are thus two possibilities of the world. In one case, item x is present in transaction t_i; in another case, item x is not in t_i. Let us call these two possibilities the two possible worlds, W_1 and W_2, respectively. We do not know which world is the real world but we do know, from the dataset, the probability of each world being the true world. In particular, if we let $P(W_i)$ be the probability that world W_i being the true world, then we have $P(W_1) = P_{t_i}(x)$ and $P(W_2) = 1 - P_{t_i}(x)$. We can extend this idea to cover cases in which transaction t_i contains other items. For example, let item y be another item in t_i with probability $P_{t_i}(y)$. If the observation of item x and item y are independently done[1], then there are four possible worlds. The probability of the world in which t_i contains both items x and y, for example, is $P_{t_i}(x) \cdot P_{t_i}(y)$. We can further extend the idea to cover datasets that contains more than one transaction. Figure 1 illustrates the 16 possible worlds derived from the patient records shown in Table 1. In traditional frequent itemset mining, the support count of an itemset X is defined as the number of transactions that contain X. For an uncertain dataset, such a support value is undefined since we do not know in the real world whether a transaction contains X with certainty. We can, however, determine the support of X with respect to any given possible world. Let us consider the worlds shown in Figure 1, the supports of itemset AB in world W_1 and W_6 are 2 and 1, respectively. If we can determine the probability of each possible world and the support of an itemset X in each world, we can determine the *expected support* of X.

Definition 1. *An itemset X is frequent if and only if its expected support not less than $\rho_s \cdot d$, where ρ_s is a user-specified support threshold.*

[1] For example, we can consider that different symptoms are diagnosed by independent medical tests.

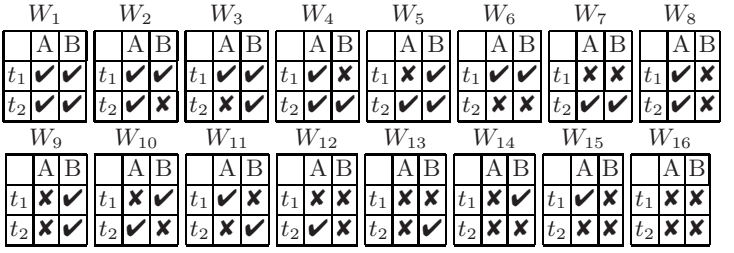


Fig. 1. 16 Possible Worlds derived from dataset with 2 transactions and 2 items

Given a world W_i and an itemset X, let us define $P(W_i)$ be the probability of world P_i and $S(X, W_i)$ be the support count of X in world W_i. Furthermore, we use $T_{i,j}$ to denote the set of items that the jth transaction, i.e., t_j, contains in the world W_i. If we assume that items' existential probabilities in transactions are determined through independent observations[2], then $P(W_i)$ and the expected support $S_e(X)$ of X are given by the following formulae:

$$P(W_i) = \prod_{j=1}^{d} \left(\prod_{x \in T_{i,j}} P_{t_j}(x) \cdot \prod_{y \notin T_{i,j}} (1 - P_{t_j}(y)) \right), \text{ and} \tag{1}$$

$$S_e(X) = \sum_{i=1}^{|W|} P(W_i) \times S(X, W_i). \tag{2}$$

where W is the set of possible worlds derived from an uncertain dataset D.

Computing $S_e(X)$ according to Equation 2 requires enumerating all possible worlds and finding the support count of X in each world. This is computationally infeasible since there are 2^m possible worlds where m is the total number of items that occur in all transactions of D. Fortunately, we can show that

$$S_e(X) = \sum_{j=1}^{|D|} \prod_{x \in X} P_{t_j}(x). \tag{3}$$

Thus, $S_e(X)$ can be computed by a single scan through the dataset D.

Proof. Let $S^{t_j}(X, W_i)$ be the support of X in transaction t_j w.r.t. possible world W_i. If $X \subseteq T_{i,j}$, $S^{t_j}(X, W_i) = 1$; otherwise, $S^{t_j}(X, W_i) = 0$.

$$\begin{aligned} S_e(X) &= \sum_{i=1}^{|W|} P(W_i) S(X, W_i) = \sum_{i=1}^{|W|} P(W_i) \sum_{j=1}^{|D|} S^{t_j}(X, W_i) \\ &= \sum_{j=1}^{|D|} \sum_{i=1}^{|W|} P(W_i) S^{t_j}(X, W_i) = \sum_{j=1}^{|D|} \sum_{X \subseteq T_{i,j}} P(W_i) = \sum_{j=1}^{|D|} \prod_{x \in X} P_{t_j}(x). \end{aligned}$$

[2] For example, the existential probabilities of two symptoms of the same patient are determined independently by two lab tests.

3 Preliminaries

Most of the algorithms devised to find frequent patterns (or itemsets) from conventional transaction datasets are based on the *Apriori* algorithm [1]. The algorithm relies on a property that all supersets of an infrequent itemset must not be frequent. Apriori operates in a bottom-up and iterative fashion. In the k^{th} iteration, the *Apriori-Gen* procedure generates all size-k candidate itemsets C^k and uses a *Subset-Function* procedure to verify their support counts. Candidate itemsets with support counts larger than a user-specified support threshold are regarded as frequent. The set of frequent k-itemsets L^k is then used by the Apriori-Gen procedure to generate candidates for next iteration. The algorithm terminates when C^{k+1} is empty.

Under our uncertainty model, the Subset-Function procedure has to be revised such that it can obtain the expected support count of each candidate. In the traditional Apriori algorithm, Subset-Function processes one transaction at a time by enumerating all size-k subsets contained in the transaction in the k^{th} iteration. The support count of a candidate is incremented by 1 if it is in C^k. By Equation 3, we will instead increment the expected support count by the product of the existential probabilities of all items $x \in X$. This modified algorithm is called the **U-Apriori** algorithm.

Inherited from the Apriori algorithm, U-Apriori does not scale well on large datasets. The poor efficiency problem becomes more serious under uncertain datasets, as mentioned in Section 1, in particular when most of the existential probabilities are of low values. Let us consider a transaction t containing three items A, B and C with existential probabilities $5\%, 0.5\%$ and 0.1%, respectively. In the Subset-Function procedure, the product of the probabilities ($0.05 \times 0.005 \times 0.001 = 0.00000025$) will be computed and the support count of candidate $\{ABC\}$ will be retrieved. By Equation 3, the support count of candidate $\{ABC\}$ should be incremented by 0.00000025 which is insignificantly small. If most of the existential probabilities are small, such insignificant increments will dominate the Subset-Function procedure and waste computational resources since in most cases an infrequent candidate will not be recognized as infrequent until most of the transactions are processed.

To illustrate the impact of items with low existential probabilities on the performance of U-Apriori, we conducted a preliminary experiment on five datasets. The datasets have the same set of frequent itemsets but are fine tuned to have different percentages of items with low existential probabilities. Let R be the percentage of items with low probabilities in a dataset. In the five datasets, R is set as $0\%, 33.3\%, 50\%, 66.6\%$ and 75% respectively. [3] In Figure 2a, we see that U-Apriori takes different amount of time to execute even though all datasets contain the same sets of frequent itemsets. We can conclude that when there are more items with low existential probabilities (larger R), U-Apriori becomes more inefficient. This result also indicates that by reducing the number of insignificant candidate increments, we might be able to reduce the execution time on all

[3] Please refer to Section 5 for the details of our data generation process.

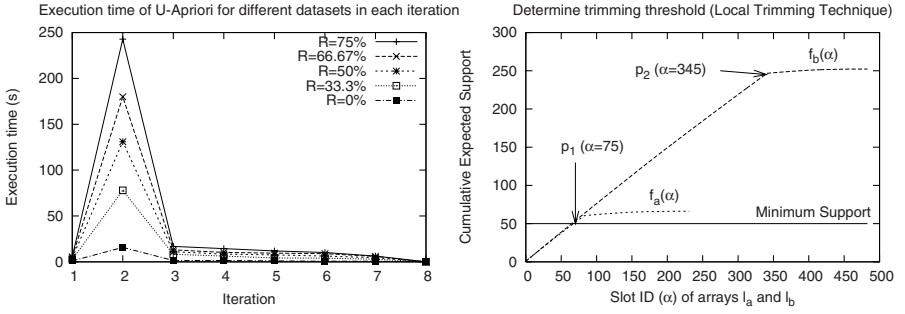


Fig. 2. a) The execution time of U-Apriori in different datasets. **b)** Cumulative expected support of the sorted arrays l_a and l_b of items a and b.

datasets to the time of the dataset with $R = 0\%$. This motivates our study on an efficient technique called *Data trimming* by exploiting the statistical properties of those items with low existential probabilities.

4 Data Trimming

To improve the efficiency of the U-Apriori algorithm, we propose a *data trimming* technique to avoid insignificant candidate support increments performed in the Subset-Function. The basic idea is to trim away items with low existential probabilities from the original dataset and to mine the trimmed dataset instead. Hence the computational cost of those insignificant candidate increments can be reduced. In addition, the I/O cost can be greatly reduced since the size of the trimmed dataset is much smaller than the original one.

Data Trimming Framework. More specifically, the data trimming technique works under a framework that consists three modules: the *trimming module*,

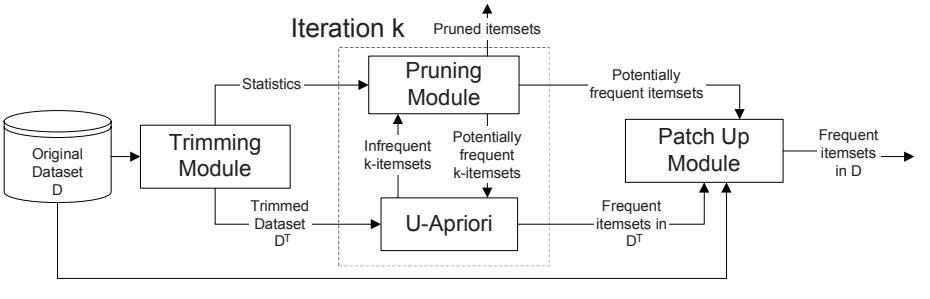


Fig. 3. The Data Trimming Framework

pruning module and *patch up module*. As shown in figure 3, the mining process starts by passing an uncertain dataset D into the *trimming module*. It first obtains the frequent items by scanning D once. A trimmed dataset D^T is constructed by removing the items with existential probabilities smaller than a trimming threshold ρ_t in the second iteration. Depending on the trimming strategy, ρ_t can be either global to all items or local to each item. Some statistics such as the maximum existential probability being trimmed for each item is kept for error estimation.

D^T is then mined by U-Apriori. Notice that if an itemset is frequent in D^T, it must also be frequent in D. On the other hand, if an itemset is infrequent in D^T, we cannot conclude that it is infrequent in D.

Definition 2. *An itemset X is potentially frequent if $S_e^T(X) \leq d\rho_s \leq S_e^T(X) + e(X)$ where $S_e^T(X)$ is the expected support of X in D^T and $e(X)$ is the upper bound of the error estimated for $S_e^T(X)$.*

Lemma 1. *An itemset X cannot be frequent if $S_e^T(X) + e(X) < d\rho_s$.*

The role of the *pruning module* is to estimate the upper bound of the mining error $e(X)$ by the statistics gathered from the *trimming module* and to prune the itemsets which cannot be frequent in D according to Lemma 1. After mining D^T, the expected supports of the frequent and potentially frequent itemsets are verified against the original dataset D by the *patch up module*.

A number of trimming, pruning and patch up strategies can be used under this framework. Due to limitation of space, we only present a simple method, called the *Local trimming, Global pruning and Single-pass patch up strategy* (the **LGS-Trimming** strategy), in this paper.

Local Trimming Strategy. The *Local trimming* strategy uses one trimming threshold $\rho_t(x)$ for each item x. $\rho_t(x)$ can be determined based on the distribution of existential probabilities of x in D. The distribution can be obtained by sorting the existential probabilities of x in D in descending order and putting them in an array l_x. We then plot the curve $f_x(\alpha) = \sum_{i=0}^{\alpha} l_x[i]$ where the y-axis is the cumulative sum of the probabilities $\sum_{i=0}^{\alpha} l_x[i]$ and the x-axis is the slot ID of l_x. Figure 2b shows the curves $f_a(\alpha)$ and $f_b(\alpha)$ of two hypotheoretical items a and b. The horizontal line labeled "minsup" is the minimum support threshold.

We regard item a as *marginally frequent* because $S_e(a)$ exceeds the minimum support by a small fraction (e.g. $d\rho_s \leq S_e(a) \leq 1.1 \times d\rho_s$). Assume $f_a(\alpha)$ intersects with the minimum support line at about $\alpha = i$. In this case, the Local trimming strategy sets the trimming threshold $\rho_t(a)$ to be $l_x[i]$, which is the existential probability of the item at the ith slot of the array l_x (i.e. $\rho_t(a) = l_a[75]$). The rationale is that the supersets of a are likely to be infrequent, therefore those insignificant candidate increments with existential probabilities smaller than $\rho_t(a)$ are likely to be redundant.

On the other hand, we classify item b as another type of items as $S_e(b) \gg d\rho_s$. The Local trimming strategy determines $\rho_t(b)$ based on the change of slope of $f_b(\alpha)$. In this case, since the chance of the supersets of b to be frequent is larger,

we adopt a more conservative approach. We use point p_2 in Figure 2b as a reference point to determine $\rho_t(b)$ (i.e. $\rho_t(b) = l_b[345]$)[4]. The reason is that if one of the supersets of b is actually infrequent, the error would be small enough for the Pruning module to obtain a tight estimation and identify it as an infrequent itemset by Lemma 1.

Global Pruning Strategy. We illustrate the *Global pruning* strategy by an example in the second iteration. Let $M^T(x)$ be the maximum of the existential probabilities of those untrimmed item x, and similarly $M^{\sim T}(x)$ for those trimmed x. We also let $S_e^T(x)$ be the sum of the existential probabilities of those untrimmed x, and similarly $S_e^{\sim T}(x)$ for those trimmed x. If an itemset {AB} is infrequent in D^T (i.e. $S_e^T(AB) < d\rho_s$), we can obtain the upper bound of the error $e(AB)$ by the following formula:

$$e(AB) = \hat{S}_e^{T,\sim T}(AB) + \hat{S}_e^{\sim T,T}(AB) + \hat{S}_e^{\sim T,\sim T}(AB). \tag{4}$$

where $\hat{S}_e^{T,\sim T}(AB)$ is an upper bound estimation of the expected support of {AB} for all transactions t with $P_t(A) \geq \rho_t(A)$ and $P_t(B) < \rho_t(B)$.

If we assume all the untrimmed items A exist with maximum existential probability $M^T(A)$, then the maximum number of transactions with an untrimmed item A which may coexist with a trimmed item B is given by $\frac{S_e^T(A) - S_e^T(AB)}{M^T(A)}$. On the other hand, if we assume all the trimmed items B exist with maximum probability $M^{\sim T}(B)$, then the maximum number of transactions with a trimmed item B is given by $\frac{S_e^{\sim T}(B)}{M^{\sim T}(B)}$. Therefore, we can obtain $\hat{S}_e^{T,\sim T}(AB)$ as shown in Equation 5. $\hat{S}_e^{\sim T,T}(AB)$ is similarly obtained.

$\hat{S}_e^{\sim T,\sim T}(AB)$ is an upper bound estimation of the expected support of {AB} for all transactions t with $P_t(A) < \rho_t(A)$ and $P_t(B) < \rho_t(B)$, assuming that the case of estimating $\hat{S}_e^{T,\sim T}(AB)$ and $\hat{S}_e^{\sim T,T}(AB)$ happens in D. It can be calculated by Equation 7 after obtaining $\hat{S}_e^{T,\sim T}(AB)$ and $\hat{S}_e^{\sim T,T}(AB)$.

$$\hat{S}_e^{T,\sim T}(AB) = min(\frac{S_e^T(A) - S_e^T(AB)}{M^T(A)}, \frac{S_e^{\sim T}(B)}{M^{\sim T}(B)}) \cdot M^T(A) \cdot M^{\sim T}(B). \tag{5}$$

$$\hat{S}_e^{\sim T,T}(AB) = min(\frac{S_e^{\sim T}(A)}{M^{\sim T}(A)}, \frac{S_e^T(B) - S_e^T(AB)}{M^T(B)}) \cdot M^{\sim T}(A) \cdot M^T(B). \tag{6}$$

$$\hat{S}_e^{\sim T,\sim T}(AB) = min(\frac{S_e^{\sim T}(A) - \hat{S}_e^{\sim T,T}(AB)}{M^{\sim T}(A)}, \frac{S_e^{\sim T}(B) - \hat{S}_e^{T,\sim T}(AB)}{M^{\sim T}(B)}) \cdot M^{\sim T}(A) \cdot M^{\sim T}(B). \tag{7}$$

Single-pass Patch Up Strategy. The *Single-pass patch up* strategy requires only one scan on the original dataset D. This strategy requires the *Apriori-Gen*

[4] Due to space limitation, we only present the abstract idea of Local trimming strategy in this paper.

procedure to include the potentially frequent itemsets during the mining process so that the set of potentially frequent itemsets will not miss any real frequent itemsets. In the patch up phase, the true expected supports of potentially frequent itemsets are verified by a single scan on the original dataset D. At the same time, the true expected supports of frequent itemsets in D^T are also recovered.

5 Experimental Evaluation

We ran our experiments on Linux Kernel version 2.6.10 machine with 1024 MB of memory. The U-Apriori algorithm and the LGS-Trimming technique were implemented using C programming language.

Data were generated in the following two-step procedure. First we generate data without uncertainty using the IBM synthetic generator used in [1]. This step is to generate dataset which contains frequent itemsets. We set the average number of items per transaction (T_{high}) to be 20, the average length of frequent itemsets (I) to be 6 and the number of transactions (D) to be 100K [5].

In the second step, we introduce uncertainty to each item of the dataset generated from the first step. Since we want to maintain the frequent patterns hidden in the dataset, we assign each items with relatively high probabilities following a normal distribution with specified mean HB and standard deviation HD. To simulate items with low probabilities, we add T_{low} number of items into each transaction. These items have probabilities in normal distribution with mean LB and standard deviation LD. Therefore, the average number of items per transaction, denoted as T, is equal to $T_{high} + T_{low}$. We use R to denote the percentage of items with low probabilities in the dataset (i.e. $R = \frac{T_{low}}{T_{high}+T_{low}}$).

As an example, $T80R75I6D100KHB90HD5LB10LD6$ represents an uncertain dataset with 80 items per transaction on average. Of the 80 items, 20 items are assigned with high probabilities and 60 items are assigned with low probabilities. The high(low) probabilities are generated following normal distribution with mean equal to 90%(10%) and standard deviation equal to 5%(6%). For simplicity, we call this dataset *Synthetic-1* in later sections.

5.1 Varying Number of Low Probability Items Per Transaction

We first investigate the CPU cost of U-Apriori and LGS-Trimming on datasets with different number of low probability items per transaction. We keep the same set of frequent itemsets in all datasets, therefore an increase in R means more low-probability items are added during the second step of data generation. We set $\rho_s = 0.5\%$ in the experiments. Figure 4a and 4b show the CPU cost and the percentage of CPU cost saving (compare with U-Apriori) of U-Apriori and LGS-Trimming as R varies from 0% to 90%.

[5] We have conducted our experiments using different values of T_{high}, I and D but due to the space limitation we only report a representative result using $T_{high}20I6D100K$ in this paper.

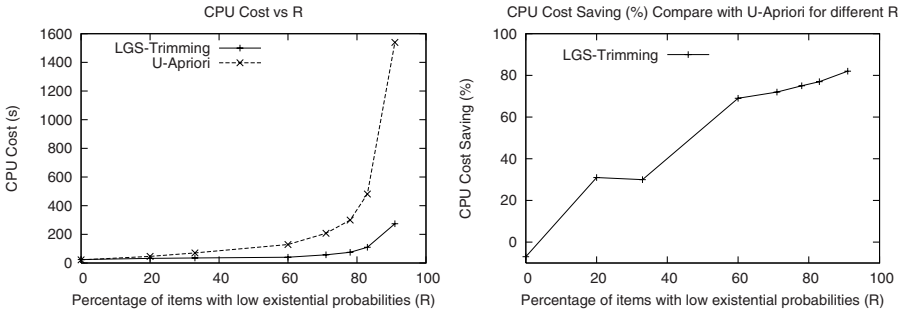


Fig. 4. CPU cost and saving with different R

From Figures 4a, we observe that the CPU cost of U-Apriori increases exponentially with the percentage of low probability items in the dataset. This is mainly due to the combinatorial explosion of subsets contained in a transaction. This leads to huge amounts of insignificant candidate support increments in the Subset-Function. For instance, when R is 90%, the average number of items per transaction with non-zero probability is about 200 (20 items with high probabilities, 180 items with low probabilities), leading to ${}_{200}C_2 = 19900$ size-2 subsets per transaction. In other words, there are 19900 candidate searches per transaction in the second iteration. When R is 50%, however, there are only ${}_{40}C_2 = 780$ candidate searches per transaction in the second iteration.

From Figure 4b, we see that the LGS-Trimming technique achieves positive CPU cost saving when R is over 3%. It achieves more than 60% saving when R is 50% or larger. When R is too low, fewer low probability items can be trimmed and the saving cannot compensate with the extra computational effort in the patch up phase. These figures suggest that the LGS-Trimming technique is very scalable to the percentage of low probability items in the dataset.

5.2 Varying Minimum Support Threshold

This section assesses the performance of U-Apriori and LGS-Trimming by varying ρ_s from 1% to 0.1%. Here we only report the result of using *Synthetic-1* in this experiment because experimental results on other datasets with different values of HB, HD, LB and LD also lead to a similar conclusion. Figures 5a and 5b show the CPU cost and the saving (in %) of the two mining techniques.

Figure 5a shows that LGS-Trimming outperforms U-Apriori for all values of ρ_s. Figure 5b shows that LGS-Trimming achieves very high and steady CPU cost saving ranging from 60% to 80%. The percentage of CPU cost saving increases gently when ρ_s increases because the low probability items become less significant to the support of itemsets when the support threshold increases. Therefore more low probability items can be trimmed, leading to better saving.

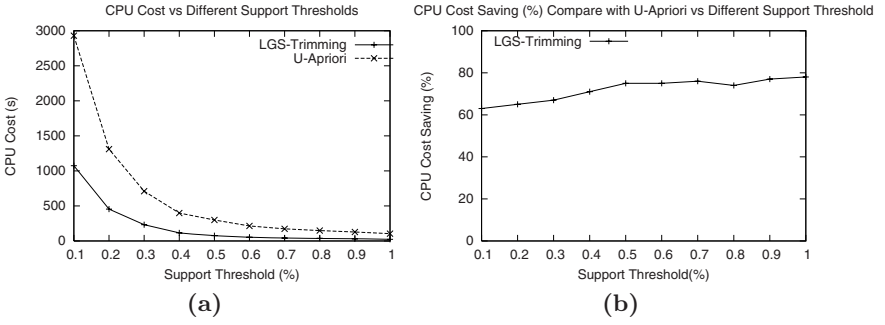


Fig. 5. CPU cost and saving with different ρ_s

5.3 CPU Cost and I/O Cost in Each Iteration

In this section we compare the CPU and I/O cost in each iteration of U-Apriori and LGS-Trimming. The dataset we use is *Synthetic-1* and we set $\rho_s = 0.5\%$. From Figure 6a, we see that the CPU cost of LGS-Trimming is smaller than U-Apriori from the second to the second last iteration. In particular, LGS-Trimming successfully relieves the computational bottleneck of U-Apriori and achieves over 90% saving in the second iteration. In the first iteration, the CPU cost of LGS-Trimming is slightly larger than U-Apriori because extra effort is spent on gathering statistics for the trimming module to trim the original dataset. Notice that iteration 8 is the patch up iteration which is the overhead of the LGS-Trimming algorithm. These figures show that the computational overhead of LGS-Trimming is compensated by the saving from the second iteration.

Figure 6b shows the I/O cost in terms of dataset scan (with respect to the size of the original dataset) in each iteration. We can see that I/O saving starts from iteration 3 to the second last iteration. The extra I/O cost in the second

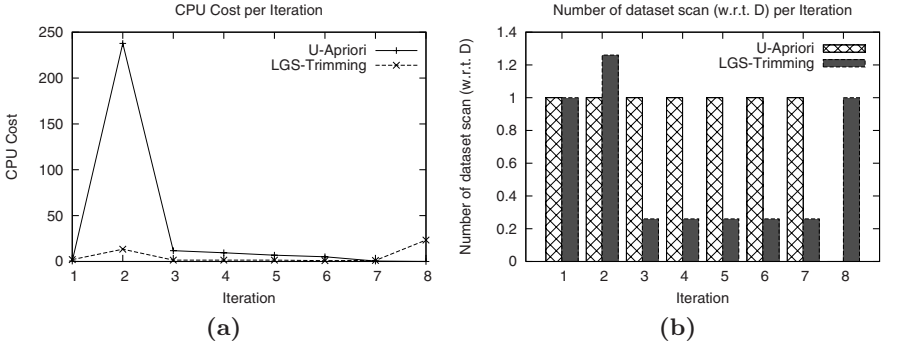


Fig. 6. CPU and I/O costs of U-Apriori and LGS-Trimming in each iteration

iteration is the cost of creating the trimmed dataset. In this case, LGS-Trimming reduces the size of the original dataset by a factor of 4 and achieves 35% I/O cost saving in total. As U-Apriori iterates k times to discover a size-k frequent itemset, longer frequent itemsets favors LGS-Trimming and the I/O cost saving will be more significant.

6 Conclusions

In this paper we studied the problem of mining frequent itemsets from existential uncertain data. We introduced the U-Apriori algorithm, which is a modified version of the Apriori algorithm, to work on such datasets. We identified the computational problem of U-Apriori and proposed a data trimming framework to address this issue. We proposed the LGS-Trimming technique under the framework and verified, by extensive experiments, that it achieves very high performance gain in terms of both computational cost and I/O cost. Unlike U-Apriori, LGS-Trimming works well on datasets with high percentage of low probability items. In some of the experiments, LGS-Trimming achieves over 90% CPU cost saving in the second iteration of the mining process, which is the computational bottleneck of the U-Apriori algorithm.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. of 20^{th} International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile, Morgan Kaufmann (1994) 487–499
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of the 11^{th} International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan, IEEE Computer Society (1995) 3–14
3. Dai, X., Yiu, M.L., Mamoulis, N., Tao, Y., Vaitis, M.: Probabilistic spatial queries on existentially uncertain data. In: SSTD. Volume 3633 of Lecture Notes in Computer Science., Springer (2005) 400–417
4. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: KDD. (1998) 80–86
5. Rushing, A., Ranganath, S., Hinke, H., Graves, J.: Using association rules as texture features. IEEE Trans. Pattern Anal. Mach. Intell. **23**(8) (2001) 845–858
6. Zimányi, E., Pirotte, A.: Imperfect information in relational databases. In: Uncertainty Management in Information Systems. (1996) 35–88

QC4 - A Clustering Evaluation Method

Daniel Crabtree, Peter Andreae, and Xiaoying Gao

School of Mathematics, Statistics and Computer Science
Victoria University of Wellington
New Zealand
daniel@danielcrabtree.com, pondy@mcs.vuw.ac.nz, xgao@mcs.vuw.ac.nz

Abstract. Many clustering algorithms have been developed and researchers need to be able to compare their effectiveness. For some clustering problems, like web page clustering, different algorithms produce clusterings with different characteristics: coarse vs fine granularity, disjoint vs overlapping, flat vs hierarchical. The lack of a clustering evaluation method that can evaluate clusterings with different characteristics has led to incomparable research and results. QC4 solves this by providing a new structure for defining general ideal clusterings and new measurements for evaluating clusterings with different characteristics with respect to a general ideal clustering. The paper describes QC4 and evaluates it within the web clustering domain by comparison to existing evaluation measurements on synthetic test cases and on real world web page clustering tasks. The synthetic test cases show that only QC4 can cope correctly with overlapping clusters, hierarchical clusterings, and all the difficult boundary cases. In the real world tasks, which represent simple clustering situations, QC4 is mostly consistent with the existing measurements and makes better conclusions in some cases.

1 Introduction

Comparing the performance of different clustering algorithms in some problem domains (i.e. web page clustering) has been problematic. Different algorithms produce clusterings with different characteristics: the clustering granularity may be coarse, so that there are just a few large clusters covering very broad topics, or fine, so that there are many small clusters of very focused topics; the clusters may be disjoint and constitute a partition of the results, or the clusters may overlap, so that the same page may appear in several clusters; the clustering may be "flat" so that all clusters are at the same level, or the clustering may be hierarchical so that lower-level clusters are sub-clusters of higher level clusters. As a result, many of the existing evaluation methods are biased towards algorithms that produce clusterings with certain characteristics. An evaluation method that fairly evaluates clusterings with different characteristics is needed; so that all clustering algorithms can be compared with a consistent method.

An example clustering domain is web page clustering, which helps users find relevant web pages by organizing the search result set from a search engine into clusters of semantically related pages. These clusters provide the user with an

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 59–70, 2007.

overview of the entire result set, and the clusters can be selected to filter the results or refine the query. Many clustering algorithms have been applied to web page clustering: K-means [1], Hierarchical Agglomerative Clustering [2], Link and Contents Clustering [3], Suffix Tree Clustering (STC) [4], Extended Suffix Tree Clustering (ESTC) [5], and Query Directed Clustering (QDC) [6]. A survey of clustering algorithms can be found in [7].

Many evaluation methods [1,4,7,8,9,10,11] are used to evaluate web clustering algorithms, but the results are often incomparable. There is probably no standard method because web page clustering algorithms produce clusterings that exhibit different characteristics, making web clustering an ideal application for an evaluation method that handles clusterings with different characteristics.

This paper proposes QC4, a new clustering evaluation method. Our preliminary research on QC4 was very briefly introduced in a short paper [12]. This paper further develops the full specifications of QC4, and evaluates it against existing measurements on synthetic test cases and real world web clustering tasks. QC4 allows clustering algorithms that produce clusterings with vastly different characteristics to be compared by generalizing the "gold-standard" approach to use a new structure for ideal clusterings and by developing new measures of quality and coverage. QC4 is currently targeted at web clustering, but is easily adapted to any domain where clusterings have different characteristics.

The next section discusses the related work. Section 3 describes and specifies QC4's richer ideal clustering structure and measurements. Section 4 evaluates QC4 by comparing it against the standard evaluation measurements using synthetic test cases and using nine clustering algorithms on four web page search result clustering tasks. Section 5 concludes the research and provides direction for future work.

2 Related Work

2.1 Approaches to Evaluation

There are two broad methodologies for evaluating clusterings. Internal quality [7,8] evaluates a clustering only in terms of a function of the clusters themselves. External quality [7,8] evaluates a clustering using external information, such as an ideal clustering. When external information is available, external quality is more appropriate because it allows the evaluation to reflect performance relative to the desired output.

There are three main approaches to evaluation using the external methodology: gold-standard [9], task-oriented [9], and user evaluation [4]. Gold-standard approaches manually construct an ideal clustering, which is then compared against the actual clustering. Task-oriented approaches evaluate how well some predefined task is solved. User evaluation approaches involve directly studying the usefulness for users and often involve observation, log file analysis, and user studies similar to those carried out in the user evaluation of Grouper [4].

Task-oriented methods have a bias towards the selected task. For example, search result reordering [4], which involves reordering the search results using the

clusters, has a bias towards small clusters, which tend to have higher quality. Randomly generating a perfect cluster of five pages is much more likely than generating a perfect cluster of fifty pages. In the extreme case of one cluster per page (singleton clustering), the clustering is evaluated as perfect, when clearly it is not.

User evaluation methods are very difficult to reproduce as they are dependent on the users. The large cost, and time involved in conducting good user evaluations is also a significant drawback. The lack of reproducibility, large cost, and time involved in conducting user evaluations makes them poor candidates for a standardized clustering evaluation method.

Therefore our evaluation method uses external information in the form of an ideal clustering to define a gold-standard and measures a clustering against this ideal clustering.

2.2 Measurements

This section discusses the measurements most commonly used to evaluate a clustering against an ideal clustering in the web clustering domain. We refer to the clusters of the ideal clustering as topics, to distinguish them from the clusters of the clustering being evaluated.

A perfect clustering matches the ideal clustering. A clustering can be less than perfect in two ways: some clusters may be of poor *quality* because they do not match any topics well, and the clustering may not include (*cover*) all the pages in the ideal clustering. There is often a tradeoff between quality and coverage, and algorithms can often be tuned to achieve one well at the cost of the other. Good overall evaluation methods must measure both factors.

The rest of the paper uses the following notation: C is a set of clusters, T is a set of topics (the clusters of the ideal clustering), and D is a set of pages. c, t, and d are individual elements of C, T, and D respectively. D_c is the pages in cluster c, D_t is the pages in topic t, and $D_{c,t}$ is the pages in both cluster c and topic t. C_d is the set of clusters containing page d and C_t is the set of clusters that best match topic t: $C_t = c_i | \operatorname{argmax}_{t_j}(D_{c_i,t_j}) = t$.

Precision and recall are common measurements used in information retrieval [13] for evaluation. The precision, $P(c,t)$, of a cluster relative to a topic is the fraction of the pages in the cluster that are also in the topic. Whereas the recall, $R(c,t)$, is the fraction of the pages in the topic that are in the cluster. The F-measure [1,8,11] combines precision and recall with equal weight on each.

$$P(c,t) = \text{Precision} = \frac{|D_{c,t}|}{|D_c|}$$
$$R(c,t) = \text{Recall} = \frac{|D_{c,t}|}{|D_t|}$$
$$F(c,t) = \text{F-measure} = \frac{2*P(c,t)*R(c,t)}{P(c,t)+R(c,t)}$$

Purity is the precision of a cluster relative to its best matching topic. Because the pages in a topic may be included in several clusters, recall is seldom used for clustering. However, we could define the recall of a topic to be the total

coverage of a topic among all clusters that best match that topic. F-measure is the f-measure of a cluster relative to its best matching topic.

$Purity(c) = \max_{t \in T}\{P(c,t)\}$
$Recall(t) = |\bigcup_{c \in C_t} D_{c,t}|/|D_t|$
$F(c) = \max_{t \in T}\{F(c,t)\}$

The Entropy and Mutual Information measures [1,8] are based on information theory [14]. The Entropy measure is the average "narrowness" of the distribution of the pages of a cluster among the topics. More precisely, it is the amount of information required to refine the cluster into the separate topics it represents. Mutual Information (MI) is an average of a measure of correspondence between each possible cluster topic pair.

$Entropy(c) = -\sum_{t \in T} P(c,t) log_{|T|} P(c,t)$
$MI = \frac{2}{|D|} \sum_{c \in C} \sum_{t \in T} |D_{c,t}| log_{|C||T|}(\frac{|D_{c,t}||D|}{|D_c||D_t|})$

Average Precision (average purity over clusters), Weighted Precision (cluster size weighted average purity over clusters), Average Entropy (average over clusters), and Weighted Entropy (cluster size weighted average over clusters) [1] can be used for overall quality evaluation. Average Recall (average over topics) and Weighted Recall (topic size weighted average over topics) [5] can be used for overall coverage evaluation. Mutual Information [8] and F (cluster size weighted average over clusters) provide overall measures that combine evaluation of quality and coverage.

Although the measurements are reasonable for some kinds of clusterings, they all have problems with overlapping clusters and hierarchical clusterings. Mutual information gives some non ideal clusterings better values than ideal clusterings. When the topics are of very different sizes, Weighted Precision, Weighted Entropy, and F give a high value for useless clusterings (such as a single cluster containing all pages). Average / Weighted Precision and Entropy only measure quality, and are maximized by a set of singleton clusters.

3 New Method - QC4

A fair clustering evaluation method should not inherently favor any particular algorithm. QC4 ensures this by minimizing the bias towards clusterings with particular characteristics (cluster granularity: coarse or fine, clustering structure: hierarchical or flat, disjoint or overlapping): if the bias towards the different possible characteristics of a clustering is minimized, then so is the bias towards the algorithms that produce those clusterings.

3.1 The Ideal Clustering

An ideal clustering is created by a human expert based on the pages to be clustered. The classical ideal clustering structure is a single level partition at a chosen granularity. QC4 uses a richer ideal clustering structure to describe clusterings with all kinds of characteristics.

QC4's ideal clustering structure is a hierarchy of topics, organised in levels, so that the set of topics at the top level represents a coarse categorisation of the pages, and the sets of topics at lower levels represent progressively finer categorisations. This allows QC4 to fairly compare algorithms that produce clusterings of different granularity and to compare algorithms that generate hierarchical clusterings.

Topics may overlap other topics (at the same and different levels), since real pages may belong to multiple topics. However, all pages must be contained in at least one topic at each level. This allows QC4 to evaluate algorithms that return overlapping clusters as well as algorithms that return partitions.

Since search engines often return outliers — pages that are unrelated to all the other pages — the hierarchy may contain a single outlier topic (present at every level) that contains all the outliers. The outlier topic must be disjoint from the other topics. QC4 handles outliers by not counting them when measuring coverage, and by removing clusters that contain a majority of outliers.

3.2 Quality and Coverage Measurements

The standard measures do not work well on hierarchical clusterings with overlapping clusters. Therefore, QC4 introduces four new measures of quality and coverage.

In addition to the notation in section 2.2, the rest of the paper uses the following notation: L is the set of levels from the topic hierarchy (eg, 1, 2, 3) and l is an individual level. T_l is the set of topics at level l, T_d is the set of topics containing page d, and $T_\emptyset$ is a set containing the outlier topic. $sub(t)$ is the set of all descendants of topic t. $lvl(t)$ is the lowest level of topic t.

Cluster Quality. Cluster Quality, $QU(c)$, is a measure of how closely a cluster matches a single topic. It is based on a modified entropy measure, $E(c)$.

The standard entropy measure of a cluster does not work with overlapping topics since pages in multiple topics are overcounted. There are two kinds of overlap: overlap of topics at different levels, and overlap of topics at the same level. Overlap between levels is handled by computing the entropy over the topics in a single level. QC4 chooses the level[1], $L(c)$, containing the topic that is the most similar to the cluster as measured by the f-measure.

$$L(c) = \text{cluster-level} = lvl(\text{argmax}_{t \in T \setminus T_\emptyset}\{F(c,t)\})$$
$$E(c) = \min_{t_b \in T_{L(c)}}\{-\sum_{t \in T_{L(c)}} P'(c,t,t_b) log_{|T_{L(c)}|} P'(c,t,t_b)\}$$

Overlap of topics at the same level is handled by computing a modified precision measure $P'(c,t,t_b)$. The modified measure removes the overcounting by temporarily removing pages in the "best" topic from the other topics, and then normalizing the precision to remove the effect of any other over counting.

$$P'(c,t,t_b) = \begin{cases} \frac{|D_{c,t}|}{|D_c|} & if\{t = t_b\} \\ \frac{(|D_c| - |D_{c,t_b}|)|D_{c,t} \setminus D_{c,t_b}|}{|D_c| \sum_{t' \in T_{L(c)} \setminus \{t_b\}} |D_{c,t'} \setminus D_{c,t_b}|} & otherwise \end{cases}$$

[1] If multiple topics maximize F, the one with lowest level is selected.

$E(c)$ measures how focused a cluster is on a single topic, choosing the appropriate level of granularity, and allowing both disjoint and overlapping topics to be handled fairly. However, it does not take cluster and topic size sufficiently into account and it does not recognize random clusters. To account for these, $E(c)$ is scaled down by a new measure that takes account of the cluster and topic size by $S_{recall}(c)$ and recognizes random clusters using $S_{random}(c)$.

$$QU(c) = (1 - E(c)) \min\{1, S_{recall}(c), S_{random}(c)\}$$

$E(c)$, being a precision/entropy based measure, gives a good value to focused clusters (all their pages belong to the same topic) regardless of the size of the clusters. However, very small clusters, even if they are highly focused, are not very useful to a user if they only contain a small fraction of the topic. To be useful, a cluster should be close to a topic by being both focused on the topic and by being of similar size to the topic. That is, the cluster should not only have good precision/entropy, but should also have good recall. QC4 scales down the quality measure of clusters that are much smaller than the topic that they are focused on by the recall measure. Since a page in a cluster may belong to multiple topics, the standard recall measure was modified to handle pages in multiple topics by averaging the recall of a cluster over all topics weighted by the modified precision $P'(c, t, t_b)$.

$$S_{recall}(c) = \max_{t_b \in T_{L(c)}} \{\sum_{t \in T_{L(c)}} P'(c, t, t_b) R'(c, t)\}$$

In the web page clustering domain, a cluster with low recall on a small topic is almost useless to the user. On the other hand, a cluster with the same low recall fraction of a very large topic will have more than enough pages for the user to understand the cluster and make an appropriate decision. Therefore, the recall measure can be modified by a non-linear function of the size of the topic to amplify the scaling for clusters focused on small topics.

$$R'(c, t) = 2^{\frac{R(c,t)-1}{R(c,t) log_2 |D_t|}}$$

Clusters that are similar to a random selection of pages from the result set provide almost no information, and will not be helpful to the user. Such a clustering should receive near zero quality. However, the modified entropy, $E(c)$, of randomly constructed clusters will generally not be the maximally bad value, especially if the topics are of varying sizes. QC4 uses a modified version of MI, $S_{random}(c)$, to scale down the quality measure of clusters that are similar to a random set of pages. $S_{random}(c)$ has to deal with overlapping topics in a single level, which it does by extracting the intersections of topics into temporary distinct topics and applying MI to the expanded, disjoint set of topics, $\rho(l)$. It also applies a threshold to ensure that only clusters that are very close to random or very small are scaled down. The resulting value is also normalized by the maximum MI to account for the varying maximum value of MI.

$$\rho(l) = \{r \subseteq D | ((\exists T_\alpha \subseteq T_l)(|r| > 0 \wedge r = \bigcap_{r' \in T_\alpha} D_{r'} - \bigcup_{r'' \in T_l \setminus T_\alpha} D_{r''})\}$$

$$S_{random}(c) = \frac{\sum_{r \in \rho(L(c))} |D_c \cap r| log_{|\rho(L(c))|} \frac{|D_c \cap r||D|}{|D_c||r|}}{0.05 \min_{t \in T_{L(c)} \setminus T_\emptyset} \{\sum_{r \in \rho(L(c))} |D_t \cap r| log_{|\rho(L(c))|} \frac{|D_t \cap r||D|}{|D_t||r|}\}}$$

Topic Coverage. Topic Coverage, $CV(t)$, is a measure of how well the pages in a topic are covered by the clusters. It is an average of the page coverage, $PC(d,t,l)$, of each of the pages in the topic. The coverage uses just level one topics because the page coverage already incorporates topics lower in the hierarchy.

$$CV(t) = \frac{\sum_{d \in D_t} PC(d,t,1)}{|D_t|}$$

A page in a topic is covered to some extent if any cluster contains the page. However, the user is unlikely to find a page if it is in a cluster that appears to be associated with a different topic, so a page will be better covered if it is contained in a cluster that matches a topic that the page is in. The better the match, the better the coverage. If a page is in topic t and cluster c, the precision $P(c,t)$ would be a good measure of how well the page is covered, as long as the page is not also in any other topics or clusters and the cluster is not merely a random selection of the pages. Both topics and clusters can overlap: a page may be in several topics and several clusters. In particular, each page in a top level topic will also be in subtopics of that topic at each level of the hierarchy. Therefore we need something more complicated than precision to measure page coverage.

QC4's page coverage measure considers all the clusters that a page is in, and also all the topics and subtopics the page is in. At each level of the topic hierarchy, it finds the average precision of the clusters that contain the page with respect to the best matching subtopics containing the page. It then recursively computes the maximum of this measure at each level to compute a page coverage measure over the whole hierarchy.

$$PC(d,t,l) = \frac{\sum_{t' \in T_l \cap T_d \cap sub(t)} \max\{PC'(d,t',l), PC(d,t',l+1)\}}{|T_l \cap T_d \cap sub(t)|}$$
$$PC'(d,t,l) = \max_{c \in \{c_i | c_i \in C_d \wedge L(c_i) = l\}} \{P(c,t) \min\{1, S_{random}(c)\}\}$$

Overall Measurements. QC4 has four overall measurements, based on the measures of cluster quality $QU(c)$ and topic coverage $CV(t)$. The overall measurements of clustering quality, AQ and WQ are the average of the cluster qualities, but in WQ they are weighted by cluster size. Similarly, the overall measurements of clustering coverage, AC and WC are the average of the topic coverages, but in WC they are weighted by topic size. The averages give a fairer measure of the smaller, fine grained clusters and topics; the weighted averages give a fairer measure of the larger, broad clusters and topics.

$$AQ = \text{average quality} = \frac{\sum_{c \in C} QU(c)}{|C|}$$
$$WQ = \text{weighted quality} = \frac{\sum_{c \in C} QU(c)|D_c|}{\sum_{c \in C} |D_c|}$$

To compute the overall coverage measures, AC and WC, the topic coverage is averaged over the top level topics of the ideal clustering.

$$AC = \text{average coverage} = \frac{\sum_{t \in T_1 \setminus T_\emptyset} CV(t)}{|T_1 \setminus T_\emptyset|}$$

$$WC = \text{weighted coverage} = \frac{\sum_{t \in T_1 \setminus T_\emptyset} CV(t)|D_t|}{\sum_{t \in T_1 \setminus T_\emptyset} |D_t|}$$

The measurements fairly evaluate both disjoint and overlapping topics, and topics of varying granularity without bias. Hierarchical and flat clusterings are considered fairly, because the measurements consider the individual clusters, not the hierarchical structure, and cope with overlapping clusters, including clusters that are subsets of other clusters.

4 Evaluation

This section describes how we evaluated QC4 by comparison with existing evaluation measurements. Evaluation of QC4 was completed in two ways: using synthetic test cases and using real world web clustering problems. The synthetic test cases highlight the problem scenarios and boundary cases where existing measurements fail. The real world web clustering tasks show that for simple clusterings, where existing measurements work reasonably well, QC4 reaches conclusions similar to those of existing measurements.

4.1 Synthetic Test Cases

To compare QC4 with existing measurements we devised an extensive set of synthetic test cases with different features present. A test is passed if the measure gives an appropriate distinction between the test cases with and without the feature. The tests were organised into eight groups shown in table 1, according to the feature being tested. The columns of table 1 give the different combinations of evaluation measurements that we considered as overall measurements to compare against QC4, where MI, F, AP, WP, AR, WR, AE, WE are mutual information, f-measure, average precision, weighted precision, average recall, weighted recall, average entropy, and weighted entropy respectively. The tests passed by each overall measurement are shown with a Y in the appropriate rows, for example, QC4 passes all eight tests and the 9th column shows that using just Weighted Precision (Purity) for overall evaluation fails seven of the eight tests.

QC4 handles the overlapping and hierarchical clusterings, but none of the other evaluation methods do. QC4 gives perfect scores only to ideal clusterings, but three of the other measures fail; for example, mutual information gives a better than perfect score to a clustering that contains an ideal clustering and a low quality cluster. QC4 includes separate measures for quality and coverage, but MI and F do not and the individual measures of precision, recall, and entropy do not measure both quality and coverage. QC4 handles clusterings with clusters or topics of vastly different sizes where one or more may be relatively large, but eight of the other measures fail; for example, when there is one big cluster containing all pages, the precision, entropy, and weighted recall measures give unduly good scores. QC4 handles clusterings with many small clusters or topics, but none of the other evaluation methods do; for example, all other measures give unduly good performance to a singleton clustering (one that has one cluster for each

Table 1. Synthetic test cases comparing QC4 with a wide range of overall evaluation measurements, where Y indicates passing all tests in that rows group of tests

	QC4	MI	F	AP WP AR WR	AE WE AR WR	WP WR	WE WR	AP	WP	AR	WR	AE	WE
Overlapping Clusters	Y	-	-	-	-	-	-	-	-	-	-	-	-
Hierarchical Clusterings	Y	-	-	-	-	-	-	-	-	-	-	-	-
Perfect Clustering	Y	-	Y	Y	Y	Y	Y	Y	Y	-	-	Y	Y
Separate Measures	Y	-	-	Y	Y	Y	Y	-	-	-	-	-	-
Large cluster/topic bias	Y	Y	-	Y	Y	-	-	-	-	Y	-	-	-
Small cluster/topic bias	Y	-	-	-	-	-	-	-	-	-	-	-	-
Random Clustering	Y	Y	-	-	Y	-	Y	-	-	-	-	Y	Y
Split Cluster	Y	Y	-	-	-	-	-	-	-	-	-	-	-

document) and in fact precision, recall, and entropy measures give perfect scores to the singleton clustering. QC4 gives low scores to random clusterings, but seven of the other measures fail; for example, the precision and recall measures can give unduly high scores to random clusterings, often exceeding the scores given to more sensible clusterings. QC4 gives lower scores when perfect clusters are split into smaller clusters, but eleven of the other measures fail; for example, splitting a perfect cluster has no effect of precision, recall, or entropy measures.

The results show that none of the current measurements for overall evaluation are satisfactory, while QC4 passes all tests. While existing measurements can still produce meaningful results and conclusions with simple clustering problems, these tests show that there are conditions under which existing methods can produce inaccurate results, especially with overlapping clusters or hierarchical clusterings. Conclusions drawn from the existing measurements are therefore questionable.

4.2 Real World Web Clustering Tasks

To evaluate QC4 on real world web clustering tasks we selected four queries (Jaguar, Salsa, GP, and Victoria University) and evaluated the performance of nine clustering algorithms (random clustering, and full text and snippet varieties of K-means [1], STC [4], ESTC [5], and QDC [6]) on each of the queries using twelve evaluation measurements (Mutual Information, F-measure and Average and Weighted versions of QC4 Quality, QC4 Coverage, Precision, Recall, Entropy). We averaged the values across the four queries and combined the average and weighted versions of each measurement by averaging them. For the overall evaluation in figure 1C, we also averaged the quality and coverage measures for QC4.

These clustering tasks represented simple clustering problems with little overlap or hierarchy, where existing measurements work reasonably well. Figures 1A,

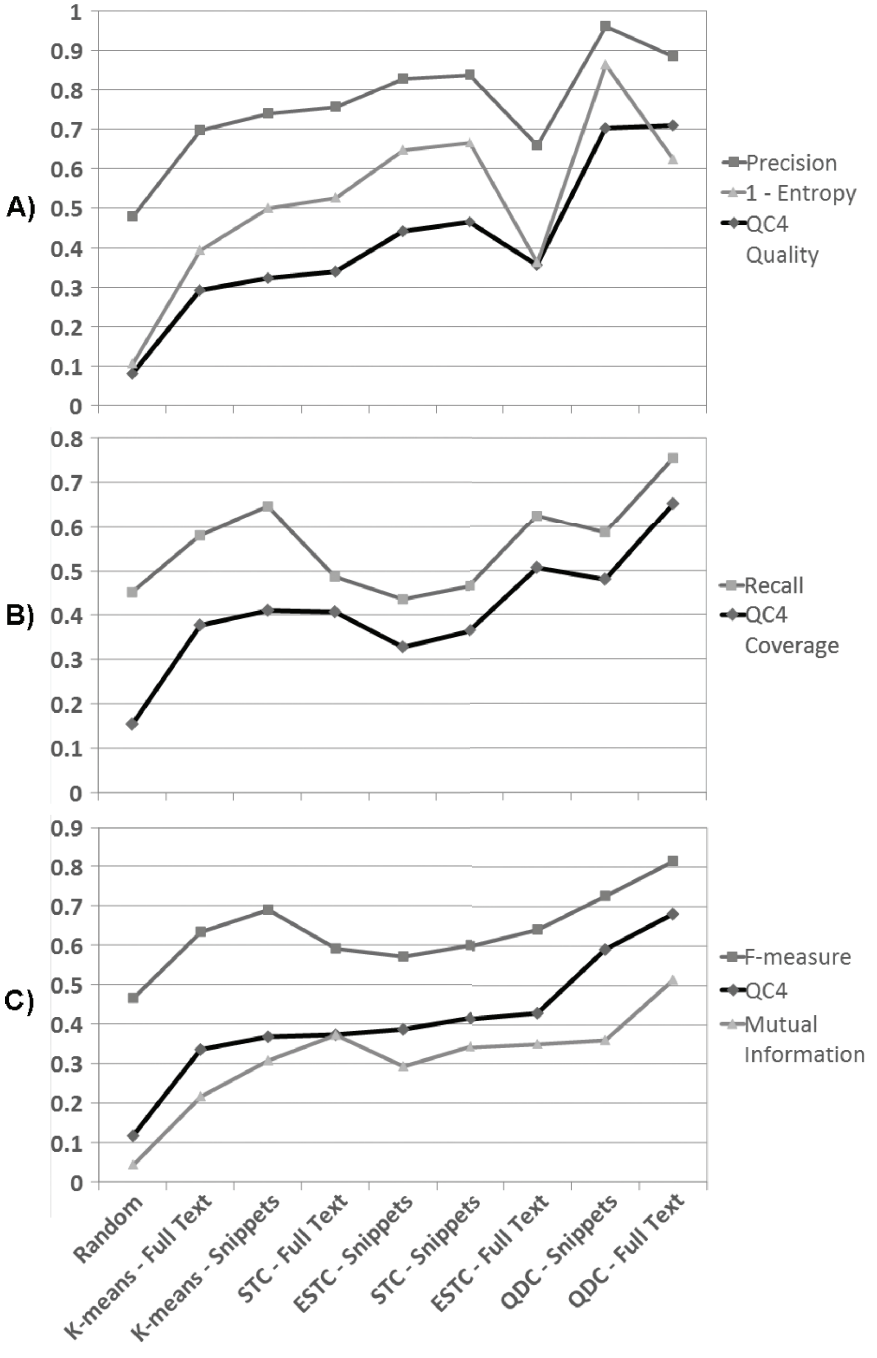


Fig. 1. Comparing measures averaged over four real world web clustering tasks. A) cluster quality measures. B) topic coverage measures. C) overall measures.

1B, and 1C show that the QC4 quality, coverages, and overall measures, respectively reach similar conclusions to those of the existing measurements.

In the few cases QC4 differs from the existing measurements, QC4 agrees with the conclusions of the relevant research literature [4,5,6], which rank the algorithms as QDC, ESTC, STC, K-means, and finally Random clustering, in order of overall web clustering performance. QC4 correctly identifies K-means as a low performing algorithm, whereas F-measure ranks its performance too highly. QC4 correctly identifies ESTC as outperforming STC, whereas mutual information incorrectly identifies STC as the higher performer. This indicates that QC4 makes sensible conclusions on real world tasks.

The real world web clustering tasks also show that QC4 is as expressive as any of the existing standard evaluation methods, and is significantly better than Precision, Recall, and F-measure due to the much lower performance given to random clusterings.

4.3 Applicability to Other Clustering Domains

QC4 has been designed and evaluated with respect to web page clustering, but it can be easily generalized to other clustering domains where clusterings feature different characteristics. The only web specific assumption in QC4 is that it is more desirable to identify small clusters than to extend the coverage of large clusters. If this assumption is not applicable in the clustering domain, the assumption can be removed by simply using the standard recall measure $R(c,t)$ instead of $R'(c,t)$ in QC4's quality measure.

5 Conclusions

This paper introduced QC4, a new clustering evaluation method that allows the fair comparison of all clustering algorithms, even those that produce clusterings with vastly different characteristics (cluster granularity: coarse or fine, clustering structure: hierarchical or flat, disjoint or overlapping, and cluster size: large or small). QC4 achieved this by generalizing the gold-standard approach to use a more general ideal clustering that can describe ideal clusterings of varying characteristics and introduced four new overall measurements that function with clusterings of different characteristics fairly in terms of cluster quality and topic coverage.

QC4 was evaluated by comparison to the standard evaluation measurements in two ways: on an extensive set of synthetic test cases and on a range of real world web clustering tasks. The synthetic test cases show that QC4 meets all the requirements of a good evaluation measurement, while all the current measurements fail with overlapping clusters, hierarchical clusterings, and some boundary cases. On simple real world web clustering tasks, where the existing methods are less affected by the conditions tested by the synthetic test cases, the results show that QC4 is at least as good as the existing evaluation measurements and gives a better evaluation in several cases.

In the future, standard test data sets can be constructed and used to evaluate standard clustering algorithms to provide a baseline for comparison. QC4 should also be evaluated on other clustering domains, especially those where clusterings have different characteristics.

Acknowledgements

Daniel Crabtree is supported by a Top Achiever Doctoral Scholarship from the Tertiary Education Commission of New Zealand and was supported during this research by an SMSCS Research Grant.

References

1. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining. (2000)
2. Ali, R., Ghani, U., Saeed, A.: Data clustering and its applications. http://members.tripod.com/asim_saeed/paper.htm (1998)
3. Wang, Y., Kitsuregawa, M.: Evaluating contents-link coupled web page clustering for web search results. In: 11th Int. Conf. on Information and Knowledge Management (CIKM 2002), McLean, VA, USA. ACM Press. (2002) 499–506
4. Zamir, O.E.: Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results. PhD thesis, University of Washington (1999)
5. Crabtree, D., Gao, X., Andreae, P.: Improving web clustering by cluster selection. In: 2005 IEEE/WIC/ACM Int. Conf. on Web Intelligence. (2005) 172–178
6. Crabtree, D., Andreae, P., Gao, X.: Query directed web page clustering. In: 2006 IEEE/WIC/ACM Int. Conf. on Web Intelligence. (2006) 202–210
7. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems **17**(2-3) (2001) 107–145
8. Strehl, A.: Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining. PhD thesis, Faculty of the Graduate School of The University of Texas at Austin (2002)
9. Tonella, P., Ricca, F., Pianta, E., Girardi, C., ITC-irst, Lucca, G.D., Fasolino, A.R., Tramontana, P., di Napoli Federico II, U., Napoli, Italy: Evaluation methods for web application clustering. In: 5th Int. Workshop on Web Site Evolution, Amsterdam, The Netherlands. (2003)
10. Meila, M.: Comparing clusterings. Technical Report 418, Department of Statistics, University of Washington (2002)
11. chiu Wong, W., Fu, A.: Incremental document clustering for web page classification. In: IEEE 2000 Int. Conf. on Info. Society in the 21st century: emerging technologies and new challenges (IS2000), Japan. (2000)
12. Crabtree, D., Gao, X., Andreae, P.: Standardized evaluation method for web clustering results. In: 2005 IEEE/WIC/ACM Int. Conf. on Web Intelligence. (2005) 280–283
13. van Rijsbergen, C.J.: Information Retrieval. Butterworths, London (1979)
14. Mackay, D.J.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press (2003)

Semantic Feature Selection for Object Discovery in High-Resolution Remote Sensing Imagery

Dihua Guo, Hui Xiong, Vijay Atluri, and Nabil Adam

MSIS Department, Rutgers University, USA
{devaguo, hui, atluri, adam}@cimic.rutgers.edu

Abstract. Given its importance, the problem of object discovery in High-Resolution Remote-Sensing (HRRS) imagery has been given a lot of attention by image retrieval researchers. Despite the vast amount of expert endeavor spent on this problem, more effort has been expected to discover and utilize hidden semantics of images for image retrieval. To this end, in this paper, we exploit a hyperclique pattern discovery method to find complex objects that consist of several co-existing individual objects that usually form a unique semantic concept. We consider the identified groups of co-existing objects as new feature sets and feed them into the learning model for better performance of image retrieval. Experiments with real-world datasets show that, with new semantic features as starting points, we can improve the performance of object discovery in terms of various external criteria.

1 Introduction

With the advances of remote sensing technology and the increases of the public interest, the remote-sensing imagery has been drawing the attention of people beyond the traditional scientific user community. Large collections of High-Resolution Remote-Sensing (HRRS) images are becoming available to the public, from satellite images to aerial photos. However, it remains a challenging task to identify objects in HRRS images. While HRRS images share some common features with traditional images, they possess some special characteristics which make the object discovery more complex and motivate our research work.

Motivating Examples. Users are interested in different types of objects on Earth as well as groups of objects with various spatial relationships. For example, consider Emergency Response Officers who are trying to find shelters to accommodate a large number of people. However, shelters are not distinguishable in Remote Sensing (RS) images. Instead, the officers could search for baseball fields, because most probably, a baseball field is connected to a school and the school could be used as a temporary shelter in emergency. In addition, qualified shelter should not be far away from water source. Therefore, the query might be "*select all the baseball fields in Newark within 1 mile from any water body*". Another interesting application domain would be urban planning. With HRRS image retrieval, we may have the task to find out "*the disinvestment area in*

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 71–83, 2007.

Hudson county industrial area". This task indicates that we need to identify the industrial areas with a lot of empty lots. While traditional Content Based Image Retrieval (CBIR) techniques discover objects such as buildings and water bodies, these two examples demonstrate that one need to discover *semantic* objects such as schools and urban areas from RS or HRRS images.

Based on the above observation, we categorize the target objects that can be recognized in RS or HRRS images into three concept levels: (1) Basic Terrain Types; (2) Individual Objects; and (3) Composite Objects. The first concept level is to distinguish the basic terrain type of the area covered by the images. There are several basic ground layouts: bare land, mountain, water, residential area, forests, etc. The second type of objects are individual objects that are recognizable in images, such as individual buildings, road segments, road intersections, cars, etc. Objects in the third concept level are composite objects. Composite objects are objects that consist of several individual objects that form a new semantics concept. For example, parks, airports, and baseball fields are all composite objects. In the motivating examples, both shelter and disinvestment area are composite objects. As one can notice, the spatial relationships among objects play a critical role in identifying composite objects and interpreting the semantics of HRRS images.

Despite the vast amount of expert effort, it is well known that the performance of CBIR is limited by the gap between low-level features and high-level semantic concepts. Recently, researchers proposed several statistical models [6,1,12,2,3,9] for analyzing the statistical relations between visual features and keywords. These methods can discover some hidden semantics of images. However, these methods annotate scenery images according to the individual objects' presence in each image. Spatial relations among objects are not taken into consideration. Those spatial relationships are critical and cannot be ignored in HRRS images. Hence, in HRRS images, users pay more attention on composite objects than on individual objects. This suggests that we have to examine the spatial relationships among objects when we try to identify objects in HRRS images.

In this paper, we investigate the problem of automatically annotating images using relevance-based statistical model on HRRS images. Specifically, we exploit a hyperclique pattern discovery method [13] to create new semantic features and feed them into the relevance-based statistical learning model. Hyperclique patterns have the ability to capture a strong connection between the overall similarity of a set of objects and can be naturally extended to identify co-existing objects in HRRS images. Traditionally, by using a training set of annotated images, the relevance-model can learn the joint distribution of the blobs and words. Here, the blobs are image segments acquired directly from image segmentation procedure. Our approach extends the meaning of blobs by identifying the co-existing objects/segments as new blobs. The proposed approach has been tested using the USGIS high-resolution orthology aerial images. Our experimental results show that, with new semantic features as starting points, the performance of learning model can be improved according to several external criteria.

2 Domain Challenges

In this section, we describe some domain challenges for object discovery in HRRS images as follows.

- First, it is nontrivial to perform feature selection for image retrieval in HRRS images. In [12], researchers developed a mechanism to automatically assign different weights to different features according to the relevance of a feature to clusters in the Corel images. However, unlike Corel Image, HRRS images are severely affected by the noise such as shadow and the surface materials of HRRS images are limited. This makes the primitive features, such as color, texture and shape, not good enough for identifying objects in HRRS images. As a result, in addition to the primitive features, the derivative features, such as geometric features and semantic features, are required for better object discovery in HRRS images. In this research, we add semantic features that capture the spatial relationships among objects to image annotation model.
- Also, HRRS images usually lack salient regions and carry a lot of noise [4]. This data problem has been largely ignored by existing approaches, thus not suitable for object discovery in HRRS images. Indeed, existing methods often use segmentation techniques which may not work well in noisy environments. Moreover, the grid technology [3], a substitute of segmentation, often assume that each grid only contains one salient object. To satisfy the assumption, we have to cut the image into very small grids. However, according to our observation, both traditional segmentation algorithms and grid technology will generate 40-120 segments/grids for a 512×512 1-foot resolution aerial image, which makes the performance of annotation model deteriorate dramatically compared to 10-20 segments/grids per image. Therefore, we propose a two-stage segmentation algorithm to accommodate the uniqueness of HRRS images.
- Finally, another challenge faced by the HRRS image annotation is the importance of measuring spatial relationships among objects. In the HRRS images, individual objects cannot determine the semantics of the entire scene by itself. Rather, the repeated occurrence of certain object in the scene or the co-occurrence of objects reflect high-level semantic concepts. For instance, if there is an remote sensing image about a city or urban area, instead of roof of individual house, people maybe more interested in identifying a park, which is the composition of grass land, pond, and curvy road. People would not be interested in large building roof alone. Nevertheless, if we identify that large building roofs have large parking lot and major road nearby, this would also be interesting, as we can annotate the image as shopping mall.

3 Object Discovery with Semantic Feature Selection

In this section, we introduce a method for **O**bject dis**C**overy with semanti**C** feat**U**re s**E**lection (OCCUE). Figure 1 shows an overview of the OCCUE method. A detailed discussion of each step of OCCUE is given in the following subsections.

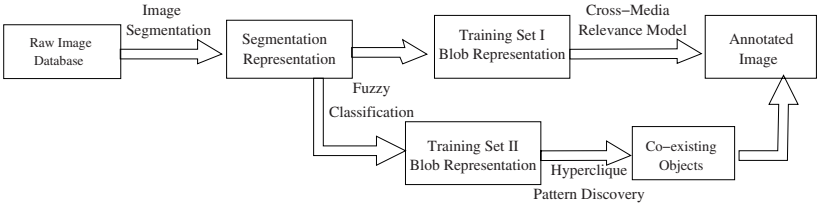


Fig. 1. A Overview of the OCCUE Method

3.1 Image Segmentation

Image segmentation divides an image into separated regions. In a large-scale HRRS image database, the images naturally belong to different semantic clusters. For example, most of HRRS images can be categorized into four main semantic clusters at the land cover level including grass, water, residence and agriculture [10]. These land-cover level semantic clusters can also be divided into semantic subclusters at an object level. For these subclusters, the distinguishing primitive features are different. Moreover, the objects in each land-cover cluster are very different. For example, the objects in urban areas are usually road segments, single house roofs, or small vegetated areas. In contrast, woods and grass are dominant in suburban areas. Likewise, different composite objects also appear in different land-cover clusters. For instance, a park is always a large contiguous vegetated area. This different scale distinguishes parks from gardens. In OCCUE, we exploit a two-step approach to increase segmentation reliability. Our two-step segmentation approach satisfies the uniqueness of RS images by segmenting images at the land-cover level first and then dividing images further into individual objects or components of an individual object.

Another major advantage of using two-step image segmentation approach is that this segmentation approach can reflect the hierarchies that exist in the structure of the real-world objects which we are detecting. By abstracting houses, buildings, roads and other objects, people can identify residential areas and the aggregation of several residential areas yields a town. This hierarchy is obviously determined by scale.

In OCCUE, we apply the texture-based algorithms proposed by [4] to segment image at the land cover level. This segmenting method consists of three major steps: (i) hierarchical splitting that recursively splits the original image into children blocks by comparing texture features of blocks, (ii) optimizing, which adjusts the splitting result, if the results of the reduced resolution images have dramatically reduced segments, (iii) merging, in which the adjacent regions with similar texture are merged until a stopping criterion is satisfied.

After the land-cover level segmentation, images are segmented into small regions using eCognition along with different input parameters according to land-cover type [5]. Each segment is represented by the traditional features, e.g. colors, textures and shapes, as well as the geometric features. eCognition utilizes

a bottom up-region-merging technique starting with one-pixel. In subsequent steps, smaller image segments are merged into bigger ones [5]. We believe that this is one of the easy-to-use and reliable segmentation tools for HRRS images, given the characteristics of the HRRS images: 1)with salt and pepper noises; 2) affected by the atmosphere and the reflective conditions.

The following extracted features represent major visual properties of each image segment.

- **Layer Values** are features concerning the pixel channel values of an image segment, mainly the spectral features, including mean, brightness, max difference, standard deviation, the ratio of layer mean value of an image segment over the all image, minimum pixel value, maximum pixel value, the mean difference to neighboring segment, the mean difference to brighter neighboring segment, mean difference to darker neighboring object.
- **Shape Features** include area (measured by pixel), length/width ratio which is the eigenvalues of the covariance matrix with the larger eigenvalue being the numerator of the factor, length, width, border length, density expressed by the area covered by the image segment divided by its radius, main direction, asymmetry, compactness (the product of the length m and the width n of the corresponding segment and divided by the number of its inner pixels), elliptic fit and rectangular fit.
- **Texture Features** evaluate the texture of an image segment based on the gray level co-occurrence matrix (GLCM) and the grey level difference vector (GLDV) of the segments pixel [5]. The grey level co-occurrence matrix (GLCM) is a tabulation of how often different combinations of pixel grey level occur in an image. A different co-occurrence matrix exists for each spatial relationship. Therefore, we have to consider all four directions (0 45, 90, 135) are summed before texture calculation. An angle of 0 represents the vertical direction, an angle of 90 the horizontal direction. Every GLCM is normalized, which guarantee the GLCM is symmetrical. The more distant to the diagonal, the greater the difference between the pixels grey level is. The GLCM matrices can be further broken down to measure the homogeneity, contrast, dissimilarity (contrast increases linearly), entropy (distributed evenly), mean, standard deviation, and correlation. GLDV is the sum of diagonals of GLCM. It counts the occurrence of references to the neighbor pixels. Similarly to GLCM matrices, GLDV can measure the angular second moment (high if some elements are large), entropy (high if all similar), mean, and contrast.
- **Position Features** refer to the positions of segments within an image.

3.2 Fuzzy Classification

After we segment the images into relatively homogeneous regions, the next step is to group similar image segments into a reasonable number of classes, referred as blob tokens in [12]. Segments in each class are similar even though they are not spatially connected. In the literature [12], unsupervised classification algorithms

is employed using the primitive features or weighted features. Using the weighted features would successfully reduce the dimensionality compared with using all primitive features as clustering algorithm input. However, we used supervised classification method that is efficient in grouping image segments into semantic meaningful blobs.

Specifically, fuzzy logic based supervised classification is applied to generate blobs. Starting with an empty class hierarchy, we manually insert sample classes and using the features description as definition of a certain class. While nearest neighbor and membership functions are used to translate feature values of arbitrary range into a value between 0 (no membership) and 1 (full membership), logical operators summarize these return values under an overall class evaluation value between 0 and 1. The advantages of fuzzy classification are [5]

- Translating feature values into fuzzy values standardizes features and allows to combine features, even of very different ranges and dimensions.
- It enables the formulation of complex feature descriptions by means of logical operations and hierarchical class descriptions.

Finally, fuzzy classification also helps to merge the neighboring segments that belong to the same class and get a new semantic meaningful image blob which truly represents the feature and not just a part of it.

3.3 Hyperclique Patterns

In this paper, hyperclique patterns [13,14] are what we used for capturing co-existence of spatial objects. The concept of hyperclique patterns is based on frequent itemsets. In this subsection, we first briefly review the concepts on frequent itemsets, then describe the concept of hyperclique patterns.

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items. Each transaction T in database D is a subset of I. We call $X \subseteq I$ an itemset. The support of X $supp(X)$ is the fraction of transactions containing X. If $supp(X)$ is no less than a user-specified minimum support, X is called a frequent itemset. The confidence of association rule $X_1 \rightarrow X_2$ is defined as $conf(X_1 \rightarrow X_2) = supp(X_1 \cup X_2)/supp(X_1)$. It estimates the likelihood that the presence of a subset $X_1 \subseteq X$ implies the presence of the other items $X_2 = X - X_1$.

If the minimum support threshold is low, we may extract too many spurious patterns involving items with substantially different support levels, such as (caviar, milk). If the minimum support threshold is high, we may miss many interesting patterns occurring at low levels of support, such as (caviar, vodka). To measure the overall affinity among items within an itemset, the h-confidence was proposed in [13]. Formally, the h-confidence of an itemset $P = \{i_1, i_2, ...i_m\}$ is defined as $hconf(P) = min_k\{conf(i_k \rightarrow P - i_k)\}$. Given a set of items I and a minimum h-confidence threshold h_c, an itemset $P \subseteq I$ is a hyperclique pattern if and only if $hconf(P) \geq h_c$. A hyperclique pattern P can be interpreted as that the presence of any item $i \in P$ in a transaction implies the presence of all other items $P - \{i\}$ in the same transaction with probability at least h_c.

This suggests that h-confidence is useful for capturing patterns containing items which are strongly related with each other. A hyperclique pattern is a maximal hyperclique pattern if no superset of this pattern is a hyperclique pattern.

3.4 Converting Spatial Relationship into Feature Representation

Approaches for modelling spatial relationships can be grouped into three categories: graph-based approaches, rule based approaches, and mathematical logic using 2D strings as the projections of the spatial relationships. However, none of this can be used as input for statistical Cross Relevance Model (CRM). In addition, we concentrate on the presence of the objects in the image rather than the complex geometric or topological spatial relationships. For example, consider a golf course, we are interested in the appearance of the well textured grassland, sand, non-rectangle water-body in a relatively small region. Whether the sand is left or right to the water-body is not important. In OCCUE, we apply hyperclique pattern discovery algorithm [13] to detect co-existing objects.

Table 1. A sample image-blob data set

Image	Blobs
in1	3,7,11,12,19,22,23,24,25
in2	3,7,6,12,13,15,18,20,23,24
in3	3,7,6,11,16,18,20,24,26
in5	7,6,10,11,12,20
in6	3,7,6,19,20,23,24,25
in7	3,7,12,19,20,23
in8	3,6,7,10,11,12,19,20,23
in9	3,6,15,11,12,20,24,26
in10	6,7,11,12,23,24
in11	3,6,7,11,12,19,22,23,24
in12	3,7,12,19,20,23,24

Example 2.1. *After segmentation, images are represented by the blob ID as shown in Table 1, let us consider a pattern* $X=\{b_3, b_7, b_{24}\}$*, which implies that blob (#3 roof type II, #7 shade type II , #24 grass type IV) usually appears together. We have* $supp(b_3) = 82\%$*,* $supp(b_7) = 91\%$*,* $supp(b_{24}) = 73\%$*, and* $supp(b_3, b_7, b_{24}) = 55\%$*. Then,* $conf(b_3 \rightarrow b_7, b_{24}) = supp(b_3, b_7, b_{24})/supp(b_3) = 67\%$*;* $conf(b_7 \rightarrow b_3, b_{24}) = supp(b_3, b_7, b_{24})/supp(b_7) = 60\%$ *;* $conf(b_{24} \rightarrow b_3, b_7) = supp(b_3, b_7, b_{24})/supp(b_{24}) = 75\%$*. Therefore,* $hconf(X) = min(conf(b_3 \rightarrow b_7, b_{24}), conf(b_7 \rightarrow b_3, b_{24}), conf(b_{24} \rightarrow b_3, b_7)) = 60\%$*. According to the definition of* ***hyperclique pattern****, pattern* $\{b_3, b_7, b_{24}\}$ *is a hyperclique pattern at the threshold 0.6. Therefore, we treat the set of these three blobs as a new semantic feature. We treated these newly discovered hyperclique pattern as new blobs in additional to the existing blobs. Meanwhile, the original blobs #3, #7, and #24 are deleted from the original table. Table 1 will be converted to Table 2. The new blobs are represented using 3 digits number in order to distinguish from the original blobs. We convert the spatial relationship into a measurable representation, so that we can apply statistical model in the next step.*

Table 2. A sample image represented in new blob

Image	Blobs
in1	11,12,19,22,23,25,**105**
in2	6,12,13,15,18,20,23,**105**
in3	11,16,18,20,26,**105**
in5	7,6,10,11,12,20
in6	6,19,20,23,25,**105**
in7	3,7,12,19,20,23
in8	3,6,7,10,11,12,19,20,23
in9	3,6,15,11,12,20,24,26
in10	6,7,11,12,23,24
in11	6,11,12,19,22,23 **105**
in12	12,19,20,23,**105**

3.5 A Model of Image Annotation

Suppose we are given an un-annotated image in image collection $\mathcal{I} \in \mathcal{C}$. We have the object representation of that image $\mathcal{I} = \{o_1 \dots o_m\}$, and want to automatically select a set of words $\{w_1 \dots w_n\}$ that reflect the content of the image.

The general approach is widely accepted by statistical modelling approach. Assume that for each image $\mathcal{I}$ there exists some underlying probability distribution P(·|I). We refer to this distribution as the relevance model of I [8,7]. The relevance model can be thought of as an urn that contains all possible objects that could appear in image $\mathcal{I}$ as well as all words that could appear in the annotation of $\mathcal{I}$. We assume that the observed image representation $\{o_1 \dots o_m\}$ is the result of m random samples from P(·|I).

In order to annotate an image with the top relevance words, we need to know the probability of observing any given word w when sampling from P(·|I). Therefore, we need to estimate the probability P(w|I) for every word w in the vocabulary. Given that P(·|I) itself is unknown, the probability of drawing the word w can be approximated by training set $\mathcal{T}$ of annotated images.

$$P(w|I) \approx P(w|o_1 \dots o_m) \tag{1}$$

$$P(w, o_1, \dots, o_m) = \sum_{J \in T} P(J) P(w, o_1, \dots, o_m | J) \tag{2}$$

Assuming that observing w and blobs are mutually independent for any given image, and identically distributed according to the underlying distribution P(·|J). This assumption guarantees we can rewrite equation (2) as follows:

$$P(w, o_1, \dots, o_m) = \sum_{J \in T} P(J) P(w|J) \prod_{i=1}^{m} P(o_i|J) \tag{3}$$

We assume the prior probability $P(J)$ follows uniform over all images in training set $mathcalT$. We follow [6] and use smoothed maximum likelihood estimates for the probabilities in equation (3). The estimations of the probabilities of blob and word given image J are obtained by:

$$P(w|J) = (1 - \alpha_J) \frac{Num(w, J)}{|J|} + \alpha_J \frac{Num(w, T)}{|T|} \tag{4}$$

$$P(o|J) = (1 - \beta_J)\frac{Num(o, J)}{|J|} + \alpha_J \frac{Num(o, T)}{|T|} \quad (5)$$

Here, $Num(w, J)$ and $Num(o, J)$ represents the actual number of times the word w or blob o occurs in the annotation of image J. $Num(w, T)$ and $Num(o, T)$ is the total number of times w or o occurs in all annotation in the training set T.$|J|$ denotes for the aggregate count of all words and blobs appearing in image J, and $|T|$ denotes the total size of the training set. The smoothing parameter α_J and β_J determine the interpolation degree between the maximum likelihood estimates and the background probabilities. Due to the different occurrence patterns between words (Zipfian distribution) and blobs (uniform distribution) in images, we separate the two smoothing parameter as α_J and β_J.

Finally, Equation (1) - (5) provide the mechanism for approximating the probability distribution $P(w|I)$ for an underlying image I. We annotate images by first estimating the probability distribution $P(w|I)$ and then select the highest ranking n words for the image.

4 Experimental Evaluation

In this section, we present experiments on real-world data sets to evaluate the performance of object discovery with semantic feature selection. Specifically, we show: (1) an example set of identified semantic spatial features, (2) a performance comparison between the OCCUE model and a state-of-the-art Cross-media Relevance Model (CRM) model [6].

4.1 The Experimental Setup

Experimental Data Sets. Since our focus in this paper is on HRRS images rather than regular scenery images, we will not adopt the popular image dataset Corel, which is considered as a benchmark for evaluating the performance of image retrieval algorithms. Instead, we use the high resolution orthoimagery of the major metropolitan areas. This data set is distributed by United States Geological Survey (USGS - http://www.usgs.gov/). The imagery is available as Universal Transverse Mercator (UTM) projection and referenced to North American Datum of 1983. For example, the New Jersey orthoimagery is available as New Jersey State Plane NAD83. The file format is Georeferenced Tagged Image File Format(GeoTIFF).

Data Preprocessing. We downloaded the images of 1-foot resolution in the New York metro area and Springfield MA. Each raw image is about 80MB, which is then be processed using the Remote Sensing Exploitation Platform (ENVI - http://www.ittvis.com/envi/). Images with blurred scene or with no major interesting objects, such as square miles of woods, are discarded. For images that contain objects we are interested in, we grid the image into small pieces (2048 $\times$ 2048 pixels). Finally, we have 800 images in our experimental data set and there are 32 features: 10 color features, 10 shape features and 12 texture features.

Keywords. The keywords used to annotate the semantics of the HRRS images are also different from the traditional scenery images. First of all, they are not attainable directly from the data seta as those of Corel images. Rather, it is manually assigned by domain experts. These keywords can be divided into three groups: keywords regard landcover, individual objects, and composite objects.

Validation. In our experiments, we divided the data set into 10 subsets with equal number of images. We performed 10-cross validation. For each experiment, 8 randomly selected sub-dataset are used as training set, a validation set of 80 images and a test set of 80 images. The validation set is used to select the model parameters. Every images in the data set is segmented into comparatively uniform regions. The number of segments in each image, and the size of each segment (measured by the number of pixels) are empirically selected using the training and validating sets.

Blobs. A fuzzy classification algorithm is applied to generate image blobs. In our experiment, we generated 30 image blobs. Table 3 shows some examples of image blobs. Also, Figure 2 shows a sample image and its blob representation.

Table 3. Examples of Blobs

ID	Description	size	color	shape	texture
1	house I	(0,1200)	(150,180)	rectangle	smooth
2	house II	(1200, 3000)	(150, 180)	rectangle	smooth
3	house III	(0, 1200)	(180, 255)	rectangle	smooth
4	grass I	(0, 2000)	(140, 160)	irregular	smooth
5	grass II	(0, 2000)	(140, 180)	irregular	rough
30	sand	(0, 5000)	(190,210)	round	rough

Blob ID
1, 2
3, 4
11, 12
28

Fig. 2. An Image and Its Blob Representation

Spatial Semantic Features. All images with identified image blobs are used to identify the co-occurrence of image blobs. Specifically, we exploited a hyperclique pattern discovery method to find complex objects that consist of co-existing image blobs, which usually form a unique high-level semantic concept and are treated as spatial semantic features. For instance, Table 4 shows some example semantic features.

4.2 Results of Composite Object Annotation

To evaluate the annotation performance, we apply some external metrics including Precision, Recall, and F-measure. Specifically, we judge the relevance of the

Table 4. An Example of Semantic Features

blobID	Comp-Object
17, 8	Golf Course
3, 20	Industrial Building
3, 4, 24	Industrial Building
1, 2, 5	Residential Building
1, 2, 9, 10	Residential Building
2, 12, 22	Baseball Field

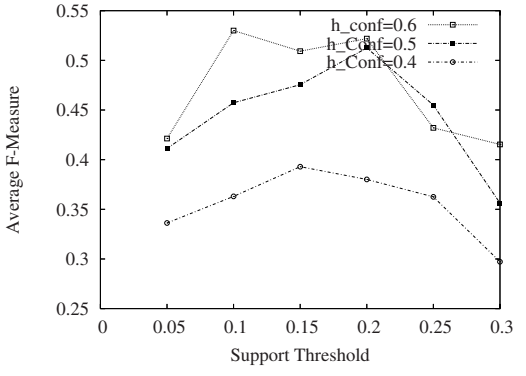


Fig. 3. F-measure Values at Different Parameters

retrieved images by looking at the manual annotations of the images. A *Recall* measure is defined as the number of the correctly retrieved images divided by the number of relevant images in the test data set. The *Precision* measure is defined as the number of correctly retrieved images divided by the number of retrieved images. In order to make a balance between the recall and precision measures, we also compute the F-measure which is defined as $\frac{2*Recall*Precision}{Recall+Precsion}$.

Parameter Selection. The hyperclique pattern discovery algorithm has two parameters: support and h-confidence. We examine the impact of these two parameters on the performance of object annotation. The minimum support and the h-confidence thresholds would affect object discovery. For example, the set of blobs (1, 2, 5, 9, 10) can be identified as co-existing objects with minimum support 0.05 and h-confidence 0.4, while it could not be identified when we change the minimum support to 0.15. Figure 3 shows the F-measure values with the change of minimum support and h-confidence thresholds. As can be seen, the F-measure values vary at different support and h-confidence thresholds. However, we can observe a general trend is that the F-measure values increase with the increase of H-confidence. Also, the maximum F-measure value is achieved when the support threshold is relatively high. This is reasonable, since a relatively high support threshold can guarantee statistical significance and provide a better coverage of objects. For this reason, in our experiments, we set relatively high support and h-confidence thresholds.

Table 5. A Performance Comparison

measures	word class	Avg. Prec.	Avg. Recall	F-Measure
CRM	land use	0.6801	0.5923	0.6332
OCCUE	land use	0.7512	0.7229	0.7368
CRM	object level	0.3013	0.1827	0.2274
OCCUE	object level	0.4682	0.3677	0.4119

A Model Comparison. We compared the annotation performance of the two models, the CRM model and the OCCUE model. We annotate each test image with 1 word from the land-cover level, 3 words from the composite object level. Table 5 shows the comparison results. In the table, we can observe that, for both land-cover level and composite-object level, the performance of OCCUE is much better than that of CRM in terms of Precision, Recall, and F-measure. For instance, for the composite-object level, the F-measure value is improved from 0.2274 (CRM) to 0.4119 (OCCUE). This improvement is quite significant.

5 Conclusions and Future Work

In this paper, we proposed a semantic feature selection method for improving the performance of object discovery in High-Resolution Remote-Sensing (HRRS) images. Specifically, we exploited a hyperclique pattern discovery technique to capture groups of co-existing individual objects, which usually form high-level semantic concepts. We treated these groups of co-existing objects as new semantic features and feed them into the learning model. As demonstrated by our experimental results, with new semantic feature sets, the learning performance can be significantly improved.

There are several potential directions for future research. First, we propose to adapt Spatial Auto-Regression (SAR) model [11] for object discovery in HRRS images. The SAR model has the ability in measuring spatial dependency, and thus is expected to have a better prediction accuracy for spatial data. Second, we plan to organize the identified semantic features as a concept hierarchy for the better understanding of new discovered high-level objects.

References

1. K. Barnard, P. Duygulu N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *Machine learning research*, 3(1):1107–1135, 2003.
2. P. Duygulu, K. Barnard, N. de Freitas, and D. Dorsyth. Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In *ECCV*, volume 4, pages 97–112, 2002.
3. SL Feng, R Manmatha, and V Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, pages 1002–1009, 2004.
4. D. Guo, V. Atluri, and N. Adam. Texture-based remote-sensing image segmentation. In *ICME*, pages 1472–1475, 2005.
5. http://www.definiens imaging.com/. ecognition userguide, 2004.

6. J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *SIGIR*, pages 254–261, 2003.
7. V. Lavrenko, M. Choquette, and W. Croft. Cross-lingual relevance models. In *SIGIR*, pages 175–182, 2002.
8. V. Lavrenko and W. Croft. Relevance-based language models. In *SIGIR*, pages 120–127, 2001.
9. Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *MISRM*, 1999.
10. G. Sheikholeslami, W. Chang, and A. Zhang. Semquery: Semantic clustering and querying on heterogeneous features for visual data. *TKDE*, 14(5):988–1002, 2002.
11. S. Shekhar and S. Chawla. *Spatial Databases: A Tour*. Prentice Hall, 2003.
12. L. Wang, L. Khan, L. Liu, , and W. Wu. Automatic image annotation and retrieval using weighted feature selection. In *IEEE-MSE*. Kulwer Publisher, 2004.
13. H. Xiong, P. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support distribution. In *ICDM*, pages 387–394, 2003.
14. H. Xiong, P. Tan, and V. Kumar. Hyperclique pattern discovery. *Data Mining and Knowledge Discovery Journal*, 13(2):219–242, 2006.

Deriving Private Information from Arbitrarily Projected Data

Songtao Guo and Xintao Wu

University of North Carolina at Charlotte
{sguo,xwu}@uncc.edu

Abstract. Distance-preserving projection based perturbation has gained much attention in privacy-preserving data mining in recent years since it mitigates the privacy/accuracy tradeoff by achieving perfect data mining accuracy. One apriori knowledge PCA based attack was recently investigated to show the vulnerabilities of this distance-preserving projected based perturbation approach when a sample dataset is available to attackers. As a result, non-distance-preserving projection was suggested to be applied since it is resilient to the PCA attack with the sacrifice of data mining accuracy to some extent. In this paper we investigate how to recover the original data from arbitrarily projected data and propose AK-ICA, an Independent Component Analysis based reconstruction method. Theoretical analysis and experimental results show that both distance-preserving and non-distance-preserving projection approaches are vulnerable to this attack. Our results offer insight into the vulnerabilities of projection based approach and suggest a careful scrutiny when it is applied in privacy-preserving data mining.

1 Introduction

Privacy is becoming an increasingly important issue in many data mining applications. A considerable amount of work on privacy preserving data mining has been investigated recently [2,1,9,7,12]. Among them, randomization has been a primary tool to hide sensitive private data for privacy preserving data mining. Random perturbation techniques aim to distort sensitive individual values while preserving some particular properties and hence allowing estimation of the underlying distribution.

Consider a data set X with n records of d attributes. The randomization based approaches generate a perturbed data set Y following some predefined procedure, e.g., additive noise based approach applies $Y = X + E$ where E is an additive noise data set, projection based approach applies $Y = RX$ to map the sensitive data into a new space where R is a transformation matrix. Usually the perturbed data Y is expected to be dissimilar to the original X while some aggregate properties (e.g., mean and covariance matrices for numerical data) of X are preserved or can be reconstructed after perturbation. The additive noise based approach has been challenged in privacy preserving data mining community and several individual value reconstruction methods have been investigated [9,7,5,6].

In this paper, our focus will be the projection based approach. A special projection based approach called *rotation projection* has recently been investigated in [4,11]. Since the transformation matrix R is required to be orthonormal (i.e., $RR^T = R^T R = I$),

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 84–95, 2007.

geometric properties (vector length, inner products and distance between a pair of vectors) are strictly preserved. Hence, data mining results on the rotated data can achieve perfect accuracy. One apriori knowledge PCA based attack was recently investigated to show the vulnerabilities of this distance preserving projected based perturbation approach when a sample dataset is available to attackers [10]. As a result, non-distance preserving projection was suggested be applied since it is resilient to the apriori knowledge PCA based attack with the sacrifice of data mining accuracy to some extent.

We investigate whether attackers can recover the original data from arbitrarily projected data (R can be any transformation matrix, hence the distance might not be preserved in the transformed space). Specifically, we propose an Apriori-Knowledge ICA based reconstruction method (AK-ICA), which may be exploited by attackers when a small subset of sample data is available to attackers. Our theoretical analysis and empirical evaluation shall show AK-ICA can effectively recover the original data with high precision when a part of sample data is a-priori known by attackers. Since the proposed technique is robust with any transformation matrix even with a small subset of sample data available, it poses a serious concern for projection based privacy preserving data mining methods.

The rest of this paper is organized as follows. In Section 2 we review the projection based perturbation approach and show current attempts to explore the vulnerability of this model. In Section 3 we briefly revisit ICA technique, which will be used when we introduce our AK-ICA attack in Section 4. We also show why AK-ICA can breach privacy from arbitrary transformation with the help of a small part of sample data. Section 5 presents our experimental results. We offer our concluding remarks and point out future work in Section 6.

2 The Projection Based Perturbation

The projection based perturbation model can be described by

$$Y = RX \tag{1}$$

Where $X \in \mathcal{R}^{p\times n}$ is the original data set consisting of n data records and p attributes. $Y \in \mathcal{R}^{q\times n}$ is the transformed data set consisting of n data records and q attributes. R is a $q \times p$ transformation matrix. In this paper, we shall assume $q = p = d$ for convenience.

In [4], the authors defined a rotation based perturbation method, i.e., $Y = RX$, where R is a $d \times d$ orthogonormal matrix satisfying $R^T R = RR^T = I$. The key features of rotation transformation are preserving vector length, Euclidean distance and inner product between any pair of points. Intuitively, rotation preserves the geometric shapes such as hyperplane and hyper curved surface in the multidimensional space. It was proved in [4] that three popular classifiers (kernel method, SVM, and hyperplane-based classifiers) are invariant to the rotation based perturbation.

Similarly, the authors in [11] proposed a random projection-based multiplicative perturbation scheme and applied it for privacy preserving distributed data mining. The random matrix $R_{k\times m}$is generated such that each entry $r_{i,j}$ of R is independent and

identically chosen from some normal distribution with mean zero and variance σ_r^2. Thus, the following properties of the rotation matrix are achieved.

$$E[R^T R] = k\sigma_r^2 I \qquad E[RR^T] = m\sigma_r^2 I$$

If two data sets X_1 and X_2 are perturbed as $Y_1 = \frac{1}{\sqrt{k}\sigma_r} RX_1$ and $Y_2 = \frac{1}{\sqrt{k}\sigma_r} RX_2$ respectively, then the inner product of the original data sets will be preserved from the statistical point of view:

$$E[Y_1^T Y_2] = X_1^T X_2$$

Previously, the authors in [13] defined a rotation-based data perturbation function that distorts the attribute values of a given data matrix to preserve privacy of individuals. Their perturbation scheme can be expressed as $Y = RX$ where R is a $d \times d$ matrix with each row or column having only two non-zero elements, which represent the elements in the corresponding R_p.

$$R = \begin{pmatrix} cos\theta_1 & 0 & \sin\theta_1 & 0 & ... \\ ... & ... & ... & ... & ... \\ 0 & -\sin\theta_2 & 0 & \cos\theta_2 & ... \\ ... & ... & ... & ... & ... \\ -\sin\theta_1 & 0 & \cos\theta_1 & 0 & ... \\ ... & ... & ... & ... & ... \\ 0 & \cos\theta_2 & 0 & \sin\theta_2 & ... \end{pmatrix}$$

It is easy to see that the perturbation matrix R here is an orthogonormal matrix when there are even number of attributes. If we have odd number of attributes, according to their scheme, the remaining one is distorted along with any previous distorted attribute, as long as some condition is satisfied.

In summary, all the projection matrices applied above tend to be orthogonal such that distance is mostly preserved. Hence they can be considered as special cases of the general projection model investigated here.

Attacking Methods

For the case where $R^T R = RR^T = I$, it seems that privacy is well preserved after rotation, however, a small known sample may be exploited by attackers to breach privacy completely. We assume that a small data sample from the same population of X is available to attackers, denoting as $\tilde{X}$. When $X \cap \tilde{X} = X^\ddagger \neq \emptyset$, since many geometric properties (e.g. vector length, distance and inner product) are preserved, attackers can easily locate $X^\ddagger$'s corresponding part, $Y^\ddagger$, in the perturbed data set by comparing those values. From $Y = RX$, we know the same linear transformation is kept between $X^\ddagger$ and $Y^\ddagger$: $Y^\ddagger = RX^\ddagger$. Once the size of $X^\ddagger$ is at least $rank(X) + 1$, the transformation matrix R can easily be derived through linear regression.

For the case where $X^\ddagger = \emptyset$ or too small, the authors in [10] proposed a PCA attack. The idea is briefly given as follows. Since the known sample and private data share the same distribution, eigenspaces (eigenvalues) of their covariance matrices are expected to be close to each other. As we know, the transformation here is a geometric rotation

which does not change the shape of distributions (i.e., the eigenvalues derived from the sample data are close to those derived from the transformed data). Hence, the rotation angles between the eigenspace derived from known samples and those derived from the transformed data can be easily identified. In other words, the rotation matrix R is recovered.

We notice that all the above attacks are just for the case in which the transformation matrix is orthonormal. In our general setting, the transformation matrix R can be any matrix (e.g. shrink, stretch, dimension reduction) rather than the simple orthonormal rotation matrix. When we try to apply the PCA attack on non-isometric projection scenario, the eigenvalues derived from the sample data are not the same as those derived from the transformed data. Hence, we cannot derive the transformation matrix R from spectral analysis. As a result, the previous PCA based attack will not work any more (See our empirical evaluation in Section 5.2).

Intuitively, one might think that the Independent Component Analysis (ICA) could be applied to breach the privacy. It was argued in [4,11] that ICA is in general not effective in breaking *privacy* in practice due to two basic difficulties in applying the ICA attack directly to the projection based perturbation. First, there are usually significant correlations among attributes of X. Second, more than one attribute may have Gaussian distributions. We would emphasize that these two difficulties are generally held in practice. Although we can not apply ICA directly to estimate X from the perturbed data $Y = RX$, we will show that there exists a possible attacking method AK-ICA in the following sections.

3 ICA Revisited

ICA is a statistical technique which aims to represent a set of random variables as linear combinations of statistically independent component variables.

Definition 1 *(ICA model) [8]*
ICA of a random vector $\mathbf{x} = (x_1, \cdots, x_m)^T$ *consists of estimating of the following generative model for the data:*

$$\mathbf{x} = A\mathbf{s} \quad or \quad X = AS$$

where the latent variables (components) s_i *in the vector* $\mathbf{s} = (s_1, \cdots, s_n)^T$ *are assumed independent. The matrix* A *is a constant* $m \times n$ *mixing matrix.*

The basic problem of ICA is to estimate both the mixing matrix A and the realizations of the independent components s_i using only observations of the mixtures x_j. Following three restrictions guarantee identifiability in the ICA model.

1. All the independent components s_i, with the possible exception of one component, must be non-Gaussian.
2. The number of observed linear mixtures m must be at least as large as the number of independent components n.
3. The matrix A must be of full column rank.

The second restriction, $m \geq n$, is not completely necessary. Even in the case where $m < n$, the mixing matrix A is identifiable whereas the realizations of the independent components are not identifiable, because of the noninvertibility of A. In this paper, we make the conventional assumption that the dimension of the observed data equals the the number of independent components, i.e., $n = m = d$. Please note that if $m > n$, the dimension of the observed vector can always be reduced so that $m = n$ by existing methods such as PCA.

The couple (A, S) is called a representation of X. Since $X{=}AS{=}(A\Lambda P)(P^{-1}\Lambda^{-1}S)$ for any diagonal matrix Λ (with nonzero diagonals) and permutation matrix P, X can never have completely unique representation.

The reason is that, both S and A being unknown, any scalar multiplier in one of the sources s_i could always be canceled by dividing the corresponding column a_i of A by the same scalar. As a consequence, we usually fixes the magnitudes of the independent components by assuming each s_i has unit variance. Then the matrix A will be adapted in the ICA solution methods to take into account this restriction. However, this still leaves the ambiguity of the sign: we could multiply an independent component by -1 without affecting the model. This ambiguity is insignificant in most applications.

4 Our AK-ICA Attack

In this section we present our AK-ICA attack which may be exploited by attackers when a subset of sample data is available. Let $\tilde{X} \subset X$ denote this sample data set consisting of k data records and d attributes. Our result shall show attackers can reconstruct X closely by applying our AK-ICA attack method when a (even small) sample of data, $\tilde{X}$, is available to attackers.

The core idea of AK-ICA is to apply the traditional ICA on the known sample data set, $\tilde{X}$, and perturbed data set, Y, to get their mixing matrices and independent components respectively, and reconstruct the original data by exploiting the relationships between them. Figure 1 shows our AK-ICA based attack.

The first step of this attack is to derive ICA representations, $(A_{\tilde{x}}, S_{\tilde{x}})$ and (A_y, S_y), from the a-priori known subset $\tilde{X}$ and the perturbed data Y respectively. Since in general we can not find the unique representation of (A, S) for a given X (recall that $X = AS = (A\Lambda P)(P^{-1}\Lambda^{-1}S)$ for any diagonal matrix Λ and perturbation matrix P in Section 2), S is usually required to have unit variance to avoid scale issue in ICA. As a consequence, only the order and sign of the signals S might be different. In the following, we shall prove there exists a transformation matrix J such that $\hat{X} = A_{\tilde{x}} J S_y$ is an estimate of the original data X in Section 4.1, and present how to identity J in Section 4.2.

4.1 Existence of Transformation Matrix *J*

To derive the permutation matrix J, let us first assume X is given. Applying the independent component analysis, we get $X = A_x S_x$ where A_x is the mixing matrix and S_x is independent signal.

input Y, a given perturbed data set
$\tilde{X}$, a given subset of original data
output $\hat{X}$, a reconstructed data set

BEGIN
1 Applying ICA on $\tilde{X}$ and Y to get
$$\tilde{X} = A_{\tilde{x}} S_{\tilde{x}}$$
$$Y = A_y S_y$$
2 Deriving the transformation matrix J by comparing the distributions of $S_{\tilde{x}}$ and S_y
3 Reconstructing X approximately as
$$\hat{X} = A_{\tilde{x}} J S_y$$
END

Fig. 1. AK-ICA Attack

Proposition 1. *The mixing matrices A_x, $A_{\tilde{x}}$ are expected to be close to each other and the underlying signals $S_{\tilde{x}}$ can be approximately regarded as a subset of S_x.*

$$\begin{aligned} A_{\tilde{x}} &\approx A_x \Lambda_1 P_1 \\ S_{\tilde{x}} &\approx P_1^{-1} \Lambda_1^{-1} \tilde{S}_x \end{aligned} \tag{2}$$

Proof. Considering an element x_{ij} in X, it is determined by the i-th row of A_x, $\boldsymbol{a}_i$, and the j-th signal vector, $\boldsymbol{s}_j$, where $\boldsymbol{a}_i = (a_{i1}, a_{i2}, \cdots, a_{id})$ and $\boldsymbol{s}_j = (s_{1j}, s_{2j}, \cdots, s_{dj})^T$.

$$x_{ij} = a_{i1}s_{1j} + a_{i2}s_{2j} + \cdots + a_{id}s_{dj}$$

Let $\tilde{\boldsymbol{x}}_p$ be a column vector in $\tilde{X}$ which is randomly sampled from X. Assume $\tilde{\boldsymbol{x}}_p = \boldsymbol{x}_j$, then the i-th element of this vector, $\tilde{x}_{ip}$ can also be expressed by $\boldsymbol{a}_i$ and the corresponding signal vector $\boldsymbol{s}_j$.

$$\tilde{x}_{ip} = a_{i1}s_{1j} + a_{i2}s_{2j} + \cdots + a_{ip}s_{pj}$$

Thus, for a given column vector in $\tilde{X}$, we can always find a corresponding signal vector in S and reconstruct it through the mixing matrix A_x. Since S_x is a set of independent components, its sample subset $\tilde{S}_x \subset S_x$ can also be regarded as a set of independent components of $\tilde{X}$ when the sample size of $\tilde{X}$ is large.

There exists a diagonal matrix Λ_1 and a permutation matrix P_1 such that

$$\begin{aligned} \tilde{X} = A_{\tilde{x}} S_{\tilde{x}} &\approx A_x \tilde{S}_x = (A_x \Lambda_1 P_1)(P_1^{-1} \Lambda_1^{-1} \tilde{S}_x) \\ A_{\tilde{x}} &\approx A_x \Lambda_1 P_1 \\ S_{\tilde{x}} &\approx P_1^{-1} \Lambda_1^{-1} \tilde{S}_x \end{aligned}$$

Proposition 2. *S_x and S_y are similar to each other and there exists a diagonal matrix Λ_2 and a permutation matrix P_2 that*

$$S_y = P_2^{-1} \Lambda_2^{-1} S_x \tag{3}$$

Proof.

$$Y = RX + E = R(A_x S_x) + E = (RA_x)S_x + E$$

Since permutation may affect the order and phase of the signals S_y, we have

$$Y = A_y S_y + E \approx (RA_x \Lambda_2 P_2)(P_2^{-1} \Lambda_2^{-1} S_x) + E$$

By comparing the above two equations, we have

$$A_y \approx RA_x \Lambda_2 P_2$$
$$S_y \approx P_2^{-1} \Lambda_2^{-1} S_x$$

Theorem 1. *Existence of J. There exists one transformation matrix J such that*

$$\hat{X} = A_{\tilde{x}} J S_y \approx X \tag{4}$$

where $A_{\tilde{x}}$ is the mixing matrix of $\tilde{X}$ and S_y is the independent components of the perturbed data Y.

Proof. Since

$$S_{\tilde{x}} \approx P_1^{-1} \Lambda_1^{-1} \tilde{S}_x$$
$$S_y \approx P_2^{-1} \Lambda_2^{-1} S_x$$

and $\tilde{S}_x$ is a subset of S_x, we can find a transformation matrix J to match the independent components between S_y and $S_{\tilde{x}}$. Hence,

$$J P_2^{-1} \Lambda_2^{-1} = P_1^{-1} \Lambda_1^{-1}$$
$$J = P_1^{-1} \Lambda_1^{-1} \Lambda_2 P_2$$

From Equation 2 and 3 we have

$$\begin{aligned} \hat{X} &= A_{\tilde{x}} J S_y \\ &\approx (A_x \Lambda_1 P_1)(P_1^{-1} \Lambda_1^{-1} \Lambda_2 P_2)(P_2^{-1} \Lambda_2^{-1} S_x) \\ &= A_x S_x \\ &= X \end{aligned}$$

4.2 Determining J

The ICA model given in Definition 1 implies no ordering of the independent components. The reason is that, both $\mathbf{s}$ and A being unknown, we can freely change the order of the terms in the sum in Definition 1, and call any of the independent component as the first one. Formally, a permutation matrix P and its inverse can be substituted in the model to give another solution in another order. As a consequence, in our case, the i-th component in S_y may correspond to the j-th component in $S_{\tilde{x}}$. Hence we need to figure out how to find the transformation matrix, J.

Since $S_{\tilde{x}}$ is a subset of S_x, each pair of corresponding components follow similar distributions. Hence our strategy is to analyze distributions of two signal data sets, $S_{\tilde{x}}$ and S_y. As we discussed before, the signals derived by ICA are normalized signals. So the scaler for each attribute is either 1 or -1. It also can be easily indicated by the distributions.

Let $S_{\tilde{x}}^{(i)}$ and $S_y^{(j)}$ denote the i-th component of $S_{\tilde{x}}$ and the j-th component of S_y and let f_i and $f_j^{'}$ denote their density distribution respectively. In this paper, we use the information difference measure $\mathcal{I}$ to measure the similarity of two distributions [1].

$$\mathcal{I}(f_i, f_j^{'}) = \frac{1}{2} E[\int_{\Omega_Z} \mid f_i(z) - f_j^{'}(z) \mid dz] \tag{5}$$

The above metric equals half the expected value of L_1-norm between the distribution of the i-th component from $S_{\tilde{x}}$ and that of the j-th component from S_y. It is also equal to $1 - \alpha$, where α is the area shared by both distributions. The smaller the $\mathcal{I}(f, f')$, the more similar between one pairs of components. The matrix J is determined so that $J[f_1^{'}, f_2^{'}, \cdots, f_d^{'}]^T \approx [f_1, f_2, \cdots, f_d]^T$.

5 Empirical Evaluations

The data set we used in our experiments is a Bank data set which was previously used in [12]. This data set contains 5 attributes (Home Equity, Stock/Bonds, Liabilities, Savings, and CDs) and 50,000 records. In our AK-ICA method, we applied JADE package[1] implemented by Jean-Francois Cardoso to conduct ICA analysis. JADE is one cumulant-based batch algorithm for source separation [3].

Since our AK-ICA attack can reconstruct individual data in addition to its distribution, in this paper we cast our accuracy analysis in terms of both matrix norm and individual-wise errors. We measure the reconstruction errors using the following measures:

$$\text{RE}(X, \tilde{X}) = \frac{1}{d \times N} \sum_{i=1}^{d} \sum_{j=1}^{N} \mid \frac{x_{ij} - \hat{x}_{ij}}{x_{ij}} \mid$$

$$\text{RE-R}_i(X, \tilde{X}) = \frac{1}{N} \sum_{j=1}^{N} \mid \frac{x_{ij} - \hat{x}_{ij}}{x_{ij}} \mid \quad i = 1, \cdots, d$$

$$\text{F-RE}(X, \tilde{X}) = \frac{\|\tilde{X} - X\|_F}{\|X\|_F}$$

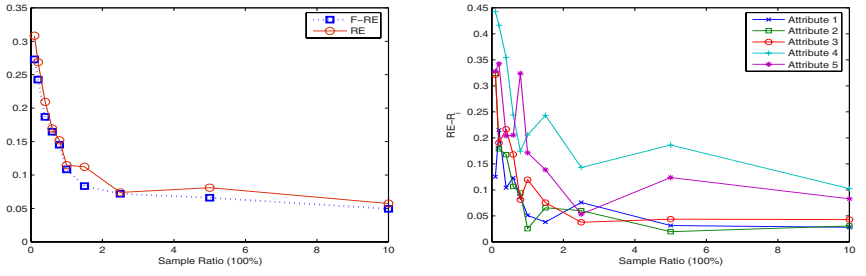
where $X, \hat{X}$ denotes the original data and the estimated data respectively, and $\| \cdot \|_F$ denotes a Frobenius norm [2].

All the above measures show how closely one can estimate the original data X from its perturbed data Y. Here we follow the tradition of using the difference as the measure

[1] http://www.tsi.enst.fr/icacentral/algos.html

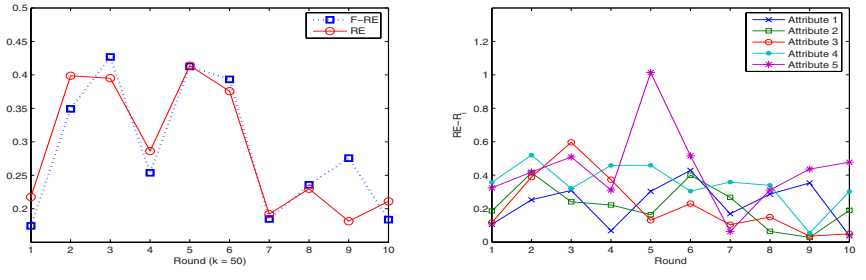
[2] The Frobenius norm of X: $\|X\|_F = \sqrt{\sum_{i=1}^{d} \sum_{j=1}^{n} x_{ij}^2}$.

to quantify how much privacy is preserved. Basically, RE (relative error) represents the average of relative errors of individual data points. RE-R_i represents the average of relative errors of the i-th attribute.F-RE denotes the relative errors between X and its estimation $\hat{X}$ in terms of Frobenius norm, which gives perturbation evaluation a simplicity that makes it easier to interpret.



(a) F-RE and RE (b) RE-R_i

Fig. 2. Reconstruction error vs. varying sample ratio



(a) F-RE and RE (b) RE-R_i

Fig. 3. Reconstruction error vs. random samples with the fixed size $k = 50$

5.1 Changing the Sample Size

In this experiment, we first evaluate how the sample size affects the accuracy of reconstruction of AK-ICA method. We change the ratio between known sample size and the original data size from 0.1% to 10%. Please note that all sizes of known samples in this experiment are small compared with the size of the original data. We set the transformation matrix R as non-orthonormal by generating all its elements from a uniform distribution.

Figure 2 shows the reconstruction error (in terms of F-RE and RE in Figure 2(a) and RE-R_i for each attribute in Figure 2(b)) decreases when the sample size is increased. This is because that the more sample data we have, the more match between

derived independent components. When we have known records which account for 1% of the original data, we could achieve very low reconstruction error (F-RE = 0.108, RE = 0.115). When the sample size is decreased, more errors are introduced. However, even with known samples which only account for 0.1% of the original data, we can still achieve very close estimations for some attributes (e.g., RE-R_i = 0.125 for attribute 1).

Next we evaluate how different sample sets $\tilde{X}$ with the same size affect AK-ICA reconstruction method, especially when sample size is very small. Here we randomly chose 10 different sample sets with the fixed size $k = 50$ (sample ratio 0.1%). Figure 3 shows the construction errors with 10 different sample sets. The performance of our AK-ICA reconstruction method is not very stable in this small sample case. For example, the first run achieves 0.1 of F-RE while the third run achieves 0.44 as shown in Figure 3(a). The instability here is mainly caused by $A_{\tilde{x}}$ which is derived from $\tilde{X}$. Since $Y = RX$ is fixed, the derived S_y doesn't change.

We also observed that for each particular attribute, its reconstruction accuracy in different rounds is not stable either. As shown in Figure 3(b), the attribute 5 has the largest error among all the attributes in round 5, however, it has the smallest error in round 7. This is because the reconstruction accuracy of one attribute is mainly determined by the accuracy of its estimate of the corresponding column vector in $A_{\tilde{x}}$. This instability can also be observed in Figure 2(b). We plan to theoretically investigate how the sample's properties(size, distribution etc.) affect reconstruction accuracy of the proposed AK-ICA attack. We also plan to investigate how the distribution of data affects reconstruction accuracy when a sample data set is fixed. As we point out in the future work, both problems are very challenging since there is no study on this problem in statistics.

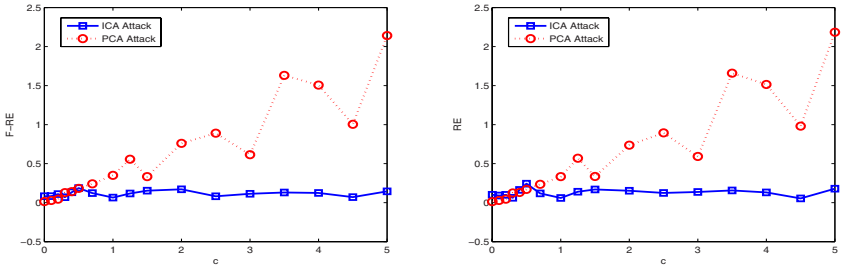
5.2 Comparing AK-ICA and PCA Attack

In this experiment, we evaluate the reconstruction performance of our approach and the PCA attack in [10]. We fix the sample ratio as 1% and apply different transformation matrices. Here R is expressed as $R = R_1 + cR_2$, where R_1 is a random orthonormal matrix, R_2 is a random matrix with uniformly distributed elements([-0.5,0.5]) and c is a coefficient. Initially, c is set as 0 which guarantees the orthonormal property for R. By increasing c, R gradually loses orthonormal property and tends to be an arbitrary transformation.

From Figure 4(a) and 4(b) we can observe that our AK-ICA attack is robust to various transformations. The reconstruction errors do not change much when the transformation matrix R is changed to more non-orthonormal. On the contrary, the PCA attack only works when R is orthonormal or close to orthonormal. When the transformation tends to be more non-orthonormal (with the increase of c as shown in Table 1), the reconstruction accuracy of PCA attack degrades significantly. For example, when we set $c = 5$, the relative reconstruction errors of PCA attack are more than 200% (F-RE=2.1414 , RE = 2.1843) while the relative reconstruction errors of AK-ICA attack are less than 20% (F-RE=0.1444 , RE = 0.1793).

Table 1. Reconstruction error of AK-ICA vs. PCA attacks by varying R

c	$\frac{\|\|cR_2\|\|_F}{\|\|R_1\|\|_F}$	AK-ICA		PCA		c	$\frac{\|\|cR_2\|\|_F}{\|\|R_1\|\|_F}$	AK-ICA		PCA	
		F-RE	RE	F-RE	RE			F-RE	RE	F-RE	RE
0	0	0.0824	0.1013	0.013	0.0126	1.5	0.8059	0.1533	0.169	0.3336	0.3354
0.2	0.1299	0.1098	0.1003	0.0451	0.0448	2	1.2755	0.1709	0.1523	0.7598	0.7368
0.3	0.1988	0.0701	0.0618	0.1288	0.1247	2.5	1.5148	0.0816	0.1244	0.8906	0.8946
0.4	0.3121	0.1336	0.1631	0.1406	0.1305	3	1.9321	0.1142	0.1373	0.6148	0.592
0.5	0.3011	0.1867	0.2436	0.1825	0.1704	3.5	2.1238	0.1303	0.1566	1.631	1.6596
0.7	0.4847	0.1227	0.1188	0.2415	0.2351	4	2.4728	0.1249	0.1314	1.5065	1.5148
1	0.539	0.065	0.0606	0.35	0.334	4.5	3.049	0.0707	0.0543	1.0045	0.9815
1.25	0.804	0.1177	0.1399	0.5565	0.5695	5	3.4194	0.1444	0.1793	2.1414	2.1843



(a) F-RE (b) RE

Fig. 4. Reconstruction error of AK-ICA vs. PCA attacks by varying R

6 Conclusion

In this paper, we have examined the effectiveness of general projection in privacy preserving data mining. It was suggested in [10] that the non-isometric projection approach is effective to preserve privacy since it is resilient to the PCA attack which was designed for the distance preserving projection approach. We proposed an AK-ICA attack, which can be exploited by attackers to breach the privacy from the non-isometric transformed data. Our theoretical analysis and empirical evaluations have shown the proposed attack poses a threat to all projection based privacy preserving methods when a small sample data set is available to attackers. We argue this is really a concern that we need to address in practice.

We noticed that the sample's properties (size, distribution etc.) would affect the reconstruction accuracy from our empirical evaluations. It is a very challenging topic to explore the theoretical relationship between those properties and the reconstruction accuracy. To our knowledge, there is no study on this topic in statistics. We plan to tackle this issue with researchers in statistics in our future work. We would also investigate how transformation matrix affects the data utility.

Acknowledgments

This work was supported in part by U.S. National Science Foundation IIS-0546027.

References

1. D. Agrawal and C. Agrawal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th Symposium on Principles of Database Systems*, 2001.
2. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 439–450. Dallas, Texas, May 2000.
3. J. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1):157–192, 1999.
4. K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In *Proceedings of the 5th IEEE International Conference on Data Mining*. Houston,TX, Nov 2005.
5. S. Guo and X. Wu. On the use of spectral filtering for privacy preserving data mining. In *Proceedings of the 21st ACM Symposium on Applied Computing*. Dijion,France, April 2006.
6. S. Guo, X. Wu, and Y. Li. On the lower bound of reconstruction error for spectral filting based privacy preserving data mining. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'06)*. Berlin,Germany, September 2006.
7. Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *Proceedings of the ACM SIGMOD Conference on Management of Data*. Baltimore, MA, 2005.
8. A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
9. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Proceedings of the 3rd International Conference on Data Mining*, pages 99–106, 2003.
10. K. Liu, C. Giannella, and H. Kargupta. An attacker's view of distance preserving maps for privacy preserving data mining. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'06)*. Berlin,Germany, September 2006.
11. K. Liu, H. Kargupta, and J. Ryan. Random projection based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transaction on Knowledge and Data Engineering*, 18(1):92–106, 2006.
12. K. Muralidhar and R. Sarathy. A general additive data perturbation method for database security. *Management Science*, 45(10):1399–1415, 1999.
13. S. Oliveira and O. Zaiane. Achieving privacy preservation when sharing data for clustering. In *Proceedings of the Workshop on Secure Data Management in a Connected World*, pages 67–82. Toronto,Canada, August 2004.

Consistency Based Attribute Reduction

Qinghua Hu, Hui Zhao, Zongxia Xie, and Daren Yu

Harbin Institute of Technology, Harbin 150001, P.R. China
huqinghua@hcms.hit.edu.cn

Rough sets are widely used in feature subset selection and attribute reduction. In most of the existing algorithms, the dependency function is employed to evaluate the quality of a feature subset. The disadvantages of using dependency are discussed in this paper. And the problem of forward greedy search algorithm based on dependency is presented. We introduce the consistency measure to deal with the problems. The relationship between dependency and consistency is analyzed. It is shown that consistency measure can reflects not only the size of decision positive region, like dependency, but also the sample distribution in the boundary region. Therefore it can more finely describe the distinguishing power of an attribute set. Based on consistency, we redefine the redundancy and reduct of a decision system. We construct a forward greedy search algorithm to find reducts based on consistency. What's more, we employ cross validation to test the selected features, and reduce the overfitting features in a reduct. The experimental results with UCI data show that the proposed algorithm is effective and efficient.

1 Introduction

As the capability of gathering and storing data increases, there are a lot of candidate features in some pattern recognition and machine learning tasks. Applications show that excessive features will not only significantly slow down the learning process, but also decrease the generalization power of the learned classifiers. Attribute reduction, also called feature subset selection, is usually employed as a preprocessing step to select part of the features and focuses the learning algorithm on the relevant information [1, 3, 4, 5, 7, 8]. In recent years, rough set theory has been widely discussed and used in attribute reduction and feature selection [6, 7, 8, 14, 16, 17]. Reduct is a proper term in rough set methodology. It means a minimal attribute subset with the same approximating power as the whole set [14]. This definition shows that a reduct should have the least redundant information and not loss the classification ability of the raw data. Thus the attributes in a reduct should not only be strongly relevant to the learning task, but also be not redundant with each other. This property of reducts exactly accords with the objective of feature selection. Thereby, the process of searching reducts, called attribute reduction, is a feature subset selection process. As so far, a series of approaches to search reducts have been published. Discernibility Matrices [11, 14] were introduced to store the features which can distinguish the corresponding pair of objects, and then Boolean operations were conducted on the matrices to search all of the reducts. The main problem of this method is space and

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 96–107, 2007.

time cost. We need a $10^4 \times 10^4$ matrix if there are 10^4 samples. What's more, it is also time-consuming to search reducts from the matrix with Boolean operations. With the dependency function, a heuristic search algorithm was constructed [1, 6, 7, 8, 16].

There are some problems in dependency based attribute reduction. The dependency function in rough set approaches is the ratio of sizes of the positive region over the sample space. The positive region is the sample set which can be undoubtedly classified into a certain class according to the existing attributes. From the definition of the dependency function, we can find that it ignores the influence of boundary samples, which maybe belong to more than one class. However, in classification learning, the boundary samples also exert an influence on the learned results. For example, in learning decision trees with CART or C4.5 learning, the samples in leaf nodes sometimes belong to more than one class [2, 10]. In this case, the nodes are labeled with the class with majority of samples. However, the dependency function does not take this kind of samples into account. What's more, there is another risk in using the dependency function in greedy feature subset search algorithms. In a forward greedy search, we usually start with an empty set of attribute, and then we add the selected features into the reduct one by one. In the first round, we need to compute the dependency of each single attribute, and select the attribute with the greatest dependency value. We find that the greatest dependency of a single attribute is zero in some applications because we can not classify any of the samples beyond dispute with any of the candidate features. Therefore, according to the criterion that the dependency function should be greater than zero, none of the attributes can be selected. Then the feature selection algorithm can find nothing. However, some combinations of the attributes are able to distinguish any of the samples although a single one cannot distinguish any of them. As much as we know, there is no research reporting on this issue so far.

These issues essentially result from the same problem of the dependency function. It completely neglects the boundary samples. In this paper, we will introduce a new function, proposed by Dash and Liu [3], called consistency, to evaluate the significance of attributes. We discuss the relationship between dependency and consistency, and employ the consistency function to construct greedy search attribute reduction algorithm. The main difference between the two functions is in considering the boundary samples. Consistency not only computes the positive region, but also the samples of the majority class in boundary regions. Therefore, even if the positive region is empty, we can still compare the distinguishing power of the features according to the sample distribution in boundary regions. Consistency is the ratio of consistent samples; hence it is linear with the size of consistent samples. Therefore it is easy to specify a stopping criterion in a consistency-based algorithm. With numerical experiments, we will show the specification is necessary for real-world applications.

In the next section, we review the basic concepts on rough sets. We then present the definition and properties of the consistency function, compare the dependency function with consistency, and construct consistency based attribute reduction in section 3.We present the results of experiments in section 4. Finally, the conclusions are presented in section 5.

2 Basic Concepts on Rough Sets

Rough set theory, which was introduced to deal with imperfect and vague concepts, has attracted a lot of attention from theory and application research areas. Data sets are usually given as the form of tables, we call a data table as an information system, formulated as $IS = < U, A, V, f >$, where $U = \{x_1, x_2, \cdots x_n\}$ is a set of finite and nonempty objects, called the universe, A is the set of attributes characterizing the objects, V is the domain of attribute value and f is the information function $f : U \times A \to V$. If the attribute set is divided into condition attribute set C and decision attribute set D, the information system is also called a decision table.

With arbitrary attribute subset $B \subseteq A$, there is an indiscernibility relation $IND(B)$:

$$IND(B) = \{< x, y >\in U \times U \mid \forall a \in B, a(x) = a(y)\}.$$

$< x, y >\in IND(B)$ means objects x and y are indiscernible with respect to attribute set B. Obviously, indiscernibility relation is an equivalent relation, which satisfies the properties of reflexivity, symmetry and transitivity. The equivalent class induced by the attributes B is denoted by

$$[x_i]_B = \{x \mid < x_i, x >\in IND(B), y \in U\}.$$

Equivalent classes generated by B are also called B-elemental granules, B-information granules. The set of elemental granules forms a concept system, which is used to characterize the imperfect concepts in the information system.

Given an arbitrary concept X in the information system, two unions of elemental granules are associated with

$$\underline{B}X = \{[x]_B \mid [x]_B \subseteq X, x \in U\},\ \overline{B}X = \{[x]_B \mid [x]_B \cap X \neq \varnothing, x \in U\}.$$

The concept X is approximated with the two sets of elemental granules. $\underline{B}X$ and $\overline{B}X$ are called lower and upper approximations of X in terms of attributes B. $\underline{B}X$ is also called the positive region. X is a definable if $\underline{B}X = \overline{B}X$, which means the concept X can be perfectly characterized with the knowledge B, otherwise, X is indefinable. An indefinable set is called a rough set. $BND(X) = \overline{B}X - \underline{B}X$ is called the boundary of the approximations. As a definable set, the boundary is empty.

Given $<U, C, D, V, f>$, C and D will generates two partitions of the universe. Machine learning is usually involved in using condition knowledge to approximate the decision and finding the mapping from the conditions to decisions. Approximating U / D with U / C, the positive and boundary regions are defined as:

$$POS_C(D) = \bigcup_{X \in U/D} \underline{C}X,\ BND_C(D) = \bigcup_{X \in U/D} \overline{C}X - \bigcup_{X \in U/D} \underline{C}X.$$

The boundary region is the set of elemental granules which can not be perfectly described by the knowledge C, while the positive region is the set of C-elemental granules which completely belong to one of the decision concepts. The size of positive or boundary regions reflects the approximation power of the condition

attributes. Given a decision table, for any $B \subseteq C$, it is said the decision attribute set D depends on the condition attributes with the degree k, denoted by $B \Rightarrow_k D$, where

$$k = \gamma_B(D) = \frac{|POS_B(D)|}{|U|}.$$

The dependency function k measures the approximation power of a condition attribute set with respect to the decision D. In data mining, especially in feature selection, it is important to find the dependence relations between attribute sets and to find a concise and efficient representation of the data.

Given a decision table $DT =< U, C \bigcup D, V, f >$, if $P \subseteq Q \subseteq C$, we have

$$\gamma_Q(D) \geq \gamma_P(D)$$

Given a decision table $DT =< U, C \bigcup D, V, f >$, $B \subseteq C$, $a \in B$, we say that the condition attribute a is indispensable if $\gamma_{(B-a)}(D) < \gamma_B(D)$, otherwise we say a is redundant. We say $B \subseteq C$ is independent if any a in B is indispensable. Attribute subset B is a reduct of the decision table if

1) $\gamma_B(D) = \gamma_C(D)$;
2) $\forall a \in B: \gamma_B(D) > \gamma_{B-a}(D)$.

A reduct of a decision table is the attribute subset which keeps the approximating capability of all the condition attributes. In the meantime it has no redundant attribute. The term of "reduct" presents a concise and complete ways to define the objective of feature selection and attribute reduction.

3 Consistency Based Attribute Reduction

A binary classification problem in discrete spaces is shown in Figure 1, where the samples are divided into a finite set of equivalence classes $\{E_1, E_2, \cdots, E_K\}$ based on their feature values. The samples with the same feature values are grouped into one equivalence class. We find that some of the equivalence classes are pure as their samples belong to one of decision classes, but there also are some inconsistent equivalence classes, such as E_3 and E_4 in figure1. According to rough set theory, they are named as decision boundary region, and the set of consistent equivalence classes is named as decision positive region. The objective of feature selection is to find a feature subset which minimizes the inconsistent region, in either discrete or numerical cases, accordingly, minimizes Bayesian decision error. It is therefore desirable to have a measure to reflect the size of inconsistent region for discrete and numerical spaces for feature selection. Dependency reflects the ratio of consistent samples over the whole set of samples. Therefore dependency doesn't take the boundary samples into account in computing significance of attributes. Once there are inconsistent samples in an equivalence class, these equivalence classes are just ignored. However, inconsistent samples can be divided into two groups: a subset of samples under the majority class and a subset under the minority classes. According

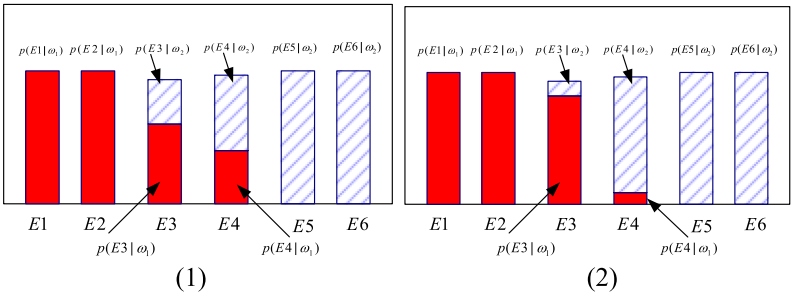


Fig. 1. Classification complexity in a discrete feature space

to Bayesian rule, only the samples under the minority classes are misclassified. Fox example, the samples in E_3 and E_4 are inconsistent in figure 1. But only the samples labeled with $P(E_3 \mid \omega_2)$ and $P(E_4 \mid \omega_1)$ are misclassified. The classification power in this case can be given by

$$f = 1 - \left[P(E_3 \mid \omega_2)P(E_3) - P(E_4 \mid \omega_1)P(E_4)\right].$$

Dependency can not reflect the true classification complexity. In the discrete cases, we can see from comparison of figure 1 (1) and (2) although the probabilities of inconsistent samples are identical, the probabilities of misclassification are different. Dependency function in rough sets can not reflect this difference.

In [3], Dash and Liu introduced the consistency function which can measure the difference. Now we present the basic definition on consistency. Consistency measure is defined by inconsistency rate, computed as follows.

Definition 1. A pattern is considered to be inconsistent if there are at least two objects such that they match the whole condition attribute set but are with different decision label.

Definition 2. The inconsistency count ξ_i for a pattern p_i of feature subset is the number of times it appears in the data minus the largest number among different class labels.

Definition 3. The inconsistency rate of a feature subset is the sum, $\sum \xi_i$, of all the inconsistency counts over all patterns of the feature subset that appears in data divided by $|U|$, the size of all samples, namely $\sum \xi_i / |U|$. Correspondingly, consistency is computed as $\delta = \left(|U| - \sum \xi_i\right) / |U|$.

Based on the above analysis, we can understand that dependency is the ratio of samples undoubtedly correctly classified, and consistency is the ratio of samples probably correctly classified.

There are two kinds of samples in $POS_B(D) \bigcup M$. $POS_B(D)$ is the set of consistent samples, while M is the set of the samples with the largest number among different class labels in the boundary region. In the paper, we will call M pseudo-consistent samples.

Property 1: Given a decision table <*U*, *C*, *D*, *f*>, $\forall B \subseteq C$, we have $0 \le \delta_B(D) \le 1$, $\gamma_B(D) \le \delta_B(D)$.

Property 2 (monotonicity): Given a decision table <*U*, *C*, *D*, *f*>, if $B_1 \subseteq B_2 \subseteq D$, we have $\delta_{B1}(D) \le \delta_{B2}(D)$.

Property 3: Given a decision table <*U*, *C*, *D*, *f*>, if and only if $U/C \subseteq U/D$, namely, the table is consistent, we have $\delta_C(D) = \gamma_C(D) = 1$

Definition 4. Given a decision table $DT = < U, C \bigcup D, V, f >$, $B \subseteq C$, $a \in B$, we say condition attribute a is indispensable in *B* if $\delta_{(B-a)}(D) < \delta_B(D)$, otherwise; we say *a* is redundant. We say $B \subseteq C$ is independent if any attribute *a* in *B* is indispensable.

$\delta_B(D)$ reflects not only the size of positive regions, but also the distribution of boundary samples. The attribute is said to be redundant if the consistency doesn't decrease when we delete it. Here the term "redundant" has two meanings. The first one is relevant but redundant, the same as the meaning in general literatures [6, 7, 8, 14, 16, 17]. The second meaning is irrelevant. So consistency can detect the two kinds of superfluous attributes [3].

Definition 5. Attribute subset *B* is a consistency-based reduct of the decision table if

(1) $\delta_B(D) = \delta_C(D)$;
(2) $\forall a \in B: \delta_B(D) > \delta_{B-a}(D)$.

In this definition, the first term guarantees the reduct has the same distinguishing ability as the whole set of features; the second one guarantees that all of the attributes in the reduct are indispensable. Therefore, there is not any superfluous attribute in the reduct.

Finding the optimal subset of features is a NP-hard problem. We require evaluating $2^N - 1$ combinations of features for find the optimal subset if there are N features in the decision table. Considering computational complexity, here we construct a forward greedy search algorithm based on the consistency function. We start with an empty set of attribute, and add one attribute into the reduct in a round. The selected attribute should make the increment of consistency maximal. Knowing attribute subset *B*, we evaluate the significance of an attribute *a* as

$$SIG(a, B, D) = \delta_{B \cup a}(D) - \delta_B(D).$$

$SIG(a, B, D)$ is the increment of consistency by introducing the new attribute *a* in the condition of *B*. The measure is linear with the size of the new consistent and pseudo-consistent samples. Formally, a forward greedy reduction algorithm based on consistency can be formulated as follows.

Algorithm: Greedy Reduction Algorithm based on Consistency
Input: Decision table $< U, C \bigcup d, f >$
Output: One reduct red .
Step 1: $\varnothing \to red$; // red is the pool to contain the selected attributes.
Step 2: For each $a_i \in A - red$ Compute

$$SIG(a_i, red, D) = \delta_{red \cup a_i}(D) - \delta_{red}(D)$$

end
Step 3: select the attribute a_k which satisfies:

$$SIG(a_k, red, D) = \max_i (SIG(a_i, red, B))$$

Step 4: If $SIG(a_k, red, D) > 0$,
$red \bigcup a_k \to red$
go to step2
else
return red
Step 5: end

In the first round, we start with an empty set, then specify $\delta_{\varnothing}(D) = 0$. In this algorithm, we generate attribute subsets with a semi-exhaustive search. Namely, we evaluate all of the rest attributes in each round with the consistency function, and select the feature producing the maximal significance. The algorithm stops when adding any of the rest attributes will not bring increment of consistency value. In real-world applications, we can stop the algorithm if the increment of consistency is less than a given threshold to avoiding the over-fitting problem. In section 4, we will discuss this problem in detail. The output of the algorithm is a reduced decision table. The irrelevant, relevant and redundant attributes are deleted from the system. The output results will be validated with two popular learning algorithms: CART and SVM, in section 4.

By employing a hashing mechanism, we can compute the inconsistency rate approximately with a time complexity of $O(|U|)$ [3]. In the worst case the whole computational complexity of the algorithm can be computed as

$$|U| \times |C| + |U| \times (|C| - 1) + \cdots + |U| = (|C| + 1) \times |C| \times |U| / 2 .$$

4 Experimental Analysis

There are two main objectives to conduct the experiments. First, we compare the proposed method with dependency based algorithm. Second, we study the classification performance of the attributes selected with the proposed algorithm, In particular, how the classification accuracy varies with adding a new feature. This can tell us where the algorithm should be stopped.

We download data sets from UCI Repository of machine learning databases. The data sets are described in table 1. There are some numerical attributes in the data sets. Here we employ four discretization techniques to transform the numerical data into

Table 1. Data description

Data set	Abbreviation	Samples	Features	Classes
Australian Credit Approval	Crd	690	15	2
Ecoli	Ecoli	336	7	7
Heart disease	Heart	270	13	2
Ionosphere	Iono	351	34	2
Sonar, Mines vs. Rocks	Sonar	208	60	2
Wisconsin Diagnostic Breast Cancer	WDBC	569	31	2
Wisconsin Prognostic Breast Cancer	WPBC	198	33	2
Wine recognition	Wine	178	13	3

categorical one: equal-width, equal-frequency, FCM and entropy. Then we conduct the dependency based algorithm [8] and the proposed one on the discretized data sets. The numbers of the selected features are presented in table 2.

From table 2, we can find there is a great problem with dependency based algorithm, where, P stands for dependency based algorithm, and C stands for consistency based algorithm. The algorithm selects two few feature for classification learning as to some data sets. As to the discretized data with Equal-width method, the dependency based algorithm only selects one attribute, while the consistency one selects 7 attributes. As to Equal-frequency method, the dependency based algorithm selects nothing for data sets Heart, Sonar and WPBC. The similar case occurs to Entropy and FCM based discretization methods. Obviously, the results are unacceptable if a feature selection algorithm cannot find anything. By contrast, the consistency based attribute reduction algorithm finds feature subsets with moderate sizes for all of the data sets. What's more, the sizes of selected features with the two algorithms are comparable if the dependency algorithm works well.

Why does the dependency based algorithm find nothing for some data sets? As we know, dependency just reflects the ratio of positive regions. The forward greedy algorithm starts off with an empty set and adds, in turn, one of the best attributes into the pool at a time, those attributes that result in the greatest increase in the dependency function, until this produces its maximum possible value for the data set. In the first turn, we need to evaluate each single attribute. For some data sets, the dependency is zero for each single attribute. Therefore, no attribute can be added into the pool in the first turn. Then the algorithm stops here. Sometimes, the algorithm can

Table 2. The numbers of selected features with different methods

	Raw	Equal-width		Equal-frequency		Entropy		FCM	
	data	P	C	P	C	P	C	P	C
Crd	15	11	11	9	9	11	11	12	11
Ecoli	7	6	6	7	7	**1**	7	**1**	6
Heart	13	10	9	**0**	8	**0**	11	**0**	8
Iono	34	**1**	7	1	7	10	8	10	9
Sonar	60	7	7	**0**	6	**0**	14	6	6
WDBC	30	12	12	6	6	7	7	8	10
WPBC	33	9	10	**0**	6	11	7	7	7
Wine	13	5	4	4	4	4	5	4	4
Aver.	25.63	7.63	8.25	--	6.63	--	8.75	--	7.63

also stop in the second turn or the third turn. However, the selected features are not enough for classification learning. Consistency can overcome this problem as it can reflect the change in distribution of boundary samples.

Now we use the selected data to train classifiers with CART and SVM learning algorithms. We test the classification power of the selected data with 10-fold cross validation. The average classification accuracies with CART and SVM are presented in tables 3 and 4, respectively. From table 3, we can find most of the reduced data can keep, even improve the classification power if the numbers of selected attributes are appropriate although most of the candidate features are deleted from the data. It shows that most of the features in the data sets are irrelevant or redundant for training decision trees; thereby, it should be deleted. However, the classification performance will greatly decrease if the data are excessively reduced, such as iono in the equal-width case and ecoli in the entropy and FCM cases**.**

Table 3. Classification accuracy with 10-fold cross validation (CART)

	Raw	Equal-width		Equal-frequency		Entropy		FCM	
	data	P	C	P	C	P	C	P	C
Crd	0.8217	0.8246	0.8246	0.8346	0.8150	0.8288	0.8186	0.8274	0.8158
Ecoli	0.8197	0.8138	0.8138	0.8197	0.8138	**0.4262**	0.8168	**0.42620**	0.8168
Heart	0.7407	0.7630	0.7630	**0**	0.7704	**0**	0.7630	**0**	0.7815
Iono	0.8755	**0.7499**	0.9064	0.7499	0.9064	0.9318	0.8922	0.9089	0.9062
Sonar	0.7207	0.7024	0.7014	**0**	0.7445	**0**	0.7448	0.6926	0.6976
WDBC	0.9050	0.9367	0.9402	0.9402	0.9508	0.9420	0.9420	0.9351	0.9315
WPBC	0.6963	0.7413	0.7024	**0**	0.7121	0.6805	0.6855	0.6955	0.6924
Wine	0.8986	0.9090	0.9035	0.9208	0.9153	0.9208	0.9437	0.8972	0.8972
Aver.	0.8098	0.8051	0.8194	--	0.8285	--	0.8258	--	0.8174

We can also find from table 4 that most of classification accuracies of reduced data decrease a little compared with the original data. Correspondingly, the average classification accuracies for all of the four discretization algorithms are a little lower than the original data. This shows that both dependency and consistency based feature selection algorithms are not fit for SVM learning because both dependency and consistency compute the distinguishing power in discrete spaces.

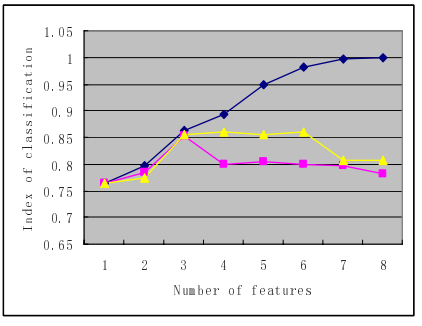
Table 5 shows the selected features based on consistency algorithm and the corresponding turns being selected for parts of the data, where we use the FCM discretized data sets. The trends of consistency and classification accuracies with

Table 4. Classification accuracy with 10-fold cross validation (SVM)

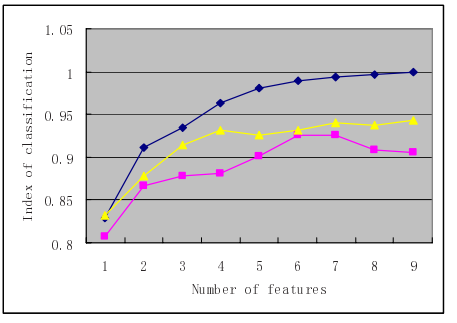
	Raw	Equal-width		Equal-frequency		Entropy		FCM	
	data	P	C	P	C	P	C	Positive	C
Crd	0.8144	0.8144	0.8144	0.8028	0.82729	0.8100	0.8275	0.8058	0.8058
Ecoli	0.8512	0.8512	0.8512	0.8512	0.8512	**0.4262**	0.8512	**0.4262**	0.8512
Heart	0.8111	0.8074	0.8074	**0**	0.8111	**0**	0.8111	**0.0000**	0.8074
Iono	0.9379	**0.7499**	0.9320	0.7499	0.9320	0.9154	0.9207	0.9348	0.9435
Sonar	0.8510	0.7398	0.7595	**0**	0.7300	**0**	0.8229	0.7074	0.7843
WDBC	0.9808	0.9668	0.9650	0.9597	0.9684	0.9561	0.9649	0.9649	0.9632
WPBC	0.7779	0.7737	0.7684	**0**	0.7737	0.7632	0.7632	0.7837	0.7632
Wine	0.9889	0.9444	0.9701	0.9660	0.9660	0.9722	0.9556	0.9486	0.9486
Aver.	0.8767	0.8309	0.8585	--	0.8575	--	0.8646	--	0.8584

Table 5. The selected Features with method FCM + Consistency

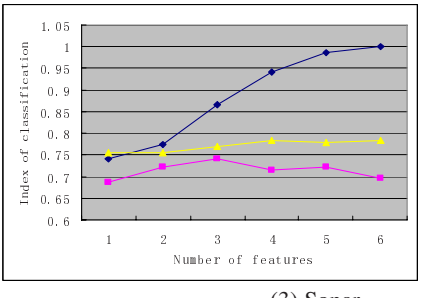
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Heart	13	12	3	10	1	4	7	5		
Iono	5	6	8	21	9	3	10	7	28	
Sonar	11	16	37	3	9	33				
WDBC	28	21	22	3	7	14	15	2	4	6
WPBC	25	33	1	7	23	18	6			



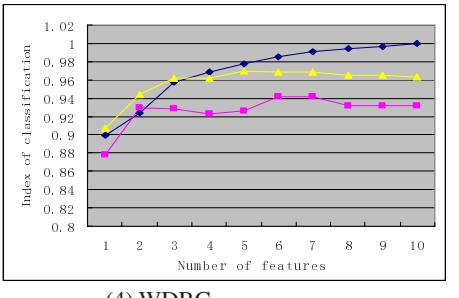
(1)Heart



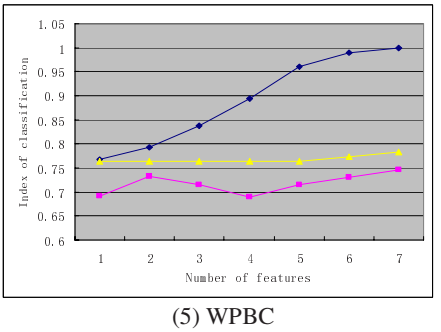
(2) Iono



(3) Sonar



(4) WDBC



(5) WPBC

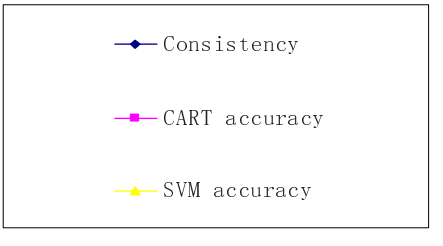


Fig. 4. Trends of consistency, accuracies with CART and SVM

CART and SVM are shown in figure 4. As to all of the five plots, the consistency monotonously increases with the number of selected attributes. The maximal value of consistency is 1, which shows that the corresponding decision table is consistent. With the selected attributes, all of the samples can be distinguished. What's more, it is

noticeable that the consistency rapidly rises at the beginning; and then slowly increases, until stops at 1. It means that the majority of samples can be distinguished with a few features, while the rest of the selected features are introduced to discern several samples. This maybe leads to the over-fitting problem. Therefore the algorithm should be ceased earlier or we need a pruning algorithm to delete the over-fitting features. The classification accuracy curves also show this problem. In figure 4, the accuracies with CART and SVM rise at first, arrive at a peak, then keep unchanged, or even decrease. In terms of classification learning, it shows the features after the peak are useless. They sometimes even deteriorate learning performance.

Here we can take two measures to overcome the problem. The first one is to stop the algorithm when the increment of consistency is less than a given threshold. The second one is to employ some learning algorithm to validate the selected features, and delete the features after the accuracy peak. However, sometimes the first one, called prepruning method, is not feasible because we usually cannot exactly predict where the algorithm should stop. The latter, called post-pruning, is widely employed. In this work, cross validation are introduced to test the selected features. Table 6 shows the numbers of selected features and corresponding classification accuracies. We can find that the classification performance improves in most of the cases. At the same time, the selected features with consistency are further reduced. Especially for data sets Heart and Iono, the improvement is high to 10% and 18% with CART algorithm.

Table 6. Comparison of features and classification performance with post-pruning

	Raw data			CART		SVM	
	features	CART	SVM	features	Accuracy	features	Accuracy
Heart	13	0.7630	0.8111	3	0.8519	4	0.8593
Iono	34	0.7499	0.9379	6	0.9260	9	0.9435
Sonar	60	0.7024	0.8510	3	0.7407	6	0.7843
WDBC	30	0.9367	0.9808	6	0.9420	5	0.9702
WPBC	33	0.7413	0.7779	7	0.7461	7	0.7837

5 Conclusions

In this paper, we introduce consistency function to overcome the problems in dependency based algorithms. We discuss the relationship between dependency and consistency, and analyze the properties of consistency. With the measure, the redundancy and reduct are redefined. We construct a forward greedy attribute reduction algorithm based on consistency. The numerical experiments show the proposed method is effective. Some conclusions are shown as follows.

Compared with dependency, consistency can reflect not only the size of decision positive region, but also the sample distribution in boundary region. Therefore, the consistency measure is able to describe the distinguishing power of an attribute set more finely than the dependency function.

Consistency is monotonous. The consistency value increases or keeps when a new attribute is added into the attribute set. What's more, some attributes are introduced into the reduct just for distinguishing a few samples. If we keep these attributes in the final result, the attributes maybe overfit the data. Therefore, a pruning technique is

required. We use 10-fold cross validation to test the results in the experiments and find more effective and efficient feature subsets.

Reference

1. Bhatt R. B., Gopal M.: On fuzzy-rough sets approach to feature selection. Pattern Recognition Letters 26 (2005) 965–975.
2. Breiman L., Freidman J., Olshen R., Stone C.: Classification and regression trees. California: Wadsworth International. 1984.
3. Dash M., Liu H.: Consistency-based search in feature selection. Artificial Intelligence 151 (2003) 155-176.
4. Guyon I., Weston J., Barnhill S., et al.: Gene selection for cancer classification using support vector machines. Machine Learning. 46 (2002) 389-422.
5. Guyon I., Elisseeff A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3 (2003) 1157-1182.
6. Hu Q. H., Li X. D., Yu D. R.: Analysis on Classification Performance of Rough Set Based Reducts. Q. Yang and G. Webb (Eds.): PRICAI 2006, LNAI 4099, pp. 423 – 433, 2006. Springer-Verlag Berlin Heidelberg.
7. Hu Q. H., Yu D. R., Xie Z. X.: Information-preserving hybrid data reduction based on fuzzy-rough techniques. Pattern Recognition Letters 27 (2006) 414-423.
8. Jensen R., Shen Q.: Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches. IEEE transactions of knowledge and data engineering 16 (2004) 1457-1471.
9. Liu H., Yu L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on knowledge and data engineering. 17 (2005) 491-502.
10. Quinlan J. R.: Induction of decision trees. Machine Learning 1 (1986) 81–106.
11. Skowron A., Rauszer C.: The Discernibility Matrices and Functions in Information Systems. Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory, Slowinski R (ed.), 1991, pp.331-362.
12. Slezak D.: 2001. Approximate decision reducts. Ph.D. Thesis, Warsaw University.
13. Ślezak D.: Approximate Entropy Reducts. Fundamenta Informaticae 53 (2002) 365– 390.
14. Swiniarski R. W., Skowron A.: Rough set methods in feature selection and recognition. Pattern recognition letters 24 (2003) 833-849.
15. Xie Z. X., Hu Q. H., Yu D. R.: Improved feature selection algorithm based on SVM and correlation. Lecture notes in computer science 3971(2006) 1373-1380.
16. Zhong N., Dong,J., Ohsuga S.: Using rough sets with heuristics for feature selection. J. Intelligent Information Systems 16 (2001) 199-214.
17. Ziarko W.: Variable precision rough sets model. Journal of Computer and System Sciences 46 (1993) 39-59.

A Hybrid Command Sequence Model for Anomaly Detection

Zhou Jian, Haruhiko Shirai, Isamu Takahashi, Jousuke Kuroiwa, Tomohiro Odaka, and Hisakazu Ogura

Graduate School of Engineering, University of Fukui, Fukui-shi, 910-8507, Japan
{jimzhou, shirai, takahasi, jousuke, odaka, ogura}@rook.i.his.fukui-u.ac.jp

Abstract. A new anomaly detection method based on models of user behavior at the command level is proposed as an intrusion detection technique. The hybrid command sequence (HCS) model is trained from historical session data by a genetic algorithm, and then it is used as the criterion in verifying observed behavior. The proposed model considers the occurrence of multiple command sequence fragments in a single session, so that it could recognize non-sequential patterns. Experiment results demonstrate an anomaly detection rate of higher than 90%, comparable to other statistical methods and 10% higher than the original command sequence model.

Keywords: Computer security; IDS; Anomaly detection; User model; GA; Command sequence.

1 Introduction

Preventative methods are widely used to safeguard against access to restricted computing resources, including techniques such as accounts, passwords, smart cards, and biometrics to provide access control and authentication [1]. However, with growing volume and sensitivity of data processed on computer systems, data security has become a serious consideration, making it necessary to implement secondary defenses such as intrusion detection [2]. Once intruders have breached the authentication level, typically using the system under a valid account, online intrusion detection is used as a second line of defense to improve the security of computer systems. Intrusion detection systems (IDS) have been studied extensively in recent years with the target of automatically monitoring behavior that violates the security policy of the computer system [3] [4] [5].

The present study focuses on anomaly detection at the command line level in a UNIX environment. Each user in a homogeneous working environment has specific characteristics of input that depending on the task, such as familiar commands and usage habits, and the topic of work will be stable within discrete periods. Users also differ individually in terms of work content and access privileges. For example, a programmer and a secretary may exhibit very different usage behaviors. One means of intrusion detection is therefore to construct a user

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 108–118, 2007.

model by extracting characteristics of user behavior from historical usage data and detecting out any variation from this typical usage pattern as a potential intruder.

The present authors have already conducted some research on anomaly detection at the command level using such a user model [7]. Detection was realized by a simple command sequence (SCS) model method, in which the user model was built by extracting command sequence fragments frequently used by the current user and seldom used by others. The model was trained by machine learning with a genetic algorithm (GA), and the method successfully detected more than 80% of anomalous user sessions in the experiment.

In this paper, a new hybrid command sequence (HCS) model is presented. The characteristics of user behavior are extracted by machine learning, and a list of unique and frequently used command combinations are built for each user. The trained HCS model can then be used as the criteria on detecting illegal user behavior (breach of authentication) or anomalous internal user behavior (misuse of privileges). These improvements provide a substantial increase in performance over the original SCS method, and it also shows comparable to other statistical methods based on the experiment of a common command sequence data set.

2 Related Work

Intruder detection systems are broadly based on two ways: anomaly detection and misuse detection. Anomaly detection is based on the assumption that on a computer system, the activity during an attack will be noticeable different from normal system activity. Misuse detection is realized by matching observed behavior with that in a knowledge base of known intrusion signatures [3]. Each of these techniques has weaknesses and strengths. Anomaly detection is sensitive to behavior that varies from historical user behavior and thus can detect some new unknown intrusion techniques, yet also often judges new but normal behavior as illegal. In contrast, misuse detection is not sensitive to unknown intruder techniques but provides a low false alarm rate.

Anomaly detection using Unix shell commands has been extensively studied. In addition to providing a feasible approach to the security of Unix systems, it's also possible to be generalized to other systems. Schonlau et al. [6] summed up six methods of anomaly detection at command line level: "Uniqueness", "Bayes one-step Markov", "Hybrid multi-step Markov", "Compression", "IPAM" and "Sequence-match". The Bayes one-step Markov method is based on one-step transitions from one command to the next. The detector determines whether the observed transition probabilities are consistent with the historical transition probabilities. The hybrid multi-step Markov method is based mostly on a high-order Markov chain and occasionally on an independence model depending on the proportion of commands in the test data that were not observed in the training data. The compression method involves constructing a reversible mapping of data to a different representation that uses fewer bytes, where new data from a given user compresses to about the same ratio as old data from the same user. The incremental probabilistic action modeling (IPAM) method is based on one-step command transition probabilities estimated from the training data with a continually exponential updating scheme. The sequence-matching method computes a similarity measure between the 10 most recent commands and a user's profile using a

command-by-command comparison of the two command sequences. The uniqueness method is based on the command frequency. Commands not seen in the training data may indicate a masquerade attempt, and the more infrequently a command is used by the user, the more indicative that the command is being used in a masquerade attack.

These approaches conducted anomaly detection in a statistical way, where deviation of system running state was monitored with a statistical value, and a threshold was used as the classify standard. However, the characteristic based classification is another way to anomaly detection. That is, it should be possible to use unique command combinations specific to each user to verify user behavior and define security policies. The SCS model was proposed by the present authors [7] as such an approach, in which the characteristic user model was constructed by extracting frequently appearing command sequence fragments for each user and applying GA-based machine learning to train the model. In this paper, the HCS model is presented as an extension of the SCS model to account for additional characteristics of user behavior. GA programming is also employed for machine learning the model from historical profile data.

3 Hybrid Command Sequence Model

3.1 Hybrid Command Sequence Model

The SCS model is constructed from the historical session profile data of individual users, where session is defined as the command sequences inputted between login and logout. A session is regarded as a basic analysis unit for user behavior, which involves activities conducted to achieve a certain missions. By analyzing user behavior in discrete periods with similar task processing, the unique behavioral characteristics for each user can be extracted. The SCS model was thus constructed to characterize user behavior, and the trained model was used to label unknown sessions. The model is trained by a GA method from historical session profile data to search command sequence fragments that frequently appeared in the current user's normal session data set $\{St\}$, but which occurred only rarely in the data sets of other users $\{Sf\}$. For each user, the number of commands in one command sequence (CS) fragment and the number of CS fragments in the learned SCS model are all variable according to the training process. For an unknown session, if it contains either CS fragment of the SCS model, it'll be labeled as legal input, otherwise as illegal. The matching between CS fragments of the SCS model and the observed session data is illustrated in Fig. 1, where the CS fragment (CS3) contains three commands C31, C32 and C33. If the observed session contains these three commands sequentially, regardless of their location in the session, it is labeled as legal. Experiment results showed that the SCS model is capable of up to 80% accuracy in detection of illegal sessions.

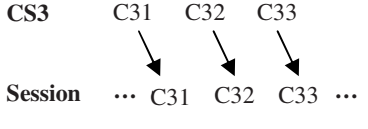


Fig. 1. Matching between a CS fragment of a SCS model and the observed session data

However, the SCS model could not fit to the situation that frequently used command combinations (combination of CS fragments) may occur in one session but not necessarily in sequential order. For multiple CS fragments in a single session may be more powerful in characterizing user behavior, the hybrid command sequence (HCS) model is therefore proposed. The HCS model, as an extension of the SCS model, could describe further characteristics such as multiple CS fragments and discrete commands in a single session.

Fig. 2 shows the structure of the HCS model. The model is constructed by multiple units, and each unit contains multiple CS fragments. The number of units in one model, the number of CS fragments in one unit, and the number of commands (denoted C in Fig. 2) in one CS fragment are all variable depending on training. An example of the HCS model is shown in Fig. 3. The different numbers of unit, CS fragment and command in the HCS model could describe different characteristics of user behavior. For example, if each unit contains only one CS fragment, the HCS model is identical to the SCS model; while if each CS fragment contains only one command, these discrete commands in the unit are used as the criteria for anomaly detection without consideration of the sequential characteristic. In the case that one unit contains both continuous sequence and discrete commands, the model is a composite of these two characteristics. The HCS model is thus a more powerful model in representing characteristics of user behavior.

When the HCS model is used to detect an unlabeled session, the session is labeled as legal if either unit of the model is found in the session, which means that all CS fragments of the unit must be contained by the session. An illustration of the matching between a unit of the HCS model and a session is shown in Fig. 4, where a unit consists of two CS fragments (CS11, CS12), the CS fragment CS11 contains three commands (C111, C112 and C113), and CS12 contains two commands (C121 and C122). Thus, if the session contains both CS11 and CS12, the session is labeled as legal.

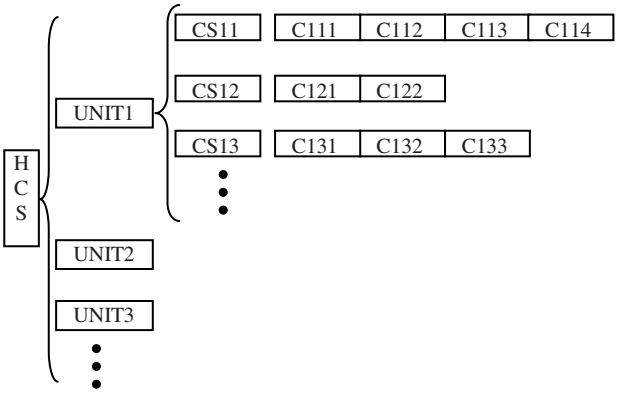


Fig. 2. Structure of the hybrid command sequence model

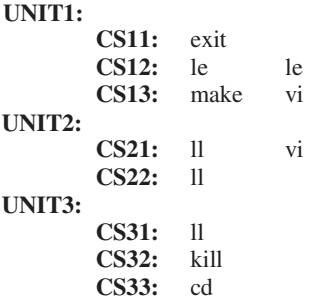


Fig. 3. Example of an HCS model

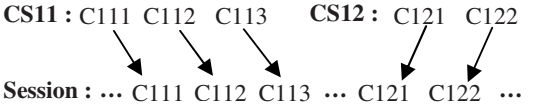


Fig. 4. Matching between a unit in the HCS model and the observed session data

3.2 Definition

Judgment of the legality of a session is a binary classification problem. The verification of *T* unlabeled sessions, with *TP* as correct acceptance classifications, *TN* as correct alarms, *FP* as false acceptances classifications, and *FN* as false alarms ($T = TP + TN + FP + FN$), is therefore given by

$$FRR = \frac{FN}{TP + FN}. \tag{1}$$

$$FAR = \frac{FP}{TN + FP}. \tag{2}$$

where *FRR* is the false rejection rate (incorrect classification of normal data as hostile), *FAR* is the false acceptance rate (incorrect classification of hostile data as normal). The quality of a detector can then be judged in terms of its ability to minimize either *FRR* or *FAR*. In reality, there is often a bias favoring on either *FRR* or *FAR*. Therefore, the overall quality of the method can be evaluated by a cost function as follows.

$$Cost = \alpha \times FRR + \beta \times FAR. \tag{3}$$

As the cost of a false alarm and a miss alarm will vary to the application, there is no way to set relative values of α and β to achieve an optimal cost in all cases. By convention, α and β in a cost function are both set to 1, given the relation

$$Cost = FRR + FAR. \tag{4}$$

3.3 Machine Leaning of HCS by GA

The training of a model with historical user data is a search problem to find specific and differential behavioral characteristics for a particular user. It's impossible to search such complex command patterns of the HCS model directly from the large data space. GA is a relatively efficient approach for searching in a large data space. In the learning stage by GA, the optimization target is to find the model that occurs frequently in the target user's normal data set $\{St\}$ and seldom in the data of other users $\{Sf\}$. The two-stage GA is employed. Encoding and implementation are described as below.

3.3.1 GA Encoding

Training of the HCS model is performed to find the optimal combinations of commands. A command table of frequently appearing commands is constructed initially, and each command is indexed by a number value. The search operation is conducted based on frequently appeared commands rather than all commands to improve searching efficiency. Each chromosome in GA has a structure similar to that of the HCS model (Fig. 2), so that each HCS model is encoded as a chromosome. Rather than binary encoding, each gene in a chromosome is encoded using the indices in the command table. In the decoding stage, each chromosome is decoded as a solution of the HCS model which contains multi-command combinations. One solution of the HCS model appears as shown in Fig. 3. The numbers of unit, CS fragment and command are all variable depending upon the initialization and the evolutionary operation of the GA.

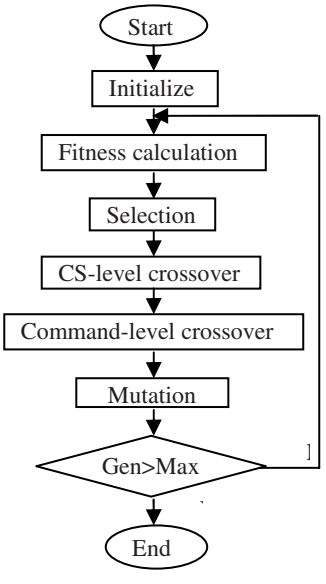


Fig. 5. GA processing

3.3.2 GA Processing

GA processing typically involves an initialization stage and an evolution stage, which includes fitness calculation, selection, crossover and mutation, as shown in Fig. 5. When initializing a population, numbers of unit, CS fragment and command are all set randomly. Here, the CS fragments are initialized with randomly selected CS fragments extracted from the user's sessions. The optimization target of the GA is to gain the minimum cost function (Eq. (3) or (4)). Therefore, the fitness function is defines as:

$$Fitness = 2 - (\alpha \times FRR + \beta \times FAR). \tag{5}$$

where $\alpha + \beta = 2$, and α and β may vary according to different applications. By convention, α and β are set to 1 in this paper, leading to the relation

$$Fitness = 2 - (FRR + FAR). \tag{6}$$

Selection is processed under the proportional selection strategy according to individuals' fitness.

Crossover, a key process of evolution in GA, is performed in a special way to account for the variability of the numbers of unit, CS fragment and command in a chromosome. Here, the two stage crossover is employed: CS-level crossover and command-level crossover, where the former provides stability and ensures evolution of the population, and the latter allows evolution of the number of commands. In CS-level crossover, a randomly chosen point is assigned as a crossover point for a pair of mated individuals. As the example shown in Fig. 6, the crossover point is 4. For command-level crossover operations, a pair of CS fragments is chosen randomly according to a probability (set at 0.1 here based on experiences) from a pair of mated individuals. Crossover points are then selected randomly for the two CS fragments separately. As the example shown in Fig. 7, the crossover point of CS33 is 2, and CS33' is 3. Result of the crossover operation is shown in Fig. 8.

Mutation is realized by randomly choosing one Gene in a chromosome according to a probability, and setting the point to a randomly selected value from the command table index.

CS11	CS12	CS13	CS21	**CS22'**	**CS23'**	**…**

CS11'	CS12'	CS13'	CS21'	**CS22**	**CS23**	**…**

Fig. 6. CS-level crossover of two individuals at point 4

CS33	C1	C2	**C3**	**C4**	**C5**

CS33'	C1'	C2'	C3'	**C4'**

Fig. 7. Selection of two CS fragments for command-level crossover

CS33	C1	C2	**C4'**

CS33'	C1'	C2'	C3'	**C3**	**C4**	**C5**

Fig. 8. Result of command-level crossover operation in Fig. 7

4 Experiments

The HCS model method was evaluated on the same "session data" as the SCS model, where session is regard as analysis unit. At same time, to compare the HCS model with other previous methods, experiments were also conducted on the common data set provided by Schonlau et al. [6]. However, the Schonlau data is a simply collection of 15000 commands for each user, without labeling the session each command belongs to. Thus, to apply the HCS model to the Schonlau data, the command set of each user is divided manually into 150 sessions with 100 commands in each session.

4.1 The Session Data

The session data set consists of historical session profile data for 13 users collected over 3 months. The users were graduate computer science students who used the computer system for programming, document editing, email, and other regular activities. Only sessions containing more than 50 commands were recorded, command arguments were excluded. A total of 1515 sessions were collected, including 83694 commands. For seven users, 529 sessions were used as the training data, and the remaining 521 sessions were used as the testing data. The 465 sessions for the other six users were used as the independent testing data [7].

The results are listed in Tables 1–3. *FRR* and *FAR* of the HCS model for the training, testing and independent data exhibits a remarkable 10% improvement compared to that achieved by the SCS method [7]. The average *FRR* for the testing data set is higher than the average *FAR* for the testing and independent data, and it shows that the HCS model is relatively more powerful for anomaly detection, but suffers from a slightly elevated *FRR*. The average *FRR* for the testing data set is also much higher than the average *FRR* for the training data, and it shows that there has some degradation when the trained model is applied to the test data. The *FAR*

Table 1. *FRR* and *FAR* for the training data (%)

Subject	*FRR*	*FAR*
User 1	3.2	6.5
User 2	10.4	0.5
User 3	14.3	3.6
User 4	16.9	19.0
User 5	14.7	0.0
User 6	1.2	1.1
User 7	3.8	1.1
Average	9.2	4.5

Table 2. *FRR* and *FAR* for the testing data (%)

Subject	*FRR*	*FAR*
User 1	5.7	6.5
User 2	26.0	0.6
User 3	31.6	3.6
User 4	19.7	25.7
User 5	41.5	3.2
User 6	18.6	3.6
User 7	17.9	1.0
Average	23.0	6.3

Table 3. *FAR* for the independent data (%)

Subject	*FAR*
User 8	6.9
User 9	0.0
User 10	19.3
User 11	12.4
User 12	0.0
User 13	17.2
Average	9.3

value for user 5 is 14.7% for the training data and 41.5% for the testing data. This therefore indicates that a certain users as user 5 are difficult to describe uniquely by the HCS model.

4.2 The Schonlau Data

Schonlau et al. collected command line data for 50 users, with 15000 commands in the set (file) for each user. The first 5000 commands do not contain any masqueraders and are intended as the training data, and the remaining 10000 commands (100 blocks of 100 commands) are seeded with masquerading users (i.e. with data of another user not among the 50 users). A masquerade starts with a probability of 1% in any given block after the initial 5000 commands, and masquerades have an 80% chance of continuing for two consecutive blocks. Approximately 5% of the testing data are associated with masquerades [6].

The time cost for the collection of data in the Schonlau set differs for each user. To adopt the session notation, the training data are divided manually into 100 sessions of 100 commands. Although this will result in some degradation of the performance of the HCS method, which is built on the notion of session, the results are still comparable with other methods. The first 5000 commands of each user are divided into 50 sessions as the training data. The 50 sessions of the current user are used as the legal training data, and 1000 sessions of other 20 randomly selected users are used as the illegal training data. Experiment results of previous methods and the HCS method based on the Schonlau data are shown on Table 4, and *Cost* is calculated according to Eq. 4. We could see efficiency of HCS is better than others, and it gains the best *Cost* value with a relative high *FRR* of 33.9%.

Table 4. Results based on the Schonlau data (%)

Method	*FAR*	*FRR*	*Cost*
1-step Markov	30.7	6.7	37.4
Hybrid Markov	50.7	3.2	53.9
IPAM	58.6	2.7	61.3
Uniqueness	60.6	1.4	62.0
Sequence Matching	63.2	3.7	66.9
Compression	65.8	5.0	70.8
HCS	1.4	33.9	35.3

5 Discussion

The HCS model is an extension of the SCS model. Besides the sequential characteristic as the SCS model, it could also describe the co-existence characteristic. As searching such complex command patterns directly from the large data space is impossible, GA, which is a relatively efficient approach for searching in a large data space, is employed for learning the HCS model. As a result, the HCS model method exhibited a 10% improvement in anomaly detection compared to that of the SCS model method.

Different from previous methods, which took all commands into account in a statistical way, the HCS model method only depends on the usages of typical command combinations owned by individual users. Therefore, processes of anomaly detection by the HCS model is more directly and interpretable than that of statistical methods. Additionally, for anomaly detection by the HCS method only needs matching operation between input commands and commands patterns of the model, it needs less computation cost than other methods.

The performance of the HCS method based on the Schonlau data is also comparable to that of the other six methods summarized by Schonlau et al. As while, the performance of the HCS method is somewhat lower on the Schonlau data than on the session data. This is partly due to the manual division of the data set into sessions, which destroys the structure of the data set. The Schonlau data was collected without the consideration of session or period. A long period of data collection may also cause a substantial shift in user behavior, reducing the performance of the HCS method. To adapt to variation of user behavior over time, the HCS model should be further maintained periodically to ensure its efficiency.

In both experiments, the *FRR* is much higher than the *FAR*. It rests on that: in the HCS model, only typical characteristics of command combination used by the current user are searched and employed to verify normal behavior of this user. For the typical characteristics of the current user are rarely used by anomaly sessions inputted by other user, it gained a high efficiency in anomaly detection. At same time, for only limited typical characteristics are extracted to describe behaviors of the current user, the normal behavior, which doesn't necessarily contain typical usages, will be apt to be judged as abnormal and leads to a high false alarm rate. That's the reason why the *FRR* is difficult to decrease even using different parameters of α and β in Eq. (5).

The start point of the HCS is different from other statistical methods. As in the Uniqueness method, commands not seen before or infrequently used are extracted and

used to label anomaly behaviors. Yet, the HCS method extracts command patterns that frequently used by the current user and seldom used by others to discriminate normal behaviors. Therefore, the two methods are complimentary in using characteristics of user behavior, and the composition of the two methods is expected to be researched in future to gain a better result.

6 Conclusions

The hybrid command sequence model (HCS) was proposed as the basis for a new intrusion detection method. The HCS model is constructed by extracting characteristics of user behavior (combinations of command sequence fragments) from historical session data by GA. When applying the learned model to the unknown session data, the HCS model method achieves a false acceptance rate of lower than 10% for anomaly detection, and provides a 10% improvement in efficiency than the previous the SCS model. The HCS method also performs comparable to other statistical techniques, even though the method approaches the problem from a different starting point. When combined with other preventative methods such as access control and authentication, the HCS method is expected to improve the security of a computer system significantly.

The HCS model also has the advantage of low computation cost in anomaly detection, requiring only a matching operation between a set of limited command combinations of the HCS model and the observed session.

For the present system still has a high false rejection rate (>20%), future work should be done to improve efficiency further more. It may include taking a wider range of characteristics of user behavior into account, and composing the HCS method with other statistical techniques. Additionally, it should also be evaluated on other context, such as fraud detection, mobile phone based anomaly detection of user behavior, and so on.

References

1. Kim, H.J.: Biometrics, Is It a Viable Proposition for Identity Authentication and Access Control? Computers & Security, vol. 14 (1995), no. 3, pp. 205-214
2. Computer Security Institute: CSI/FBI Computer Crime and Security Survey Results Quantify Financial Losses. Computer Security Alert (1998), no. 181
3. Biermann, E., Colete, E., and Venter, L.M.: A Comparison of Intrusion Detection Systems. Computers & Security, vol. 20 (2001), no. 8, pp. 676-783
4. Axelsson S.: Intrusion Detection Systems: A Survey and Taxonomy. Technical Report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden
5. Murali, A., Rao, M.: A Survey on Intrusion Detection Approaches. Proc. of ICICT 2005, pp. 233-240
6. Schonlau, M., DuMouchel, W., Ju, W., Karr, A., Theus, M., and Vardi, Y.: Computer Intrusion: Detecting Masquerades. Statistical Science, vol. 16 (2001), no. 1, pp. 58-74
7. Odaka, T., Shirai, H., and Ogura, H.: An Authentication Method Based on the Characteristics of the Command Sequence. IEICE, vol. J85-D-I (2002), no. 5, pp. 476-478

σ-Algorithm: Structured Workflow Process Mining Through Amalgamating Temporal Workcases

Kwanghoon Kim[1] and Clarence A. Ellis[2]

[1] Collaboration Technology Research Lab
Department of Computer Science
KYONGGI UNIVERSITY
San 94-6 Yiui-dong Youngtong-ku Suwon-si Kyonggi-do, 442-760, South Korea
kwang@kyonggi.ac.kr
http://ctrl.kyonggi.ac.kr
[2] Collaboration Technology Research Group
Department of Computer Science
UNIVERSITY OF COLORADO AT BOULDER
Campus Box 430, Boulder, Colorado, 80309-0430, USA
skip@cs.colorado.edu

Abstract. Workflow Management Systems help to execute, monitor and manage work process flow and execution. These systems, as they are executing, keep a record of who does what and when (e.g. log of events). The activity of using computer software to examine these records, and deriving various structural data results is called workflow mining. The workflow mining activity, in general, needs to encompass behavioral (process/control-flow), social, informational (data-flow), and organizational perspectives; as well as other perspectives, because workflow systems are "people systems" that must be designed, deployed, and understood within their social and organizational contexts. In this paper[1], we especially focus on the behavioral perspective of a structured workflow model that preserves the proper nesting and the matched pair properties. That is, this paper proposes an ICN-based mining algorithm that rediscovers a structured workflow process model. We name it *σ-Algorithm*, because it is incrementally amalgamating a series of temporal workcases (workflow traces) according to three types of basic merging principles conceived in this paper. Where, a temporal workcase is a temporally ordered set of activity execution event logs. We also gives an example to show that how the algorithm works with the temporal workcases.

Keywords: Workflow Management System, Events Log, Workflow Mining, Process Rediscovery, Temporal Workcase, Workflow Process Mining Framework.

[1] This research was supported by the Kyonggi University Overseas Research Grant 2004.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 119–130, 2007.

1 Introduction

A Workflow Management System (WfMS) is defined as a system that (partially) automates the definition, creation, execution, and management of work processes through the use of software that is able to interpret the process definition, interact with workflow participants, and invoke the use of IT tools and applications. Steps of a work process are called activities, and jobs or transactions that flow through the system are called workcases or workflow instances. Such a WfMS and its related technologies have been constantly deployed and so gradually hot-issued in the IT arena. This atmosphere booming workflows modeling and reengineering is becoming a catalyst for triggering emergence of the concept of workflow mining that rediscovers several perspectives—control flow, data flow, social, and organizational perspectives—of workflows from workflow execution histories collected at runtime.

In this paper, we especially focus on the control flow perspective of the workflow mining functionality. A workflow model is described by several entities, such as activity, role, actor, invoked applications, and relevant data. The control flow perspective specifies the transition precedences—sequential, conjunctive(AND) and disjunctive(OR) execution sequences—among the activities, and it is represented by the concept of workflow process model defined in this paper by using the graphical and formal notations of the information control net (ICN)[10]. Also, we assume that the workflow process model keeps the proper nesting and the matched pairing properties in modeling the conjunctive and the disjunctive transitions—AND-split, AND-join nodes and OR-split, OR-join nodes—in order to compose a structured workflow model[16]. Based upon the concept of the structured workflow process model, we propose a workflow process mining algorithm that is a means of rediscovering a structured workflow process model from log of activity execution events. A workflow event log is typically an interleaved list of events from numerous workcases—workflow instances.

In the remainder of this paper, we are going to show that our mining algorithm is able to handle all of the possible activity execution cases through the concepts of temporal workcase. At first, the next section presents the meta-model of the structured workflow process model with graphical and formal notations. In the main sections of this paper, we present a workflow process mining framework and the detailed description of the workflow process mining algorithm with respect to its basic principles and constructs with some examples. Finally, we discuss the constraints of the workflow process mining algorithm and its related work.

2 Structured Workflow Process Model

In this paper, we use the information control net methodology[10] to represent wokflow models. The information control net (ICN) was originally developed in order to describe and analyze information flow by capturing several entities within office procedures, such as activities, roles, actors, precedence, applications, and repositories. It has been used within actual as well as hypothetical

automated offices to yield a comprehensive description of activities, to test the underlying office description for certain flaws and inconsistencies, to quantify certain aspects of office information flow, and to suggest possible office restructuring permutations. In this section, especially, we focus on the activities and their related information flows by defining structured workflow process model through its graphical and formal representations.

2.1 Graphical Representation

As shown in Fig. 1, a workflow process model consists of a set of activities connected by temporal orderings called activity transitions. In other word, it is a predefined set of work steps, called activities, and a partial ordering (or control flow) of these activities. Activities can be related to each other by combining sequential transition types, disjunctive transition types (after activity α_A, do activity α_B or α_C, alternatively) with predicates attached, and conjunctive transition types (after activity α_A, do activities α_B and α_C concurrently). An activity is either a compound activity containing another subprocess, or a basic unit of work called an elementary activity. An elementary activity can be executed in one of three modes: manual, automatic, or hybrid.

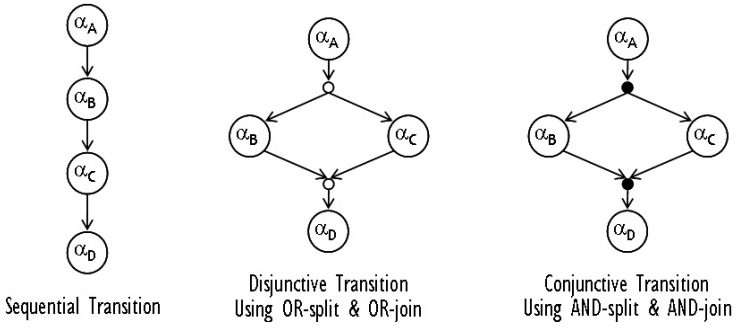


Fig. 1. Graphical Notations for the Basic Transition Types

2.2 Formal Representation

The structured workflow process model needs to be represented by a formal notation that provides a means to eventually specify the model in textual language or in database, and both. The following definition is the formal representation of the structured workflow process model:

Definition 1. Structured Workflow Process Model(SWPM). *A basic structured workflow process model is formally defined through 4-tuple* $\Gamma = (\delta, \kappa, \boldsymbol{I}, \boldsymbol{O})$ *over an activity set* $\boldsymbol{A}$*, and a transition-condition set* $\boldsymbol{T}$*, where*

- $\boldsymbol{I}$ *is a finite set of initial input repositories, assumed to be loaded with information by some external process before execution of the model;*

- $\boldsymbol{O}$ *is a finite set of final output repositories, which is containing information used by some external process after execution of the model;*
- $\delta = \delta_i \cup \delta_o$,
  where, $\delta_o : \boldsymbol{A} \longrightarrow \wp(\alpha \in \boldsymbol{A})$ *is a multi-valued mapping function of an activity to its set of (immediate) successors, and* $\delta_i : \boldsymbol{A} \longrightarrow \wp(\alpha \in \boldsymbol{A})$ *is a multi-valued mapping function of an activity to its set of (immediate) predecessors;*
- $\kappa = \kappa_i \cup \kappa_o$,
  where $\kappa_i : \boldsymbol{T} \longrightarrow \wp(\alpha \in \boldsymbol{A})$ *is a multi-valued mapping function of an activity to its incoming transition-conditions* $(\in \boldsymbol{T})$ *on each arc,* $(\delta_i(\alpha), \alpha)$*; and* $\kappa_o : \boldsymbol{T} \longrightarrow \wp(\alpha \in \boldsymbol{A})$ *: is a multi-valued mapping function of an activity to its outgoing transition-conditions* $(\in \boldsymbol{T})$ *on each arc,* $(\alpha, \delta_o(\alpha))$.

Summarily, the structured workflow process model will be constructed by Structured Modeling Methodology [16] preserving the proper nesting and the matched pairing properties, and its formal definition implies that the structured ordering of a workflow process model can be interpreted as the ordered combination of the following basic transition types graphically depicted in Fig. 1.

(1) Sequential Transition
incoming $\rightarrow \delta_i(\alpha_B) = \{\{\alpha_A\}\}$; *outgoing* $\rightarrow \delta_o(\alpha_B) = \{\{\alpha_C\}\}$;
(2) OR Transition
or-split $\rightarrow \delta_o(\alpha_A) = \{\{\alpha_B\}, \{\alpha_C\}\}$; *or-join* $\rightarrow \delta_i(\alpha_D) = \{\{\alpha_B\}, \{\alpha_C\}\}$;
(3) AND Transition
and-split $\rightarrow \delta_o(\alpha_A) = \{\{\alpha_B, \alpha_C\}\}$; *and-join* $\rightarrow \delta_i(\alpha_D) = \{\{\alpha_B, \alpha_C\}\}$;

3 Structured Workflow Process Mining Algorithm

In this section, we propose a workflow mining framework that eventually rediscovers a structured workflow process model from the workflow execution events log. The framework is made up of a series of concepts and algorithms. However, we particularly focus on the mining algorithm and its directly related concept—temporal workcase. Finally, in order to prove the correctness of the algorithm, we show how it works for a typical structured workflow process model, as example, comprising the three types of control flow transition.

3.1 Framework

The workflow process mining framework is illustrated in Fig. 2. The framework starts from the event logs written in XWELL (XML-based Workflow Event Log Language) [7], by which the workflow event logging mechanism of a workflow enactment engine stores all workflow process execution event histories triggered by the engine components. The XWELL needs to be standardized so that heterogeneous workflow mining systems are able to collect the event logs without any additional data transformations. In general, the event logs might be produced by the workflow engine's components, like *event triggering components*, *event formatting components* and *event logging components*. Once, a log agent receives

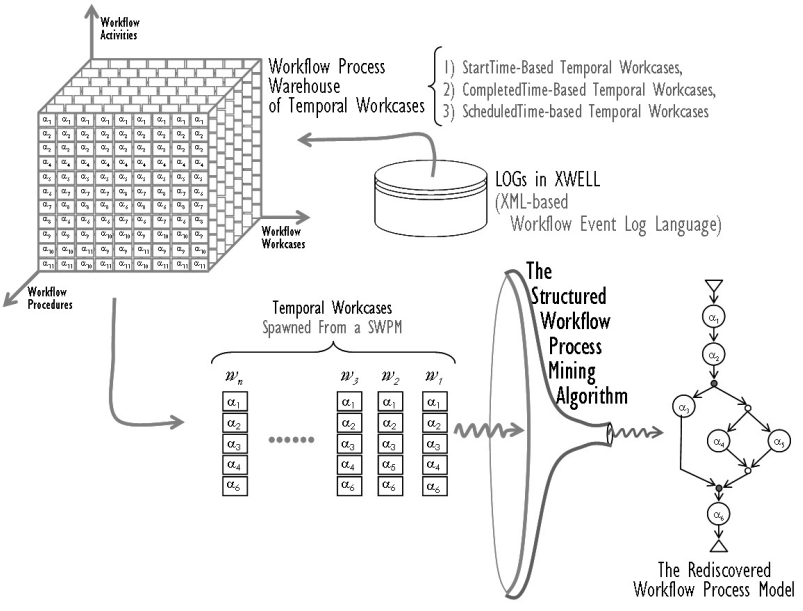


Fig. 2. The Workflow Process Mining Framework

event logs and then transforms them into XML-based log messages, and store the transformed messages onto the Log File Storage.

Based on the XML-based event logs on the log file storage, we can build a workflow process mining warehouse that shapes into a cube with three dimensions, such as workflow process models, temporal workcases, and activities. From the cube we extract a set of temporal workcases (traces) that is instantiated from a structured workflow process model. A temporal workcase is a temporal order of activity executions within an instance of the corresponding workflow process model, and it will be formally represented a workcase model. The details of the temporal workcase and its related models are precisely defined in the next section. Finally, the workflow process mining algorithm rediscovers a structured workflow process model by incrementally amalgamating a series of workcase models, $\omega_1 \sim \omega_n$, one-by-one. The details of the algorithm and its operational example are described in the next sections, too.

3.2 σ-Algorithm

This section gives a full detail of the workflow process mining algorithm and demonstrates the algorithm's correctness though an example of structured workflow process model. In order to mine a workflow process model if we use one of the existing algorithms[9], we have to assume that there might be many and possibly infinite workflow process models (if fake activities are allowed) that could be mined from a set of traces, even though some of these models are very easy to

compute, the others are not, and that we must pick one reasonable model out of the infinitely many models as a final output of the algorithm. However, we take a fundamentally different approach to conceive an algorithm. More specifically, our algorithm will build up one *reasonable* model by amalgamating one trace after another, each of which is embodied in a workcase model. In summary, the central idea of our approach is as follows:

- *The algorithm repeatedly modifies a temporarily rediscovered workflow process model, which is called reasonable model, by incorporating one at a time into it until running out all traces. Thus, it is an incremental algorithm; after seeing the first trace the algorithm generates a new reasonable model, and upon seeing the second trace it merges into the existing reasonable model, and so forth. Conclusively, the algorithm is made up of a series of rewrite operations that transform the reasonable model plus one trace into a new reasonable model until bringing up to the last. The final reasonable model becomes the structured workflow process model rediscovered from all traces of the corresponding workflow process model.*

3.2.1 Workflow Traces: Temporal Workcases

As a workflow process instance executes, a temporal execution sequence of its activities is produced and logged into a database or a file; this temporal execution sequence is called *workflow trace* or *temporal workcase*, which is formally defined in Definition 3. The temporal workcase is made up of a set of *workflow event logs* as defined in the following Definition 2. Also, we would define the concept of *workflow process log* in Definition 4, which is produced from a set of temporal workcases spawned from a single structured workflow process model.

Definition 2. Workflow Event Log. *Let* $\boldsymbol{we} = (\alpha, pc, wf, c, ac, \varepsilon, p^*, t, s)$ *be a workflow event, where* α *is a workitem (activity instance) number,* $\boldsymbol{pc}$ *is a package number,* $\boldsymbol{wf}$ *is a workflow process number,* $\boldsymbol{c}$ *is a workflow instance (case) number,* $\boldsymbol{ac}$ *is an activity number,* ε *is an event type, which is one of* {*Scheduled, Started, Changed, Completed*}, $\boldsymbol{p}$ *is a participant or performer,* $\boldsymbol{t}$ *is a timestamp, and* $\boldsymbol{s}$ *is an activity state, which is one of* {*Inactive, Active, Suspended, Completed, Terminated, Aborted*}*. Note that * indicates multiplicity.*

In general, we consider a workflow event log to be stored in an XML format. An XML based workflow event log language has been studied and proposed in [7] for the purpose of workflow mining. Because of the page length limitation, we now assume to simply use the language to describe the XML schema of a workflow event logs in this paper.

Definition 3. Workflow Trace (Temporal Workcase). *Let WT(*$\boldsymbol{c}$*) be the workflow trace of process instance* $\boldsymbol{c}$*, where* $WT(\boldsymbol{c}) = (we_1, \ldots, we_n)$*. Especially, the workflow trace is called temporal workcase, TW(*$\boldsymbol{c}$*), if all activities of its underlined process instance are successfully completed. There are three types of temporal workcases according to the events type—Scheduled, Started, Completed:*

- *ScheduledTime Temporal Workcase*
  $\{we_i | we_i.c = \boldsymbol{c} \wedge we_i.e = \textit{'Scheduled'} \wedge we_i.t \leq we_j.t \wedge i < j \wedge 1 \leq i, j \leq n\}$,*which is a temporally ordered workflow event sequence based upon the scheduled time stamp.*
- *StartedTime Temporal Workcase*
  $\{we_i | we_i.c = \boldsymbol{c} \wedge we_i.e = \textit{'Started'} \wedge we_i.t \leq we_j.t \wedge i < j \wedge 1 \leq i, j \leq n\}$, *which is a temporally ordered workflow event sequence based upon the stated time stamp.*
- *CompletedTime Temporal Workcase*
  $\{we_i | we_i.c = \boldsymbol{c} \wedge we_i.e = \textit{'Completed'} \wedge we_i.t \leq we_j.t \wedge i < j \wedge 1 \leq i, j \leq n\}$, *which is a temporally ordered workflow event sequence based upon the completed time stamp.*

As shown in the definition of temporal workcase, there are three types of temporal workcases differentiated from the temporal information (the event's timestamp) logged when the corresponding activity's workitem event was happened. Originally, in the workflow event log schema[7], the events that are associated with the workitem are related to `Scheduled`, `Started` and `Completed` in order to form the types of temporal workcases to be used in the workflow mining algorithm.

Definition 4. Workflow Process Log *and* Warehouse. *Let* $I_i = \{c_1^i, ..., c_m^i\}$ *be a set of completed process instances (*$\boldsymbol{m}$ *is the number of the process instances) that have been instantiated from a workflow process model,* I_i*. A workflow process warehouse consists of a set of workflow process logs,* $WL(I_1)$*, ...,* $WL(I_n)$*, where* $WL(I_i) = \forall WT(c^i \in I_i)$*, and* $\boldsymbol{n}$ *is the number of workflow process models managed in a system.*

Based on these defined concepts, we are able to prepare the temporal workcases that become the input data of the workflow mining algorithm proposed in this paper. Additionally, according to the types of temporal workcases, we can build three different types of workflow process logs and their warehouses as defined in Definition 4. Conclusively speaking, the workflow mining algorithm may consider taking the temporal workcases, as input data, coming from one of three workflow process warehouse types—ScheduledTime-based Warehouse, StartedTime-based Warehouse, and CompletedTime-based Warehouse. Also, the algorithm may simultaneously take two types of temporal information such as ScheduledTime/CompletedTime or StartedTime/CompletedTime to rediscover structured workflow process models. In this case, the algorithm needs to take two types of the temporal workcases, each of which is belonged to its warehouse type, respectively. The algorithm presented in this paper will be taking care of the StartedTime-based workflow process warehouse as the source of the temporal workcases. Nevertheless, it is sure for the algorithm to be able to be extended so as to handle two types of the temporal workcases as its input data.

3.2.2 Workcase Model

Each of the temporal workcases, as the input data of the algorithm, is represented into a workcase model through a series of converting operations of the algorithm. In the following Definition 5, we formally define the workcase model, and also it can be graphically represented, too, as shown in Fig. 3. The primary reason we use the formal workcase model is that because it is surely convenient in composing the workflow mining algorithm.

Definition 5. Workcase Model (WCM). *A workcase model is formally defined through 3-tuple* $\boldsymbol{W} = (\omega, \boldsymbol{P}, \boldsymbol{S})$ *over an activity set* $\boldsymbol{A}$*, where*

- $\boldsymbol{P}$ *is a predecessor activity of some external workcase model, which is connected into the current workcase model;*
- $\boldsymbol{S}$ *is a successor activity of some external workcase model, which is connected from the current workcase model;*
- $\omega = \omega_i \cup \omega_o$,
  where, $\omega_o : \boldsymbol{A} \longrightarrow \wp(\alpha \in \boldsymbol{A})$ *is a single-valued mapping function of an activity to its immediate successor in a temporal workcase, and* $\omega_i : \boldsymbol{A} \longrightarrow \wp(\alpha \in \boldsymbol{A})$ *is a single-valued mapping function of an activity to its immediate predecessor in a temporal workcase.*

3.2.3 The Basic Amalgamating Principles

As described in the previous section, a structured workflow process model is designed through the three types of control transitions—sequential, disjunctive and conjunctive transition—with keeping the matched pair and proper nesting properties. Therefore, the workflow mining algorithm must be obligated to rediscover these transitions by amalgamating the temporal workcases of a workflow process log. The basic idea of the amalgamation procedure conducted by the algorithm is to incrementally amalgamate one workcase model after another. Also, during the amalgamation procedure works, the most important thing is to observe and seek those three types of transitions.

Precisely, the basic amalgamating principles seeking each of the transition types are as follows: if a certain activity is positioned at the same temporal order in all workcase models, then the activity is to be involved in a sequential transition; else if the activity is at the different temporal order in some workcase models, then we can infer that the activity is to be involved in a conjunctive transition; otherwise if the activity is either presented in some workcase models or not presented in the other workcase models, then it has got to be involved in a disjunctive transition.

As simple examples of the amalgamating principles, we algorithmically illustrate the amalgamation procedures rediscovering a conjunctive transition and a disjunctive transition through simple examples. As an example of the conjunctive transition, suppose we examine the workflow process log of a structured workflow process model that has three activities, a_1, a_3 and a_4, and try to amalgamate two specific workcase models; the temporal order of a_3 and a_4 in one workcase model, is reversed on the other workcase model. Therefore, we can

infer that the activities, a_3 and a_4, are involved in a conjunctive transition of the structured workflow process.

As an example of the disjunctive transition, we also assume that we examine the workflow process log of a structured workflow process model that has four activities, a_1, a_3, a_4 and a_5, and try to amalgamate two specific workcase models; the temporal order of a_1 and a_5 in one workcase model is same on the other workcase model; also, the positions of a_3 and a_4 on the temporal order are same in these two workcase models respectively, and, while on the other, the activities, a_3 and a_4, are not presented in these two workcase models at the same time. Therefore, we can infer that the activities, a_3 and a_4, are involved in a disjunctive transition of the structured workflow process.

3.2.4 SWPM Rediscovering Algorithm

Based upon the basic amalgamating principles, we conceive a workflow mining algorithm in order to rediscover a reasonable structured workflow process model from a workflow process log. We name it σ-*Algorithm*, because its basic idea is to incrementally amalgamate the temporal workcases, which is just reflecting the conceptual idea of the summation operator($\sum$) in mathematics. Because of the page limitation we would not make a full description of the algorithm in here. However, we just introduce the detailed algorithm as follows, which is pseudo-coded as detail as possible with some explanations in comments, so that one is able to easily grasp the algorithm without the full description.

PROCEDURE SWPMRediscovery():

```
1:  Input : A Set of Temporal Workcases, ∀(wc[i], i = 1..m);
2:              where, wc[1] == START(▽), wc[m] == END(△);
3:  Output : (1) A Rediscovered Structured Workflow Process Model (SWPM), R = (δ, κ, I, O);
4:              - The Activity Set of SWPM, A = {α_1 .. α_n}, (wc[i], i = 1..m) ∈ A;
5:           (2) A Set of Workcase Models (WCMs), ∀ W = (ω, P, S);
6:
7:  Initialize : δ_i(START(▽)) ← {NULL};
8:               δ_o(END(△)) ← {NULL};
9:  PROCEDURE WPMRediscovery()
10: BEGIN
11:     WHILE ((wc[] ← readOneWorkcase()) ≠ EOF) DO
12:         i ← 1;
13:         WHILE (wc[i] ≠ END(△)) DO
14:             ω_o(wc[i]) ← wc[i + 1]; i ← i + 1; ω_i(wc[i]) ← wc[i − 1];
15:         END WHILE
16:         /* Rediscovering the temporary RWPM from the current WCM */
17:         FOR (i = 1; i < m; i + +;) DO
18:             IF (Is δ_o(wc[i]) an empty-set?) THEN
19:                 δ_o(wc[i]) ← ω_o(wc[i]); continue;
20:             END IF
21:             IF (isANDTransition(wc[i], ω_o(wc[i])) == TRUE) THEN
22:                 continue;
23:             END IF
24:             FOR (each set, a, of sets in δ_o(wc[i])) DO
25:                 SWITCH (checkupTransition(a, ω_o(wc[i])) DO
26:                     Case 'fixed transition':
27:                     Case 'sequential relationship':
28:                         δ_o(wc[i]) ← ω_o(wc[i]);
29:                         break;
30:                     Case 'conjunctive transition (AND-split)':
31:                         ANDset ← makeANDTransition(a, ω_o(wc[i]));
32:                         δ_o(wc[i]) ← δ_o(wc[i]) ∪ ANDset;
```

```
33:                                   eliminatePreviousTransition(a, ω_o(wc[i]));
34:                                   break;
35:                             Case 'disjunctive transition (OR-split)':
36:                                   ORset ← makeORTransition(a, ω_o(wc[i]));
37:                                   δ_o(wc[i]) ← δ_o(wc[i]) ∪ ORset;
38:                                   eliminatePreviousTransition(a, ω_o(wc[i]));
39:                                   break;
40:                             Default: /* Exceptions */
41:                                   printErrorMessage();
42:                                   break;
43:                       END SWITCH
44:                 END FOR
45:           END FOR
46:      END WHILE
47:      finishupSWPM(); /* with its input-activity sets, (δ_i(wc[i]), i = 1..n)
48:                                                   and its transition-conditions */
49:      δ_i(α_1) ← {START(▽)}; δ_o(α_n) ← {END(△)};
50:      PRINTOUT
51:           (1) The Rediscovered Structured Workflow Process Model, SWPM, R = (δ, κ, I, O);
52:           (2) A Set of the Workcase Models, WCMs, ∀W = (ω, P, S);
53: END PROCEDURE
```

Finally, the algorithm's operational example is algorithmically illustrated in Fig. 3. The right-hand side of the figure is the rediscovered structured workflow process model that the algorithm mines from the temporal workcases, which are the typical four temporal workcases possibly produced from the original structured workflow process model. As might be expected, the algorithm doesn't care the original model; nevertheless, we need it to generate a set of temporal workcases and verify the algorithm. Fortunately, we are able to imagine that the original model produces the following four *StartedTime temporal workcases*: (1) $a_1 \rightarrow a_2 \rightarrow a_4 \rightarrow a_3 \rightarrow a_6$ (2) $a_1 \rightarrow a_2 \rightarrow a_5 \rightarrow a_3 \rightarrow a_6$ (3) $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_4 \rightarrow a_6$ (4) $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow a_5 \rightarrow a_6$.

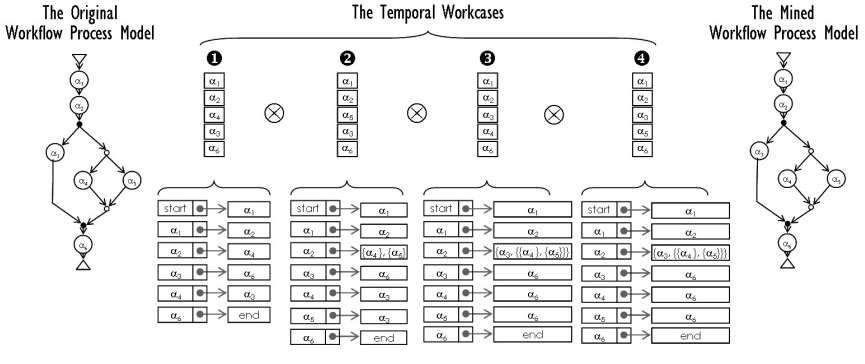


Fig. 3. An Operational Example mining a Structured Workflow Process Model

3.3 Constraints of the Algorithm

As emphasized in the previous sections, this algorithm is operable on the concept of structured workflow process model that retains the proper nesting and

matched pair properties[16]. Keeping these properties causes to constrain the algorithm as well as the modeling work; nevertheless, it might be worthy to preserve the constraints because they can play a very important role in increasing the integrity of the workflow model. Additionally, not only the improperly nested workflow model makes its analysis complicated, but also the workflow model with unmatched pairs may be stuck and run into a deadlock situation during its runtime execution.

Another important issue in designing workflow mining algorithms is about how to handle loop transitions in a workflow process model, because they may produce not only a lot of workflow event logs but also much more complicated patterns of temporal workcases. Precisely, according to the number of repetitions and the inside structure of a loop transition, the model's execution may generate very diverse and complicated patterns of temporal workcases. Therefore, the algorithm proposed in this paper has got to be extended in order to properly handle the loop transitions. We would leave this issue to our future research work.

4 Related Works

So far, there have been several workflow mining related researches and developments in the workflow literature. Some of them have proposed the algorithms [1,3,4,5,8,9,11,13,15] for workflow mining functionality, and others have developed the workflow mining systems and tools[2,6]. Particularly, as the first industrial application of the workflow mining, J. Herbsta and D. Karagiannisb in [2] presented the most important results of their experimental evaluation and experiences of the InWoLvE workflow mining system. However, almost all of the contribution are still focusing on the development of the basic functionality of workflow mining techniques. Especially, W.M.P. van der Aalst's research group, through the papers of [1,9,14], proposed the fundamental definition and the use of workflow mining to support the design of workflows, and described the most challenging problems and some of the workflow mining approaches and algorithms. Also, Clarence Ellis's research group newly defined the scope of workflow mining concept from the view point of that workflow systems are "people systems" that must be designed, deployed, and understood within their social and organizational contexts. Thus, they argue in [11,12] that there is a need to expand the concept of workflow discovery beyond the process dimension to encompass multidimensional perspective such as social, organizational, and informational perspectives; as well as other perspectives. This paper is the partial result of the collaborative research on mining the workflow's multidimensional perspectives.

5 Conclusion

In this paper, we proposed a mining algorithm rediscovering a structured workflow process from the temporal workcases out of a workflow process log. The algorithm is based on the structured workflow process model designed by the

information control net workflow modeling methodology, and we showed that it is able to properly handle the three different types of control transitions—sequential, conjunctive and disjunctive transitions—through an operational example. Also, we need to extend the algorithm to cope with the loop transition in the near future. In a consequence, workflow mining methodologies and systems are rapidly growing and coping with a wide diversity of domains in terms of their applications and working environments. So, the literature needs various, advanced, and specialized workflow mining techniques and architectures that are used for finally giving feed-backs to the redesign and reengineering phase of the existing workflow models. We strongly believe that this work might be one of those impeccable attempts and pioneering contributions for improving and advancing the workflow mining technology.

References

1. W. M. P. van der Aalst, et al: Workflow mining: A survey of issues and approaches. *Journal of Data & Knowledge Engineering*, Vol. 47, Issue 2, pp. 237-267, 2003
2. Joachim Herbsta, et al: Workflow mining with InWoLvE. *Journal of Computers in Industry*, Vol. 53, Issue 3, Elsevier, 2004
3. Guido Schimm: Mining exact models of concurrent workflows. *Journal of Computers in Industry*, Vol. 53, Issue 3, Elsevier, 2004
4. Shlomit S. Pinter, et al: Discovering workflow models from activities' lifespans. *Journal of Computers in Industry*, Vol. 53, Issue 3, Elsevier, 2004
5. Kwanghoon Kim and Clarence A. Ellis: Workflow Reduction for Reachable-path Rediscovery in Workflow Mining. *Series of Studies in Computational Intelligence: Foundations and Novel Approaches in Data Mining*, Vol. 9, pp. 289-310, 2006
6. Kwanghoon Kim: A Workflow Trace Classification Mining Tool. *International Journal of Computer Science and Network Security*, Vol.5, No. 11, pp. 19-25, 2005
7. Kwanghoon Kim, et al: A XML-Based Workflow Event Logging Mechanism for Workflow Mining. *Proc. ICCSA 2007, To-be Published*, 2007
8. R. Agrawal, et al: Mining Process Models from Workflow Logs. *Proc. Int. Conf. on Extending Database Technology*, 1998
9. A.K.A. de Medeiros, et al: Process Mining: Extending the alpha-algorithm to Mine Short Loops. *BETA Working Paper Series*, 2004
10. C. Ellis: Information Control Nets: A Mathematical Model of Information Flow. *ACM Proc. Conf. on Simulation, Modeling and Measurement of Computer Systems*, pp. 225-240. ACM, 1979
11. C. Ellis, et al: Workflow Mining: Definitions, Techniques, and Future Directions. *Workflow Handbook 2006*, pp. 213-228, 2006
12. C. Ellis, et al: Beyond Workflow Mining. *Lecture Notes in Computer Science*, Vol. 4102, pp. 49-64, 2006
13. R. Silva, J. Zhang, and J.G. Shanahan. Probabilistic Workflow Mining. *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*, 2005
14. W. M. P. van der Aalst, A. K. A. de Medeiros, and A. J. M. M. Weijters. Genetic Process Mining. *Proc. Int. Conf. on ATPN*, pages 48-69, 2005
15. W. Gaaloul and C. Godart. Mining Workflow Recovery from Event Based Logs. *Lecture Notes in Computer Science*, Vol. 3649, pp. 169-185, 2005
16. R. Liu and A. Kumar. An Analysis and Taxonomy of Unstructured Workflows. *Lecture Notes in Computer Science*, Vol. 3649, pp. 268-284, 2005

Multiscale BiLinear Recurrent Neural Network for Prediction of MPEG Video Traffic

Min-Woo Lee[1], Dong-Chul Park[1], and Yunsik Lee[2]

[1] Dept. of Information Eng. and Myongji IT Eng. Research Inst.
Myong Ji University, Korea
{monolio,parkd}@mju.ac.kr
[2] SoC Research Center, Korea Electronics Tech. Inst., Seongnam, Korea
leeys@keti.re.kr

Abstract. A MPEG video traffic prediction model in ATM networks using the Multiscale BiLinear Recurrent Neural Network (M-BLRNN) is proposed in this paper. The M-BLRNN is a wavelet-based neural network architecture based on the BiLinear Recurrent Neural Network (BLRNN). The wavelet transform is employed to decompose the time-series to a multiresolution representation while the BLRNN model is used to predict a signal at each level of resolution. The proposed M-BLRNN-based predictor is applied to real-time MPEG video traffic data. When compared with the MLPNN-based predictor and the BLRNN-based predictor, the proposed M-BLRNN-based predictor shows 16%-47% improvement in terms of the Normalized Mean Square Error (NMSE) criterion.

Keywords: MPEG, Recurrent Neural Networks.

1 Introduction

The dynamic nature of bursty traffic data in Asynchronous Transfer Mode (ATM) networks may cause severe network congestion when a number of bursty sources are involved. Therefore, the demand for dynamic bandwidth allocation to optimally utilize the network resources and satisfy Quality of Service(QoS) requirements should be taken into account. In order to dynamically adapt for bandwidth allocation, prediction of the future network traffic generated by end-users according to the observed past traffic in the network plays a very important role in ATM networks. Various traffic prediction models have been proposed for MPEG video traffic prediction. Classical linear models such as the Autogressive (AR) model [1] and adaptive linear model [2] have been widely used in practice. However, these models may be not suitable for predicting traffic over ATM networks due to the bursty characteristics of these networks.

A number of new nonlinear techniques have been proposed for MPEG video traffic prediction. Among them, the neural network (NN)-based models have received significant attention [3,4]. These recent studies reported that satisfactory traffic prediction accuracy can be achieved for a single-step prediction,i.e., the prediction for only next frame. However, the sing-step prediction may not

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 131–137, 2007.

be suitable in application such as dynamic bandwidth allocation since it is impractical to reallocate the bandwidth frequently for a single frame. Therefore, multi-step prediction of MPEG video traffic should be explored.

In this paper, a MPEG video traffic prediction model using a Multiscale BiLinear Recurrent Neural Network (M-BLRNN) [5]is proposed. The M-BLRNN is a wavelet-based neural network architecture based on the BiLinear Recurrent Neural Network (BLRNN)[6]. The M-BLRNN is formulated by a combination of several individual BLRNN models in which each individual model is employed for predicting the signal at a certain level obtained by the wavelet transform.

The remainder of this paper is organized as follows: Section 2 presents a review of multiresolution analysis with the wavelet transform. A brief review of the BLRNN is given in Section 3. The proposed M-BLRNN-based predictor is presented in Section 4. Section 5 presents some experiments and results on several real-time MPEG data sets including a performance comparison with the traditional MLPNN-based predictor and BLRNN-based predictor. Concluding remarks provided in Section 6 close the paper.

2 Multiresolution Wavelet Analysis

The multiresolution analysis produces a high quality local representation of a signal in both the time domain and the frequency domain. The wavelet transform [7] has been proven suitable for the multiresolution analysis of time series data [8].

The *à* trous wavelet transform was first proposed by Shensa [7] and the calculation of the *à* trous wavelet transform can be described as follows: First, a low-pass filter is used to suppress the high frequency components of a signal and allow the low frequency components to pass through. The smoothed data $c_j(t)$ at a given resolution j can be obtained by performing successive convolutions with the discrete low-pass filter h,

$$c_j(t) = \sum_k h(k) c_{j-1}(t + 2^{j-1}k) \tag{1}$$

where h is a discrete low-pass filter associated with the scaling function and $c_0(t)$ is the original signal. A suitable low-pass filter h is the B_3 spline, defined as $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$.

From the sequence of the smoothing of the signal, the wavelet coefficients are obtained by calculating the difference between successive smoothed versions:

$$w_j(t) = c_{j-1}(t) - c_j(t) \tag{2}$$

By consequently expanding the original signal from the coarsest resolution level to the finest resolution level, the original signal can be expressed in terms of the wavelet coefficients and the scaling coefficients as follows:

$$c_0(t) = c_J(t) + \sum_{j=1}^{J} w_j(t) \tag{3}$$

where J is the number of resolutions and $c_J(t)$ is the finest version of the signal. Eq.(3) also provides a reconstruction formula for the original signal.

3 BiLinear Recurrent Neural Networks

The BLRNN is a simple recurrent neural network, which has a robust ability in modeling dynamically nonlinear systems and is especially suitable for time-series data. The model was initially proposed by Park and Zhu [6]. It has been successfully applied in modeling time-series data [6,9]. In the following, we summarize a simple BLRNN that has N input neurons, M hidden neurons and where $K = N - 1$ degree polynomials is given. The input signal and the nonlinear integration of the input signal to hidden neurons are:

$$\boldsymbol{X}[n] = [x[n], x[n-1], ..., x[n-K]]^T$$
$$\boldsymbol{O}[n] = [o_1[n], o_2[n], ..., o_M[n]]^T$$

where T denotes the transpose of a vector or matrix and the recurrent term is a $M \times K$ matrix

$$\boldsymbol{Z}_p[n] = [o_p[n-1], o_p[n-2], ..., o_p[n-K]]$$

And

$$\begin{aligned} s_p[n] &= w_p + \sum_{k_1=0}^{N-1} a_{pk_1} o_p[n-k_1] \\ &+ \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{N-1} b_{pk_1k_2} o_p[n-k_1] x[n-k_2] \\ &+ \sum_{k_2=0}^{N-1} c_{pk_2} x[n-k_2] \\ &= w_p + \boldsymbol{A}_p^T \boldsymbol{Z}_p^T[n] + \boldsymbol{Z}_p[n] \boldsymbol{B}_p^T \boldsymbol{X}[n] + \boldsymbol{C}_p^T \boldsymbol{X}[n] \end{aligned} \tag{4}$$

where w_p is the weight of bias neuron. $\boldsymbol{A}_p$ is the weight vector for the recurrent portion, $\boldsymbol{B}_p$ is the weight matrix for the bilinear recurrent portion, and $\boldsymbol{C}_p$ is the weight vector for the feedforward portion and $p = 1, 2..., M$.

More detailed information on the BLRNN and its learning algorithm can be found in [6,9].

4 Multiscale BiLinear Recurrent Neural Network

The M-BLRNN is a combination of several individual BLRNN models where each individual BLRNN model is employed to predict the signal at each

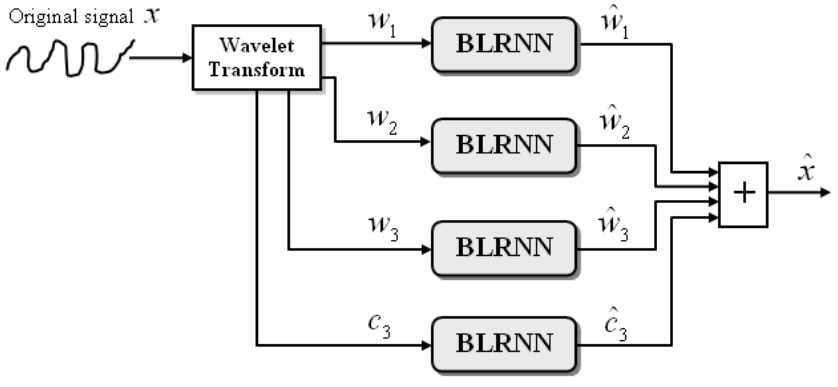


Fig. 1. Example of Multiscale BiLinear Recurrent Neural Network with 3 resolution levels

resolution level obtained by the wavelet transform [5]. Fig. 1 illustrates an example of the M-BLRNN with three levels of resolution.

The prediction of a time-series based on the M-BLRNN can be separated into three stages. In the first stage, the original signal is decomposed into the wavelet coefficients and the scaling coefficients based on the number of resolution levels. In the second stage, the coefficients at each resolution level are predicted by an individual BLRNN model. It should be noted that the predictions of coefficients at each resolution level are independent and can be done in parallel. In the third stage, all the prediction results from each BLRNN are combined together using the reconstruction formula given in Eq.(3):

$$\hat{x}(t) = \hat{c}_J(t) + \sum_{j=1}^{J} \hat{w}_j(t) \tag{5}$$

where $\hat{c}_J(t)$, $\hat{w}_j(t)$, and $\hat{x}(t)$ represent the predicted values of the finest scaling coefficients, the predicted values of the wavelet coefficients at level j, and the predicted values of the time-series, respectively.

5 Experiments and Results

The experiments were conducted based on several real-time MPEG trace sequences provided by the University of the Wuerzburg, Wuerzburg, Germany. These trace sequences can be downloaded at

http://www3.informatik.uni-wuerzburg.de/MPEG/

We selected 4 typical trace sequences for training and testing: "*Star Wars*", "*Mr. Bean*", "*New Show*", and "*Silence of the Lambs*". From these sequences, the first 1,000 frames of the "*Star Wars*" sequence were used for training while the remaining of "*Star Wars*" and other sequences were saved for testing. All

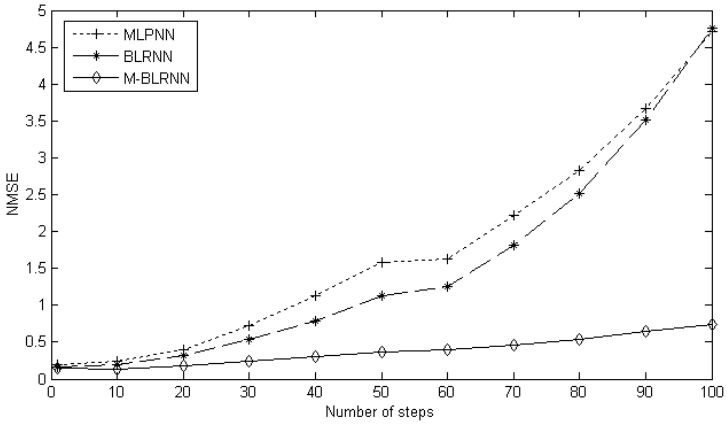


Fig. 2. Prediction performance versus number of steps for the *"Silence of the Lambs"* video trace

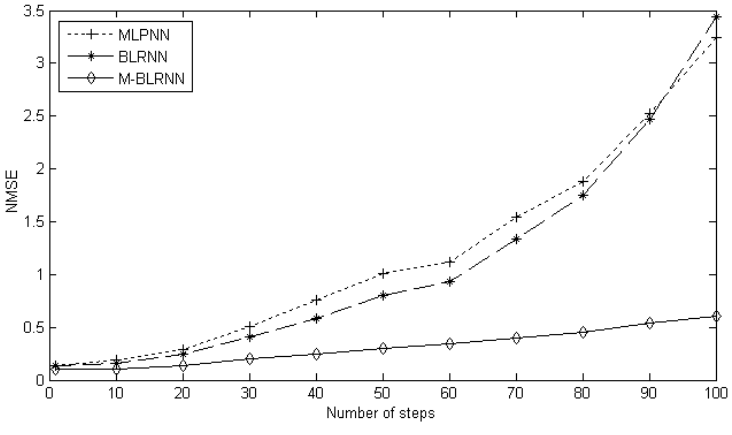


Fig. 3. Prediction performance versus number of steps for the *"News Show"* video trace

data were subsequently normalized in a range (0,1) to render it suitable for inputs of neural networks.

In order to demonstrate the generalization ability of the M-BLRNN-based predictor, a M-BLRNN model with 3 resolution levels using the adaptive learning algorithm is employed. Based on the statistical analysis of correlations, each individual BLRNN model in the M-BLRNN model shares a 24-10-1 structure and 3 recursion lines in which the indices denote the number of neurons in the input layer, the hidden layer and the output layer, respectively. A traditional BLRNN employing a structure of 24-10-1 with 3 recursion lines and a MLPNN model employing a structure of 24-10-1 are also employed for a performance comparison. The iterated multistep prediction [11] was employed to perform the

multistep prediction of the real-time MPEG video traffic. To measure the performance of the multistep prediction, the normalized mean square error (NMSE) was employed.

Figs. 2 and 3 show the prediction performance on the remainder of the the *"Silence of the Lambs"* and the *"News Show"*, respectively. As can be seen from these figures, the M-BLRNN-based predictor that employs the wavelet-based neural network architecture outperforms both the traditional MLPNN-based predictor and the BLRNN-based predictor. In particular, the M-BLRNN-based predictor can predict up to a hundred steps with a very small degradation of performance whereas the traditional MLPNN-based predictor and the BLRNN-based predictor fail to do so.

6 Conclusion

A MPEG traffic prediction model using a Multiscale BiLinear Recurrent Neural Network (M-BLRNN) is proposed in this paper. The M-BLRNN is a wavelet-based neural network architecture based on the BiLinear Recurrent Neural Network (BLRNN). The proposed M-BLRNN-based predictor is applied to the long-term prediction of MPEG video traffic. Experiments and results on several real-time MPEG data sets show a significant improvement in comparison with the traditional MLPNN-based predictor and BLRNN-based predictor. This confirms that the proposed M-BLRNN is an efficient tool for dynamic bandwidth allocation in ATM networks.

References

1. Nomura, N., Fujii, T., Ohta, N.: Basic Characteristics of Variable Rate Video Coding in ATM Environment. IEEE J. Select. Areas Commun., Vol. 7 (1989) 752-760.
2. Adas, A.M.: Using Adaptive Linear Prediction to Support Real-time VBR Video under RCBR Network Service Model. IEEE/ACM Trans. Networking 6 (1998) 635-644.
3. Doulamis, A.D., Doulamis, N.D., Kollias, S.D.: An Adaptable Neural Network Model for Recursive Nonlinear Traffic Prediction and Modeling of MPEG Video Sources. IEEE Trans. Neural Networks 14 (2003) 150 - 166.
4. Bhattacharya, A., Parlos, A.G., Atiya, A.F.: Prediction of MPEG-coded Video Source Traffic using Recurrent Neural Networks. IEEE Trans. on Acoustics, Speech, and Signal Processing 51 (2003) 2177 - 2190.
5. Park, D.C., Tran, C.N., Lee, Y.: Multiscale BiLinear Recurrent Neural Networks and Their Application to the Long-Term Prediction of Network Traffic, LNCS 3973 (2006), 196-201
6. Park, D.C., Zhu, Y.: Bilinear Recurrent Neural Network. IEEE ICNN, Vol. 3, (1994) 1459-1464.
7. Shensa, M.J.: The Discrete Wavelet Transform: Wedding the $\grave{A}$ Trous and Mallat Algorithms. IEEE Trans. Signal Proc. 10 (1992) 2463-2482.

8. Alarcon-Aquino, V., Barria, J.A.: Multiresolution FIR Neural-Network-Based Learning Algorithm Applied to Network Traffic Prediction. IEEE Trans. Sys. Man. and Cyber. PP(99) (2005) 1-13.
9. Park, D.C., Jeong, T.K.: Complex Bilinear Recurrent Neural Network for Equalization of a Satellite Channel. IEEE Trans on Neural Network 13 (2002) 711-725.
10. Kruschke, J.K., Movellan, J.R.: Benefits of Gain: Speeded Learning and Minimal Hidden Layers in Back-propagation Networks. IEEE Trans. on Systems, Man and Cybernetics 21(1) (1991) 273-280.
11. Parlos, A.G., Rais, O.T., Atiya, A.F.: Multi-step-ahead Prediction using Dynamic Recurrent Neural Networks. IJCNN '99. Int. Joint Conf. on Neural Networks, Vol. 1, (1999) 349 - 352.

An Effective Multi-level Algorithm Based on Ant Colony Optimization 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. An important application of graph partitioning is data clustering using a graph model — the pairwise similarities between all data objects form a weighted graph adjacency matrix that contains all necessary information for clustering. 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 based on ant colony optimization(ACO) for bisecting graph. The success of our algorithm relies on exploiting both the ACO method and the concept of the graph core. Our experimental evaluations on 18 different graphs show that our algorithm produces encouraging 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 partitioning problem and is NP-Complete [3]. It is also NP-Hard to find good approximate solutions for this problem [4]. 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 [5],[6]. The survey by Alpert and Kahng [7] provides a detailed description and comparison of various such schemes which can be classified as *move-based* approaches, *geometric representations*, *combinatorial* formulations, and *clustering* approaches.

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) [5] proposed a heuristic algorithm for partitioning graphs. The KL algorithm is an iterative improvement algorithm that consists of making several improvement passes. It starts with an initial bipartitioning and tries to improve it by every pass. A pass consists of the identification of two subsets of vertices, one from each part such that can lead to an improved partition if

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 138–149, 2007.

the vertices in the two subsets switch sides. Fiduccia and Mattheyses (FM) [6] proposed a fast heuristic algorithm for bisecting a weighted graph by introducing the concept of cell *gain* into the KL algorithm. These algorithms belong to the class of *move-based* approaches in which the solution is built iteratively from an initial solution by applying a move or transformation to the current solution. Move-based approaches are the most frequently combined with stochastic hill-descending algorithms such as those based on Simulated Annealing [8], Tabu Search [8],[9], Genetic Algorithms [10], Neural Networks [11], etc., which allow movements towards solutions worse than the current one in order to escape from local minima. For example, Leng and Yu [12],[13] proposed a boundary Tabu Search refinement algorithm that combines an effective Tabu Search strategy with a boundary refinement policy for refining the initial partitioning.

As the problem sizes reach new levels of complexity recently, it is difficult to compute the partitioning directly in the original graph and a new class of graph partitioning algorithms have been developed that are based on the multi-level paradigm. The multi-level graph partitioning schemes consist of three phases [14],[15],[16]. During the *coarsening phase*, a sequence of successively coarser graph is constructed by collapsing vertex and edge until its size is smaller than a given threshold. The goal of the *initial partitioning phase* is to compute initial partitioning of the coarsest graph such that the balancing constraint is satisfied and the partitioning objective is optimized. During the *uncoarsening phase*, the partitioning of the coarser graph is successively projected back to the next level finer graph and an iterative refinement algorithm is used to optimize the objective function without violating the balancing constraint.

In this paper, we present a multi-level algorithm which integrates an effective matching-based coarsening scheme and a new ACO-based refinement approach. Our work is motivated by the multi-level *ant* colony algorithm(MACA) of Korošec who runs basic *ant* colony algorithm on every level graph in [17] and Karypis who introduces the concept of the graph *core* for coarsening the graph in [16] and supplies **MeTiS** [14], 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 [18]. Our comparative experiments show that our algorithm produces excellent 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 multi-level ACO refinement algorithm. Section 5 experimentally evaluates our algorithm and compares it with **MeTiS**. Finally, Section 6 provides some concluding remarks and indicates the directions for further research.

2 Mathematical Description

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 $V^1 \cap V^2 = \varnothing$, is called a *bipartitioning* of V. Let $S(A) = \sum_{v \in A} S(v)$ denotes 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 different sides. 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 the 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 $(1\text{-}r)S(V)/2 \leq S(V^1), S(V^2) \leq (1+r)S(V)/2$. 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*.

3 Motivation

ACO is a novel population-based meta-heuristic framework for solving discrete optimization problems [19],[20]. It is based on the indirect communication among the individuals of a colony of agents, called *ants*, mediated by trails of a chemical substance, called *pheromone*, which real *ants* use for communication. It is inspired by the behavior of real *ant* colonies, in particular, by their foraging behavior and their communication through *pheromone* trails. The *pheromone* trails are a kind of distributed numeric information which is modified by the *ants* to reflect their experience accumulated while solving a particular problem. Typically, solution components which are part of better solutions or are used by many *ants* will receive a higher amount of *pheromone* and, hence, will more likely be used by the *ants* in future iterations of the algorithm. The collective behavior that emerges is a form of *autocatalytic* behavior. The process is thus characterized by a *positive feedback* loop, where the probability with which *ant* chooses a solution component increases with the number of *ants* that previously chose the same solution component.

The main idea of ACO is as follows. Each *ant* constructs candidate solutions by starting with an empty solution and then iteratively adding solution components until a complete candidate solution is generated. At every point each *ant* has to decide which solution component to be added to its current partial solution according to a *state transition rule*. After the solution construction is completed, the *ants* give feedback on the solutions they have constructed by depositing *pheromone* on solution components which they have used in their solution according to a *pheromone updating rule*.

In [21], Langham and Grant proposed the Ant Foraging Strategy (AFS) for k-way partitioning. The basic idea of the AFS algorithm is very simple: We have k colonies of *ants* that are competing for food, which in this case represents the vertices of the graph. At the end the *ants* gather food to their nests, i.e. they partition the graph into k subgraphs. In [17], Korošec presents the MACA approach that is enhancement of the AFS algorithm with the multi-level paradigm. However, since Korošec simply runs the AFS algorithm on every level ℓ graph $G_l(V_l,E_l)$, most of computation on the coarser graphs is wasted. Furthermore, MACA comes into collision with the key idea behind the multi-level approach. The multi-level graph partitioning schemes needn't the direct partitioning algorithm on $G_l(V_l,E_l)$ in the *uncoarsening and refinement phase*, but the refinement algorithm that improves the quality of the finer graph $G_l(V_l,E_l)$ partitioning $P_{G_l}=\{V_l^1,V_l^2\}$ which is projected from the partitioning $P_{G_{l+1}}=\{V_{l+1}^1,V_{l+1}^2\}$ of the coarser graph $G_{l+1}(V_{l+1},E_{l+1})$.

In this paper, we present a new multi-level *ant* colony optimization refinement algorithm(MACOR) that combines the ACO method with a boundary refinement policy. It employs ACO in order to select two subsets of vertices $V_l^{1'} \subset V_l^1$ and $V_l^{2'} \subset V_l^2$ such that $\{ (V_l^1 - V_l^{1'}) \cup V_l^{2'} , (V_l^2 - V_l^{2'}) \cup V_l^{1'} \}$is a bisection with a smaller edge-cut. It has distinguishing features which are different from the MACA algorithm. First, MACA exploits two or more colonies of *ants* to compete for the vertices of the graph, while MACOR employs one colony of *ants* to find $V_l^{1'}$ and $V_l^{2'}$ such that moving them to the other side improves the quality of partitioning. Second, MACA is a partitioning algorithm while MACOR is a refinement algorithm. Finally, MACOR is a boundary refinement algorithm whose runtime is significantly smaller than that of a non-boundary refinement algorithm, since the vertices moved by MACOR are boundary vertices that straddle two sides of the partition and only the gains of boundary vertices are computed.

In [14], Karypis presents the sorted heavy-edge matching (SHEM) algorithm that identifies and collapses together groups of vertices that are highly connected. 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 [22], Sediman introduces 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. Vladimir gives an $O(m)$-time algorithm for cores decomposition of networks and $O(m\cdot \log(n))$-time algorithm to compute the *core* numbering in the context of sum-of-the-edge-weights in [23],[24] respectively. In [16], Amine and Karypis introduce the concept of the graph *core* for coarsening the *power-law* graphs. In [13], Leng present the core-sorted heavy-edge matching (CSHEM) algorithm that combines the concept of the graph *core* with the SHEM scheme. Firstly, CSHEM sorts the vertices of the graph descendingly based on the *core* number of the vertices by the algorithm in [24]. 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.

In our multi-level algorithm, we adopt the MACOR algorithm during the *refinement phase* , the greedy graph growing partition (GGGP) algorithm [14] during the *initial partitioning phase*, an effective matching-based coarsening scheme during the *coarsening phase* that uses the CSHEM algorithm on the original graph and the SHEM algorithm on the coarser graphs. The pseudocode of our multi-level algorithm is shown in Algorithm 1.

Algorithm 1 (Our multi-level algorithm)

INPUT: original graph $G(V,E)$
OUTPUT: the partitioning P_G of graph G
/**coarsening phase*/*
$l = 0$
$G_l(V_l,E_l)$=$G(V,E)$
$G_{l+1}(V_{l+1},E_{l+1})$=CSHEM($G_l(V_l,E_l)$)
While ($|V_{l+1}| > 20$) do
  $l = l + 1$
  $G_{l+1}(V_{l+1},E_{l+1})$=SHEM($G_l(V_l,E_l)$)
End While
/**initial partitioning phase*/*
P_{G_l}=GGGP(G_l)
/**refinement phase*/*
While ($l \geq 1$) do
  P'_{G_l}=MACOR(G_l,P_{G_l})
  Project P'_{G_l} to $P_{G_{l-1}}$;
  $l = l - 1$
End While
P_G=MACOR(G_l,P_{G_l})
Return P_G

4 An Effective Multi-level Ant Colony Optimization Refinement Algorithm

Informally, the MACOR algorithm works as follows: At time zero, an initialization phase takes place during which the internal and external degrees of all vertices are computed and initial values for *pheromone* trail are set on the vertices of graph G. In the main loop of MACOR, each *ant*'s tabu list is emptied and each *ant* chooses ($V^{1'}$, $V^{2'}$) by repeatedly selecting boundary vertices of each part according to a *state transition rule* given by Equation(1) (2), moving them into the other part, updating the gains of the remaining vertices and etc. After constructing its solution, each *ant* also modifies the amount of *pheromone* on the moved vertices by applying the *local updating rule* of Equation(3). Once all *ants* have terminated their solutions, the amount of *pheromone* on vertices is modified again by applying the *global updating rule* of Equation(4). The process is iterated until the cycles counter reaches the maximum number of cycles NC_{max}, or the MACOR algorithm stagnates.

The pseudocode of the MACOR algorithm is shown in Algorithm 2. The cycles counter is denoted by t and *Best* represents the best partitioning seen so far. The initial values for *pheromone* trail is denoted by τ_0=1/ε, where ε is total number of *ants*. At cycle t, let $\tau_v(t)$ be the *pheromone* trail on the vertex v and $tabu^k(t)$ be the tabu list of *ant* k, $Best^k(t)$ represents the best partitioning found by *ant* k and the current partitioning of *ant* k is denoted by $P^k(t)$, the *ant* k also stores the internal and external degrees of all vertices and boundary vertices independently which be denoted as $ID^k(t)$, $ED^k(t)$ and $boundary^k(t)$ respectively. Let $allowed^k(t)$ be denoted as the *candidate* list which is a list of preferred vertices to be moved by *ant* k at cycle t and is equal to $\{V - tabu^k(t)\} \bigcap boundary^k(t)$.

Algorithm 2 (MACOR)

INPUT: initial bipartitioning P, maximum number of cycles NC_{max}
balance constraint r, similarity tolerance φ, maximum steps s_{max}
OUTPUT: the best partitioning *Best*, cut of the best partitioning $cut(Best)$
/**Initialization**/
$t = 0$
$Best = P$
For every vertex v in $G = (V, E)$ do
$ID_v = \sum_{(v,u)\in E \wedge P[v]=P[u]} W(v,u)$
$ED_v = \sum_{(v,u)\in E \wedge P[v]\neq P[u]} W(v,u)$
Store v as *boundary vertex* if and only if $ED_v > 0$;
$\tau_v(t) = \tau_0$
End For
/**Main loop**/
For $t = 1$ to NC_{max} do
For $k = 1$ to ε do
$tabu^k(t) = \varnothing$
Store $P^k(t) = P$ and $Best^k(t) = P$ independently;
Store $ID^k(t)$, $ED^k(t)$, $boundary^k(t)$ of $G = (V, E)$ independently;
For $s = 1$ to s_{max} do
Decide the *move direction* of the current step s;
If exists at least one vertex $v \in allowed^k(t)$ then
Choose the vertex v to move as follows

$$v = \begin{cases} \arg \max_{v \in allowed^k(t)} [\tau_v(t)]^\alpha \cdot \left[\eta_v^k(t)\right]^\beta & \text{if } q \le q_0 \\ w & \text{if } q > q_0 \end{cases} \tag{1}$$

Where the vertex w is chosen according to the probablity

$$p_w^k(t) = \begin{cases} \frac{[\tau_w(t)]^\alpha \cdot \left[\eta_w^k(t)\right]^\beta}{\sum_{u \in allowed^k(t)} [\tau_u(t)]^\alpha \cdot [\eta_u^k(t)]^\beta} & \text{if } w \in allowed^k(t) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Else
Break;
End If
Update $P^k(t)$ by moving the vertex v to the other side;
Lock the vertex v by adding to $tabu^k(t)$;
original cut Minus its original *gain* as the *cut* of $P^k(t)$;
Update $ID_u^k(t)$, $ED_u^k(t)$, *gain* of its neighboring vertices u and $boundary^k(t)$;
If ($cut(P^k(t)) < cut(Best^k(t))$ and $P^k(t)$ satisfies constraints r) then $Best^k(t) = P^k(t)$
End If
End For /*$s \leq s_{max}$*/
Apply the *local update rule* for the vertices v moved by *ant* k

$$\tau_v(t) \leftarrow (1-\rho) \cdot \tau_v(t) + \rho \cdot \triangle\tau_v^k(t) \tag{3}$$

Adjust q_0 if $similarity((V^{1'}, V^{2'})^k, (V^{1'}, V^{2'})^{(k\text{-}1)}) \geq \varphi$;
End For /*$k \leq \varepsilon$*/
If $\min\limits_{1 \leq k \leq \varepsilon} cut(Best^k(t)) < cut(Best)$ then
Update *Best* and $cut(Best)$;
End If
Apply the *global update rule* for the vertices v moved by *global-best ant*

$$\tau_v(t) \leftarrow (1-\xi) \cdot \tau_v(t) + \xi \cdot \triangle\tau_v^{gb} \tag{4}$$

For every vertex v in $G = (V, E)$ do
$\tau_v(t+1) = \tau_v(t)$
End For
End For /*$t \leq NC_{max}$*/
Return *Best* and $cut(Best)$

In the MACOR algorithm, a *state transition rule* given by Equation(1)(2) is called *pseudo-random-proportional rule*, where q is a random number uniformly distributed in $[0 \ldots 1]$ and q_0 is parameter $(0 \leq q_0 \leq 1)$ which determines the relative importance of exploitation versus exploration. If $q \leq q_0$ then the best vertex, according to Equation(1), is chosen(exploitation), otherwise a vertex is chosen according to Equation(2)(exploration). To avoid trapping into *stagnation behavior*, MACOR adjusts dynamically the parameter q_0 based on the solutions similarity between $(V^{1'}, V^{2'})^k$ and $(V^{1'}, V^{2'})^{(k\text{-}1)}$ found by *ant* k and *k-1*. In Equation(1)(2), α and β denote the relative importance of the *pheromone* trail $\tau_v(t)$ and *visibility* $\eta_v^k(t)$ respectively, $\eta_v^k(t)$ represents the *visibility* of *ant* k on the vertex v at cycle t and is given by:

$$\eta_v^k(t) = \begin{cases} \sqrt{1.0 + ED_v^k(t) - ID_v^k(t)} & \text{if } (ED_v^k(t) - ID_v^k(t)) \geq 0 \\ \sqrt{1.0/(ID_v^k(t) - ED_v^k(t))} & \text{otherwise} \end{cases} \tag{5}$$

In Equation(3), ρ is a coefficient and represents the local evaporation of *pheromone* trail between cycle t and $t+1$ and the term $\triangle\tau_v^k(t)$ is given by:

$$\triangle\tau_v^k(t) = \begin{cases} \frac{cut(Best^k(t)) - cut(P)}{cut(P)\cdot\varepsilon} & \text{if } v \text{ was moved by } ant\ k \text{ at cycle } t \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

In Equation(4), ξ is a parameter and represents the global evaporation of *pheromone* trail between *cycle* t and $t+1$ and the term $\triangle\tau_v^{gb}$ is given by:

$$\triangle\tau_v^{gb} = \begin{cases} \frac{cut(Best) - cut(P)}{cut(P)} & \text{if } v \text{ was moved by } global\text{-}best\ ant \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

5 Experimental Results

We use the 18 graphs in our experiments that are converted from the hypergraphs of the ISPD98 benchmark suite [18] and range from 12,752 to 210,613 vertices. Each hyperedge is a subset of two or more vertices in hypergraph. We convert hyperedges into edges by the rule that every subset of two vertices in hyperedge can be seemed as edge. We create the edge with unit weight if the edge that connects two vertices doesn't exist, else add unit weight to the weight of the edge. Next, we get the weights of vertices from the ISPD98 benchmark. Finally, we store 18 edge-weighted and vertex-weighted graphs in format of **MeTiS** [14]. The characteristics of these graphs are shown in Table 1.

Table 1. The characteristics of 18 graphs to evaluate our algorithm

benchmark	vertices	hyperedges	edges
ibm01	12752	14111	109183
ibm02	19601	19584	343409
ibm03	23136	27401	206069
ibm04	27507	31970	220423
ibm05	29347	28446	349676
ibm06	32498	34826	321308
ibm07	45926	48117	373328
ibm08	51309	50513	732550
ibm09	53395	60902	478777
ibm10	69429	75196	707969
ibm11	70558	81454	508442
ibm12	71076	77240	748371
ibm13	84199	99666	744500
ibm14	147605	152772	1125147
ibm15	161570	186608	1751474
ibm16	183484	190048	1923995
ibm17	185495	189581	2235716
ibm18	210613	201920	2221860

We implement the MACOR algorithm in ANSI C and integrate it with the leading edge partitioner **MeTiS**. In the evaluation of our multi-level 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 GGGP algorithm during the *initial partitioning phase* that consistently finds smaller edge-cuts than other algorithms, the boundary KL (BKL) refinement algorithm during the *uncoarsening and refinement phase* because BKL can produce smaller edge-cuts when coupled with the SHEM algorithm. These measures are sufficient to guarantee that our experimental evaluations are not biased in any way.

Table 2. Min-cut bipartitioning results with up to 2% deviation from exact bisection

benchmark	MeTiS(α)		our algorithm(β)		ratio(β:α)		improvement	
	MinCut	AveCut	MinCut	AveCut	MinCut	AveCut	MinCut	AveCut
ibm01	517	1091	259	531	0.501	0.487	49.9%	51.3%
ibm02	4268	11076	1920	5026	0.450	0.454	55.0%	54.6%
ibm03	10190	12353	4533	5729	0.445	0.464	55.5%	53.6%
ibm04	2273	5716	2221	3037	0.977	0.531	2.3%	46.9%
ibm05	12093	15058	8106	9733	0.670	0.646	33.0%	35.4%
ibm06	7408	13586	2111	5719	**0.285**	0.421	**71.5%**	57.9%
ibm07	3219	4140	2468	3110	0.767	0.751	23.3%	24.9%
ibm08	11980	38180	10500	13807	0.876	**0.362**	12.4%	**63.8%**
ibm09	2888	4772	2858	3905	0.990	0.818	1.0%	18.2%
ibm10	10066	17747	5569	7940	0.553	0.447	44.7%	55.3%
ibm11	2452	5095	2405	3423	0.981	0.672	1.9%	32.8%
ibm12	12911	27691	5502	13125	0.426	0.474	57.4%	52.6%
ibm13	6395	13469	4203	6929	0.657	0.514	34.3%	48.6%
ibm14	8142	12903	8435	10114	**1.036**	0.784	**-3.6%**	21.6%
ibm15	22525	46187	17112	25102	0.760	0.543	24.0%	45.7%
ibm16	11534	22156	8590	12577	0.745	0.568	25.5%	43.2%
ibm17	16146	26202	13852	18633	0.858	0.711	14.2%	28.9%
ibm18	15470	20018	15494	18963	1.002	**0.947**	-0.2%	**5.3%**
average					**0.721**	**0.589**	**27.9%**	**41.1%**

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 results, two measures are obtained in twenty runs whose random seed is different to each other. For all experiments, we allow the balance constraint up to 2% deviation from exact bisection by setting r to 0.02, i.e., each partition must have between 49% and 51% of the total vertices size. We also set the number of vertices of the current level graph as the

value of parameter s_{max}. Furthermore, we adopt the experimentally determined optimal set of parameters values for MACOR, α=2.0, β=1.0, ρ=0.1, ξ=0.1, q_0=0.9, φ=0.9, NC_{max}=80, ε=10.

Table 2 presents *min-cut bipartitioning* results allowing up to 2% deviation from exact bisection and Fig. 1 illustrates the MinCut and AveCut comparisons of two algorithms on 18 graphs. As expected, our algorithm reduces the AveCut by 5.3% to 63.8% and reaches 41.1% average AveCut improvement. Although our algorithm produces partition whose MinCut is up to 3.6% worse than that of **MeTiS** on two benchmarks, we still obtain 27.9% average MinCut improvement and between -3.6% and 71.5% 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 two hours.

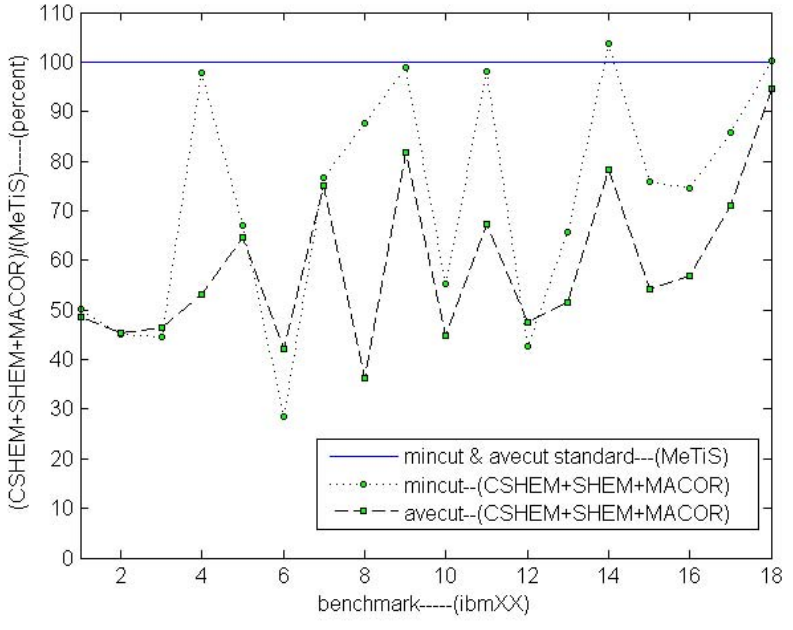


Fig. 1. The MinCut and AveCut comparisons of two algorithms on 18 graphs

6 Conclusions

In this paper, we have presented an effective multi-level algorithm based on ACO. The success of our algorithm relies on exploiting both the ACO method 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. For example, we note that adopting the CSHEM algorithm alone leads to poorer experimental results than

the combination of CSHEM with SHEM. We need to find the reason behind it and develop a better matching-based coarsening scheme coupled with MACOR. In the MinCut evaluation of benchmark ibm14 and ibm18, our algorithm is 3.6% worse than **MeTiS**. Therefore, the second question is to guarantee find good approximate solutions by setting optimal set of parameters values for MACOR.

Acknowledgments

This work was supported by the international cooperation project of Ministry of Science and Technology of PR China, grant No. CB 7-2-01, 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 would 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., He, X., Zha, H., Gu, M., 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. Bui, T., Leland, C.: Finding good approximate vertex and edge partitions is NP-hard. Information Processing Letters, Vol. 42 (1992) 153–159
5. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. Bell System Technical Journal, Vol. 49 (1970) 291–307
6. Fiduccia, C., Mattheyses, R.: A linear-time heuristics for improving network partitions. Proc. 19th Design Automation Conf (1982) 175–181
7. Alpert, C.J., Kahng, A.B.: Recent directions in netlist partitioning. Integration, the VLSI Journal, Vol. 19 (1995) 1–81
8. Tao, L., Zhao, Y.C., Thulasiraman, K., Swamy, M.N.S.: Simulated annealing and tabu search algorithms for multiway graph partition. Journal of Circuits, Systems and Computers (1992) 159–185
9. Kadłuczka, P., Wala, K.: Tabu search and genetic algorithms for the generalized graph partitioning problem. Control and Cybernetics (1995) 459–476
10. Żola, J., Wyrzykowski, R.: Application of genetic algorithm for mesh partitioning. Proc. Workshop on Parallel Numerics (2000) 209-217
11. Bahreininejad, A., Topping, B.H.V., Khan, A.I.: Finite element mesh partitioning using neural networks. Advances in Engineering Software (1996) 103–115
12. Leng, M., Yu, S., Chen, Y.: An effective refinement algorithm based on multi-level paradigm for graph bipartitioning. The IFIP TC5 International Conference on Knowledge Enterprise, IFIP Series, Springer (2006) 294–303
13. Leng, M., Yu, S.: An effective multi-level algorithm for bisecting graph. The 2nd International Conference on Advanced Data Mining and Applications, Lecture Notes in Artifical Intelligence Series, Springer-Verlag (2006) 493–500

14. 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)
15. Selvakkumaran, 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
16. Amine, A.B., Karypis, G.: Multi-level 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
17. Korošec, P., Šilc, J., Robič, B.: Solving the mesh-partitioning problem with an ant-colony algorithm, Parallel Computing (2004) 785–801
18. Alpert, C.J.: The ISPD98 circuit benchmark suite. Proc. Intel Symposium of Physical Design (1998) 80–85
19. Dorigo, M., Gambardella.,L.: Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation (1997) 53–66
20. Dorigo, M., Maniezzo, V., Colorni., A.: Ant system: Optimization by a colony of cooperating agents. IEEE Trans on SMC (1996) 29–41
21. Langham, A.E., Grant, P.W.: Using competing ant colonies to solve k-way partitioning problems with foraging and raiding strategies. Advances in Artificial Life, Lecture Notes in Computer Science Series, Springer-Verlag (1999) 621–625
22. Seidman, S.B.: Network structure and minimum degree. Social Networks (1983) 269–287
23. Batagelj, V., Zaveršnik, M.: An $O(m)$ Algorithm for cores decomposition of networks. Journal of the ACM (2001) 799–804
24. Batagelj, V., Zaveršnik, M.: Generalized cores. Journal of the ACM (2002) 1–8

A Unifying Method for Outlier and Change Detection from Data Streams Based on Local Polynomial Fitting

Zhi Li[1], Hong Ma[1], and Yongbing Mei[2]

[1] Department of Mathematics, Sichuan University, Chengdu, 610064, China
zhili_mail@163.com
[2] Southwest China Institute of Electronic Technology, Chengdu, 610036, China
meiybmail@sina.com

Abstract. Online detection of outliers and change points from a data stream are two very exciting topics in the area of data mining. This paper explores the relationship between these two issues, and presents a unifying method for dealing with both of them. Previous approaches often use parametric techniques and try to give exact results. In contrast, we present a nonparametric method based on local polynomial fitting, and give approximate results by fuzzy partition and decision. In order to measure the possibility of being an outlier and a change point, two novel score functions are defined based on the forward and backward prediction errors. The proposed method can detect outliers and changes simultaneously, and can distinguish between them. Comparing to the conventional parametric approaches, our method is more convenient for implementation, and more appropriate for online and interactive data mining. Simulation results confirm the effectiveness of the proposed method.

Keywords: Data stream, outlier, change point, data mining, local polynomial fitting, fuzzy partition.

1 Introduction

As there is a growing number of emerging applications of data streams, mining of data streams is becoming increasingly important. Recent research indicates that online mining of the changes in data streams is one of the core issues with applications in event detection and activity monitoring [1]. On the other hand, outlier detection is also a popular issue in data mining, which is closely related to fraud detection, abnormality discovery and intrusion detection [2]. In the previous literature, outlier detection and change detection are often derived from respective problems and are addressed independently. Both statistical methods and fuzzy approaches have been employed to solve these two issues, such as methods based on regression analysis, hidden Markov model (HMM), hierarchical Bayesian model and fuzzy clustering, fuzzy entropy principle [7], etc.

However, the outliers and change points often exist simultaneously in real data streams. It's necessary to design a unifying method to detect the outliers and change points simultaneously. Thus, in this paper, we explore the relationship between outlier and change detection, and deal with them together. Intuitively, an outlier is a point

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 150–161, 2007.

largely deviating from the holistic regularity, while a change point is a point from which the holistic regularity changes. Although outlier and change are two different concepts, they can be unified based on a probabilistic model. When a data stream is modeled as a time series with some probabilistic structure, both outliers and changes can be defined by the variation of statistical regularity, and the only difference is the variation kind. In [4], a unifying framework for mining outliers and changes was proposed, but the two issues were dealt with at two different stages. In [5], we have developed this work into a one-stage framework based on the forward and backward predictions. However, these two methods need pre-selected parametric models, and the parameters must be estimated adaptively in real time implementation. These will increase the difficulty in application, and are the drawbacks of all the parametric methods.

In this paper, we propose a nonparametric unifying method for online mining outliers and changes from data streams. The data stream herein is modeled as a time series with some probabilistic structure. An outlier is defined as a point with both small forward and backward conditional density, while a change is a point with small forward conditional density and large backward conditional density. In order to measure the possibility of being an outlier and a change point, we define two score functions based on the forward and backward prediction errors. Unlike parametric approaches, all predictions are estimated using the local polynomial fitting technique [6] which does not need parameter estimation, but approximates the predictions by fitting a polynomial using the local data around the testing point. This nonparametric method provides many advantages. For example, there's no need to determine the type of the time series model. The prediction accuracy will not be affected by the parameter estimation error. It is appropriate to both the linear and nonlinear time series, which is difficult for parametric methods.

Approaches proposed in the previous literature often try to give an exact partition among outliers, changes, and normal points. However, exact answers from data streams are often too expensive, and approximate answers are acceptable [3]. So in this paper, fuzzy partition and decision approaches are used to alarm possible outliers and changes. The magnitude of the possibility is visualized by the values of membership functions based on which people can make their own decisions. Thus, we believe that our method will be more effective in online and interactive mining of outliers and changes.

The rest of the paper is organized as follows: In Section 2, we formulate the problem of outlier and change detection, and give formal definition of outlier and change point. We give a brief introduction to the local polynomial fitting technique in Section 3, and present the unifying nonparametric outlier and change detection method in Section 4. Simulation results on several data sets are provided in Section 5 and a section of conclusion follows.

2 Problem Formulation

In this section, we will formulate the problem of outlier and change detection from the statistical point of view. While the term "outliers" or "changes" sounds general and

intuitive, it is far from easy to define them. One natural description is that an outlier is a point largely deviates from the holistic regularity, while a change point is a point from which the holistic regularity changes. Although this description is inexplicit, it suggests something common between outlier and change. The holistic regularity varies at both outlier and change point, and only the type of the variation is different. Therefore, detection of outliers and changes is to find the variations of the regularity and distinguish between the different types. Considering a data stream $\{x_t : t = 1, 2, \cdots\}$, if it is modeled with some probabilistic structure, its conditional probability distribution can be incrementally learned from the data stream every time a datum x_t is input. That means we can learn the statistical regularity of the data stream adaptively and find the variations.

We model the real data stream $\{x_t\}$ as a local stationary time-series. Here, each x_t is regarded as a continuous random variable. We use the notation $p(x_t \mid x_{t-L}^{t-1})$ to represent the conditional density of x_t given by $x_{t-L}, \cdots, x_{t-2}, x_{t-1}$, and call it *forward conditional density*. Similarly, the notation $p(x_t \mid x_{t+1}^{t+L})$ is used to represent the conditional density of x_t given by $x_{t+1}, x_{t+2}, \cdots, x_{t+L}$, and is named *backward conditional density*. Then, the formal definition of outlier and change point is given as follows: an outlier x_t is a point with small $p(x_t \mid x_{t-L}^{t-1})$ and small $p(x_t \mid x_{t+1}^{t+L})$, while a change point x_t is a point with small $p(x_t \mid x_{t-L}^{t-1})$ and large $p(x_t \mid x_{t+1}^{t+L})$. Here, we are only interested in the sudden changes.

Now, some criterions should be selected to measure the possibility of being an outlier and a change point. In many previous literature, parametric time-series model is employed, and the form of the conditional density function is pre-decided. Then, the unknown parameters in the conditional density function can be estimated adaptively from $\{x_t\}$. Two score functions are often used as criterions in the parametric approaches [4], [5]. One is based on logarithmic loss:

$$Score(x_t) = -\log p(x_t \mid x_{t-L}^{t-1}, \theta_{t-1}), \tag{1}$$

where $p(x_t \mid x_{t-L}^{t-1}, \theta_{t-1})$ is the estimated parametric conditional density function at time point $t-1$. Another one is based on quadratic loss:

$$Score(x_t) = (x_t - \hat{x}_t)^2, \tag{2}$$

where, $\hat{x}_t$ denotes the prediction for x_t given $x_{t-L}, \cdots, x_{t-2}, x_{t-1}$ based on the estimated conditional density function as follows:

$$\hat{x}_t = E[x_t \mid x_{t-L}^{t-1}] = \int x p(x \mid x_{t-L}^{t-1}, \theta_{t-1}) dx. \tag{3}$$

However, the estimation for the parametric conditional density function is based on parametric modeling and enough data. In online data mining, only limited data are available for parametric modeling. Thus the modeling biases may arise with high

probability and the detection accuracy will be degraded. Moreover, many data in applications exhibit nonlinear features that require nonlinear models to describe. However, beyond the linear parametric models, there are infinitely many nonlinear forms that can be explored. This would be a daunting task for any analysts to try one model after another. In addition, both of the two score functions only consider the forward conditional density, which can not distinguish between the outliers and changes. Thus, in this paper, we propose two novel score functions based on the forward and backward prediction errors (see Section 4.1), and employ a simpler and effective nonparametric approach, the local polynomial fitting, to calculate the predictions.

3 Local Polynomial Fitting

Local polynomial fitting is a widely used nonparametric technique. It possesses various nice statistical properties [6]. Consider a bivariate sequence $\{(X_t, Y_t): t = 1, \cdots, N\}$ that can be regarded as a realization from a stationary time series. We are interested in estimating Y_t by X_t, and the best estimation of Y_t based on $X_t = x$ is the conditional expectation of Y_t given $X_t = x$. Define a regression function in the following form:

$$m(x) = E(Y_t \mid X_t = x), \tag{4}$$

then Y_t can be expressed as follows:

$$Y_t = m(x) + \sigma(x)\varepsilon_t, \tag{5}$$

where $\sigma^2(x) = Var(Y_t \mid X_t = x)$, and ε_t is a random variable that satisfies $E(\varepsilon_t \mid X_t) = 0$, $Var(\varepsilon_t \mid X_t) = 1$.

Denote an arbitrary value of the regression function by $m(x_0)$. Local polynomial fitting is a method for estimating $m(x_0)$. Since the form of $m(x)$ is not specified, a remote data point from x_0 provides very little information about $m(x_0)$. Hence, we can only use the local data around x_0. Assume that $m(x)$ has the $(p+1)$ derivative at the point x_0. By Taylor's expansion, for x in the local neighborhood of x_0, we have the *local model*:

$$m(x) \approx \sum_{j=0}^{p} \beta_j (x - x_0)^j, \tag{6}$$

where $\beta_j = m^{(j)}(x_0) / j!$ and are called *local parameters*. One can estimate the local parameters by minimizing

$$\sum_{t=1}^{N} \left\{ Y_t - \sum_{j=0}^{p} \beta_j (X_t - x_0)^j \right\}^2 K_h(X_t - x_0). \tag{7}$$

The weight function $K_h(\cdot)$ is defined as $K_h(\cdot) \triangleq K(\cdot/h)/h$, where $K(\cdot)$ is a *kernel function* and h is a window bandwidth controlling the size of the local area. The parameter p is named *fitting order*. Formula (7) means the local parameters are estimated by fitting the local model (6) using the local data in the area $[x_0 - h,\ x_0 + h]$. Since $\hat{m}(x_0) = \hat{\beta}_0$, estimate for $m(x_0)$ is actually the weight least square (WLS) solution to the minimizing problem of (7).

4 Outlier and Change Detection

As mentioned before, both of the two score functions (1) and (2) need parametric estimation of the conditional density function, and can not distinguish between the outliers and changes. So in this section, we define two novel score functions based on the forward and backward prediction errors. These two scores are then used to alarm possible outliers and changes based on fuzzy partition and decision. If the prediction of x_t is regarded as a regression function, it can be calculated by local polynomial fitting without parametric modeling. Thus, the outliers and changes can be detected by nonparametric techniques.

4.1 Forward and Backward Scores

We first consider two bivariate sequences $\{(x_{t-L}, x_{t-L+1}), \cdots, (x_{t-2}, x_{t-1})\}$ and $\{(x_{t+2}, x_{t+1}), \cdots, (x_{t+L}, x_{t+L-1})\}$. Then two regression functions can be defined as the *forward prediction* of x_t which means prediction of x_t given by x_{t-1}:

$$m_f(x) = E(x_t \mid x_{t-1} = x), \tag{8}$$

and the *backward prediction* of x_t which means prediction of x_t given by x_{t+1}:

$$m_b(x) = E(x_t \mid x_{t+1} = x). \tag{9}$$

Similar local models can be defined as (6), and *forward and backward local parameters* can be defined as

$$\beta_j^{(f)} = m_f^{(j)}(x_{t-1})/j!, \text{ and } \beta_j^{(b)} = m_b^{(j)}(x_{t+1})/j!,\ \ j = 0, \cdots, p. \tag{10}$$

Fitting the local models using the forward data $\{x_{t-L}, \cdots, x_{t-2}\}$ and the backward data $\{x_{t+2}, \cdots, x_{t+L}\}$ respectively, estimates for the forward and backward predictions can be obtained:

$$\hat{\beta}^{(f)} = (X_f^T W_f X_f)^{-1} X_f^T W_f y_f, \tag{11}$$

and

$$\hat{\beta}^{(b)} = (X_b^T W_b X_b)^{-1} X_b^T W_b y_b. \tag{12}$$

where

$$y_f = (x_{t-L+1},...,x_{t-1})^T,\ y_b = (x_{t+1},...,x_{t+L-1})^T,$$
$$W_f = diag(K_h(x_{t-L} - x_{t-1}),\cdots,K_h(x_{t-2} - x_{t-1})),$$
$$W_b = diag(K_h(x_{t+2} - x_{t+1}),\cdots,K_h(x_{t+L} - x_{t+1})),$$
$$X_f = \begin{pmatrix} 1 & (x_{t-L} - x_{t-1}) & \cdots & (x_{t-L} - x_{t-1})^p \\ \vdots & \vdots & & \vdots \\ 1 & (x_{t-2} - x_{t-1}) & \cdots & (x_{t-2} - x_{t-1})^p \end{pmatrix},$$
$$X_b = \begin{pmatrix} 1 & (x_{t+2} - x_{t+1}) & \cdots & (x_{t+2} - x_{t+1})^p \\ \vdots & \vdots & & \vdots \\ 1 & (x_{t+L} - x_{t+1}) & \cdots & (x_{t+L} - x_{t+1})^p \end{pmatrix}.$$

Thus the forward and backward predictions of x_t are $\hat{\beta}_0^{(f)}$ and $\hat{\beta}_0^{(b)}$, based on which two novel score functions are defined to measure the possibility of being an outlier and a change point. One is *Forward Score*:

$$Score_f(x_t) = (x_t - \hat{\beta}_0^{(f)})^2 \big/ \hat{\sigma}_f^2, \tag{13}$$

another one is *Backward Score*:

$$Score_b(x_t) = (x_t - \hat{\beta}_0^{(b)})^2 \big/ \hat{\sigma}_b^2. \tag{14}$$

Where $\hat{\sigma}_f^2$ is the moment estimate for the variance of the forward data $\{x_{t-L},\cdots,x_{t-2}\}$, and $\hat{\sigma}_b^2$ is the moment estimate for the variance of the backward data $\{x_{t+2},\cdots,x_{t+L}\}$. Dividing by the estimated variance is to make the scores more adaptive to the data stream with varying variance.

Predictions based on local polynomial fitting do not need pre-selected parametric models, and can be adjusted to both the linear and nonlinear data streams. Furthermore, the window bandwidth h is always small enough to keep the mined outliers outside the local data, which otherwise may degrade the detection performance in parametric methods. So we believe that our method is simpler and effective, and more convenient for implementation.

4.2 Fuzzy Partition and Decision

According to the definition of outlier and change point (see Section 2), an outlier always has both large forward and backward scores, while a change point usually has a large forward score and a small backward score. Here, these characters will be used basing on fuzzy partition and decision theory to distinguish between outliers and change points.

We consider the data set $X \triangleq \{x_t\}$ as a domain, and define four fuzzy sets on it:

$$\begin{aligned} FNormalX &= \{(x_t, \mu FNormalX = S_f(x_t)) \mid x_t \in X\}, \\ BNormalX &= \{(x_t, \mu BNormalX = S_b(x_t)) \mid x_t \in X\}, \\ NotFnormalX &= \{(x_t, \mu NotFnormalX = 1 - S_f(x_t)) \mid x_t \in X\}, \\ NotBnormalX &= \{(x_t, \mu NotBnormalX = 1 - S_b(x_t)) \mid x_t \in X\}, \end{aligned} \tag{15}$$

where $S_f(x_t) \triangleq S(Score_f(x_t))$, $S_b(x_t) \triangleq S(Score_b(x_t))$, and

$$S(x) = \begin{cases} 1, & x \le a \\ 1 - 2\left(\dfrac{x-a}{b-a}\right)^2, & a < x \le (a+b)/2 \\ 2\left(\dfrac{b-x}{b-a}\right)^2, & (a+b)/2 < x \le b \\ 0, & x > b \end{cases}. \tag{16}$$

The parameters a, b in (16) are two predefined constants that are used to control the value of the membership functions.

Then, we define two fuzzy sets named as *Outlier* and *Change* respectively as

$$\begin{aligned} Outlier &= NotFnormalX \cap NotBnormalX, \\ Change &= NotFnormalX \cap BNormalX. \end{aligned} \tag{17}$$

Their membership functions are

$$\begin{aligned} \mu Outlier &= \min(\mu NotFnormalX, \mu NotBnormalX), \\ \mu Change &= \min(\mu NotFnormalX, \mu BNormalX). \end{aligned} \tag{18}$$

Finally, point x_t with high value of $\mu Outlier$ is highly probably an outlier, while point x_t with high value of $\mu Change$ is highly probably a change point.

Note that there is another character of a change point x_t. That is x_{t-1} often has a small forward score and a large backward score. Hence, if one wants to reduce the false alarm rate, he can add another four fuzzy sets:

$$\begin{aligned} PFNormalX &= \{(x_t, \mu PFNormalX = S_f(x_{t-1})) \mid x_t \in X\}, \\ PBNormalX &= \{(x_t, \mu PBNormalX = S_b(x_{t-1})) \mid x_t \in X\}, \\ NotPFnormalX &= \{(x_t, \mu NotPFnormalX = 1 - S_f(x_{t-1})) \mid x_t \in X\}, \\ NotPBnormalX &= \{(x_t, \mu NotPBnormalX = 1 - S_b(x_{t-1})) \mid x_t \in X\}, \end{aligned} \tag{19}$$

Then the data set *Change* can be revised to

$$Change = PFNormalX \cap NotPBnormalX \cap NotFnormalX \cap BNormalX, \tag{20}$$

and its membership function is

$$\mu Change = \min(\mu PFNormalX, \mu NotPBnormalX, \mu NotFnormalX, \mu BNormalX). \quad (21)$$

The possibility of being an outlier or a change can be visualized by the values of the membership functions. Analysts can set a threshold to alarm possible outliers and changes. Users can also make their own decisions according to the membership functions and the practical experience. So we believe that our method which synthesizes both statistical and fuzzy approaches will be more effective in interactive online mining of outliers and changes.

4.3 Parameter Selection

In the proposed detection method, some parameters are essential to the detection performance, such as the bandwidth h of the weight function, and the fitting order p.

It is shown in [6] that, for all choices of p, the optimal kernel function is *Epanechnikov kernel* which is $K(z) = \frac{3}{4}(1-z^2)_+$. Nevertheless, some other kernels have comparable efficiency for practical use of p. Hence, the choice of the kernel function is not critical.

Selection of the bandwidth h is important for the detection performance. Too large bandwidth will result in large estimated bias, while too small bandwidth will results in large estimated variance. A basic idea for searching the optimal bandwidth is to minimize the estimated mean integrated square error (MISE) which is defined as

$$h_{opt} = \arg\min_h \int \{(Bias(\hat{m}(x)))^2 + Var(\hat{m}(x))\} dx \quad (22)$$

However, the solution of (22) is too complex for practical use. In this paper, we employ a more convenient method to find a suboptimal bandwidth. First, we set an acceptable threshold of the MISE denoted by δ and an initial value of h, which is $h = [X^f_{\max} - X^f_{\min})]/(L-1)$ for forward parameter estimation, and $h = [X^b_{\max} - X^b_{\min})]/(L-1)$ for backward parameter estimation. Here, $X^f_{\max} \triangleq \max(x_{t-L}, \cdots, x_{t-2})$, and $X^f_{\min} \triangleq \min(x_{t-L}, \cdots, x_{t-2})$. The $X^b_{\max}$ and $X^b_{\min}$ are defined similarly. If $MISE(h) > \delta$, then multiply h by an expanding factor $C > 1$, i.e. $h = Ch$, until it satisfies $MISE(h) \le \delta$. An advisable value of C is 1.1. This searching algorithm can find a reasonable h quickly.

From the analysis in [6], we know that local polynomial fitting with odd order is better than that with even order. Increasing fitting order will increase computational complexity. So we set $p = 1$ for most cases and add it to 3 if necessary.

5 Simulations

We evaluate our methods by numerical simulations using different data sets.

Case (1). The first data set is generated from an AR(2) model:

$$x_t = a_1 x_{t-1} + a_2 x_{t-2} + e_t, \quad (23)$$

where, e_t is a Gaussian random variable with mean 0 and variance 1, and $a_1 = 0.6, a_2 = -0.5$. The data length is 10000. The mean of data changes at time $t = 1000\Delta\tau + 1$ ($\Delta\tau = 1, 2, \cdots, 9$) with change size $\Delta x = 10 - \Delta\tau$. Outliers occur at time $t = 1000\Delta\tau + 501$ ($\Delta\tau = 0, 1, \cdots, 9$) with deviation size $\Delta x = 10 - 0.8(\Delta\tau + 1)$. Fig.1 (a) shows the data set 1 and the membership functions of the fuzzy sets *Outlier* and *Change* at different time points. Here, we set $a = 8$, $b = 30$. As shown in the figure, the outliers and changes can be distinguished and detected simultaneously if the size is not very small.

Fig.1 (b) shows false alarm rate versus effective alarm rate of the outlier detection for data set 1. The effective alarm points are defined as the points during the area $[t^* - 10, t^* + 10]$ where t^* is the true non-normal point. Three different detection methods are compared. They are the proposed method, the CF method proposed in [4], and the parametric method proposed in [5] which is denoted by CIS method. We test the outlier of size 2.8 at time t=8501 for 1000 independent runs. It is observed that for the linear data stream with changing mean and constant variance, the proposed method performs comparably to the other two parametric methods.

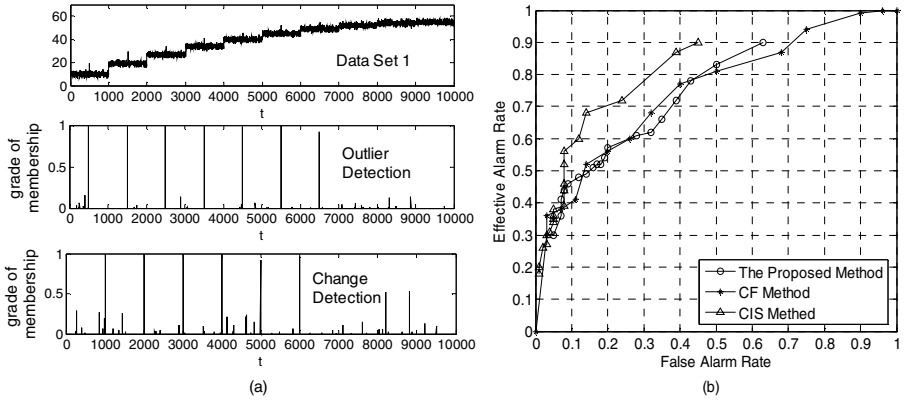


Fig. 1. Outlier and change detection for data set 1. (a) shows the data set 1, and the membership functions of *Outlier* and *Change*. (b) shows the false alarm rate vs. the effective alarm rate of outlier detection for data set 1.

Case (2). In this case, we use the similar AR(2) model as data set 1. The only difference is the variance of e_t varies gradually: $\sigma_e^2(t) = 0.1/[0.01 + (10000 - t)/10000]$. Changes and outliers occur at the same time points as data set 1, but all with size 1. The second data set and the membership functions of *Outlier* and *Change* are given in Fig.2 (a). Here we set $a = 25$, $b = 60$. Similar as the case (1), Fig.2 (b) shows false alarm rate versus effective alarm rate of the change detection for data set 2. Here, we testing the change point of size 5 at time t=5001. Comparing Fig.1 and Fig.2, we can see the advantage of the proposed score functions. Because of dividing by the estimated variance, the influence of the slow varying variance has been decreased a lot. That's why the proposed method outperforms the other two parametric methods in this case.

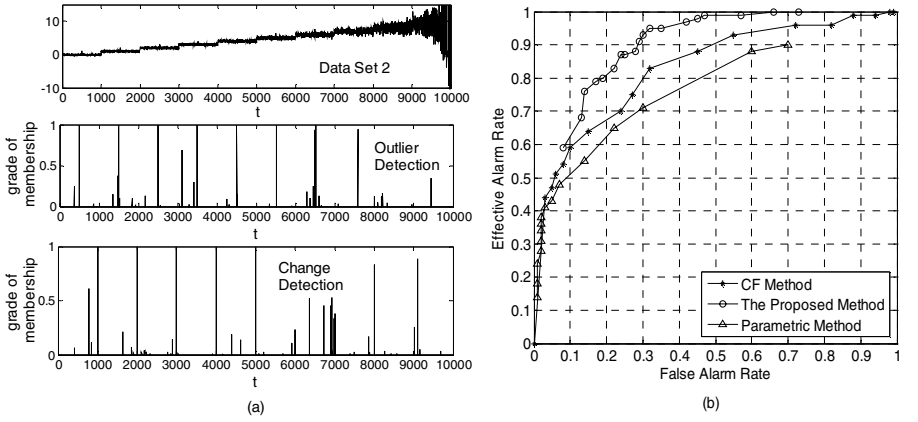


Fig. 2. Outlier and change detection for data set 2. (a) shows the data set 2, and the membership functions of *Outlier* and *Change*. (b) shows the false alarm rate vs. the effective alarm rate of change detection for data set 2.

Case (3). In this case, we change the AR(2) model to a nonlinear time series model, the ARCH(1) model:

$$X_t = \sigma_t e_t, \text{ and } \sigma_t^2 = c_0 + b_1 X_{t-1}^2. \tag{24}$$

where $e_t \sim N(0,1)$, $c_0 = 0.5$ and $b_1 = 0.5$. The mean of data also changes at $t = 1000\Delta\tau + 1$ $(\Delta\tau = 1, 2, \cdots, 9)$ with size $\Delta x = 10 - \Delta\tau$. Outliers occur at time $t = 1000\Delta\tau + 501$ $(\Delta\tau = 0, 1, \cdots, 9)$ with deviation size 7. Fig.3 (a) shows the third data set and the membership functions of *Outlier* and *Change*. Curves of false alarm rate versus effective alarm rate of outlier and change detection for data set 3 are shown in Fig.3 (b). Here, we test the outlier of size 7 at time t=1501, and the change

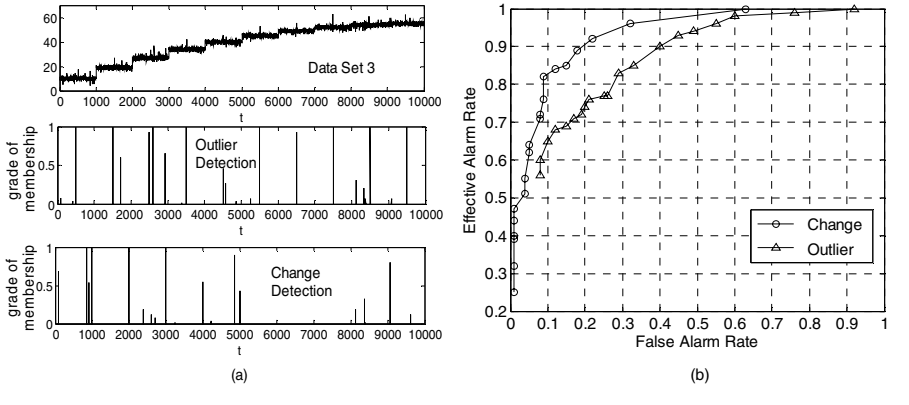


Fig. 3. Outlier and change detection for data set 3. (a) shows the data set 3, and the membership functions of *Outlier* and *Change*. (b) shows the false alarm rate vs. the effective alarm rate of outlier and change detection.

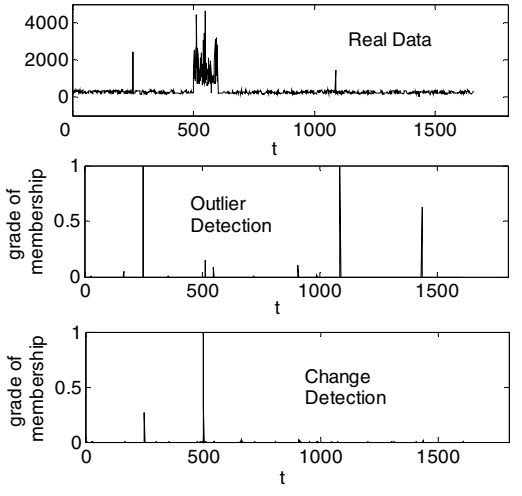


Fig. 4. Outlier and change detection for real data

point of size 3 at time t=7001. It is easy to see the proposed nonparametric detection method is also appropriate to the nonlinear data streams, which is difficult for the parametric methods.

Case (4). The real data case. Here, we test our method by a real data set sampling from the dataset KDD Cup 1999 which is prepared for network intrusion detection. There are 3 intrusions in this real data set, respectively at time t=250, t=1087, and t=1434. The mean and variance of the normal data suddenly change at t=501, and recover at t=602. We present the real data set and the membership functions of *Outlier* and *Change* in Fig.4. It is shown that the proposed method is effective in the real data case. The intrusions are detected as outliers, and the sudden change of the normal data is detected as change points.

6 Conclusion

This paper presents a unifying method for outlier and change detection from data streams. Unlike conventional parametric methods, the proposed method is based on a nonparametric technique, the local polynomial fitting. Fuzzy partition and decision method are used to alarm possible outliers and changes. The proposed method is more appropriate to online and interactive data mining. Simulation results reveal its robustness and efficiency.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No.10571127) and the Specialized Research Fund for the Doctoral Program of Higher Education (No.20040610004).

References

1. G. Dong, J. Han, L.V.S. Lakshmanan, J. Pei, H. Wang and P.S. Yu: Online Mining of Changes from Data Streams: Research Problems and Preliminary Results. Proceedings of the ACM SIGMOD Workshop on Management and Processing of Data Streams. ACM, New York (2003) 225-236
2. Zakia Ferdousi and Akira Maeda: Unsupervised Outlier Detection in Time Series Data. Proceedings of ICDEW'06. IEEE Computer Society, Washington, DC (2006) 51-56
3. M. Garofalakis, J. Gehrke, and R. Rastogi.: Querying and mining data streams: You only get one look. Proceedings of SIGMOD'02. ACM, New York (2002) 635-642
4. Jun-ichi Takeuchi and Kenji Yamanishi: A Unifying Framework for Detecting Outliers and Change Points from Time Series. IEEE Transaction on Knowledge and Engineering. IEEE Press, USA (2006) 482-492
5. Zhi Li, Hong Ma, Yongdao Zhou: A Unifying Method for Outier and Change Detection from Data Streams. Proceedings of CIS'06. IEEE Press, USA (2006) 580-585
6. Jianqing Fan and Qiwei Yao: Nonlinear Time Series: Nonparametric and Parametric Methods. Springer-Verlag, New York (2002) 215-246.
7. H. D. Cheng, Y. H. Chen and X. H. Jiang: Unsupervised Change Detection Using Fuzzy Entropy Principle. IEEE International Geoscience and Remote Sensing Symposium. IEEE Press, USA (2004) 2550-2553

Simultaneous Tuning of Hyperparameter and Parameter for Support Vector Machines

Shizhong Liao and Lei Jia

School of Computer Science and Technology
Institute of Knowledge Science and Engineering
Tianjin University, Tianjin 300072, P. R. China
szliao@tju.edu.cn, ljia@tju.edu.cn

Abstract. Automatic tuning of hyperparameter and parameter is an essential ingredient and important process for learning and applying Support Vector Machines (SVM). Previous tuning methods choose hyperparameter and parameter separately in different iteration processes, and usually search exhaustively in parameter spaces. In this paper we propose and implement a new tuning algorithm that chooses hyperparameter and parameter for SVM simultaneously and search the parameter space efficiently with a deliberate initialization of a pair of starting points. First we derive an approximate but effective radius margin bound for soft margin SVM. Then we combine multiparameters of SVM into one vector, converting the two separate tuning processes into one optimization problem. Further we discuss the implementation issue about the new tuning algorithm, and that of choosing initial points for iteration. Finally we compare the new tuning algorithm with old gradient based method and cross validation on five benchmark data sets. The experimental results demonstrate that the new tuning algorithm is effective, and usually outperforms those classical tuning algorithms.

Keywords: Support Vector Machines; Model Selection; Radius Margin Bound; Tuning Algorithm.

1 Introduction

Choosing hyperparameter and parameter is an important and indispensable process for learning and applying Support Vector Machines (SVM) [1,2,3]. The common tuning strategies, like cross validation and grid search, which searched in the hyperparameter space exhaustively, would become intractable, because these strategies attempted to run the algorithm on every possible value of the hyperparameter vector. Some researchers explored the possibilities of meta learning and evolutionary learning approaches [4,5,6]. Chapelle proposed a gradient descent approach to choosing hyperparameter [7], which reduced the numbers of searching steps drastically. Keerthi studied the implementation issue of this gradient based strategy in [8].

Let θ be the hyperparameter vector, and δ denote some estimation of SVM, such as single validation estimate, support vector count, Jaakkola-Haussler bound, Opper-Winther bound, radius margin bound, and span bound. The general framework for previous tuning methods could be summarized as follows:

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 162–172, 2007.

1. *Initialize* θ,
2. *Solve the optimization problem:* $g(\theta) = \min_\delta G(\theta, \delta)$,
3. *Update hyperparameter* θ *with certain strategy,*
4. *Go to step 2 unless termination condition is satisfied.*

There are two nested optimization problems in the framework. Whatever the strategy is, δ must be calculated in step 2 before θ is updated. This is the disadvantage that limits the convergence speed.

To address the problem, we propose a new framework that combines steps 2 and 3 into one:

1. *Initialize* X,
2. *Compute* $f(X)$,
3. *Update* X *with certain strategy,*
4. *Go to step 2 unless terminating condition is satisfied,*

where $X = (\theta^T, \alpha^T)^T$, θ and α are hyperparameter and *Lagrange* multiplier of soft margin SVM respectively. Then θ and $\delta(\alpha)$ can be updated in one iteration, and the optimal classifier and hyperparameter can be obtained simultaneously.

The paper is organized as follows. In Section 2 we lay the theoretical foundation for the new tuning framework. We derive a new formula of soft margin SVM, and obtain an approximate but effective radius margin bound. Then the new tuning framework can be constructed based on these results. In Section 3 we address the implementation issue. We design an algorithm by combining Sequential Unconstrained Minimization Technique and Variable Metric Method [9,10], and describe initialization problem of starting points for iteration. In Section 4 we present experimental results of the new tuning algorithm compared with other usual approaches on a variety of databases. We end in Section 5 with conclusions and future works.

2 Constructing New Tuning Model

2.1 New Formula of Soft Margin SVM

The soft margin SVM can be described as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i, \\ s.t. \quad & y_i(w^T \Phi(x_i) + b) \geq 1 - \xi_i,\ \xi_i > 0, \end{aligned}$$

where ξ_i represents the training error and C called penalty parameter adjusts the error. Let $\tilde{z} = \left(\Phi(x_i)^T, e_i^T / \sqrt{C}\right)^T$, in which e_i is a 0-vector of l length except the *ith* component is 1, then the soft margin SVM problem can be converted to the standard SVM form:

$$\begin{aligned} \min \quad & \frac{1}{2} \tilde{w}^T \tilde{w}, \\ s.t. \quad & y_i(\tilde{w}^T \tilde{z}_i + b) \geq 1, \end{aligned}$$

where $\tilde{w} = (w^T, \sqrt{C}\xi)^T$. Further more, from the solution of the dual problem we can get

$$\tilde{w} = \sum_i \alpha_i y_i \Phi(\tilde{z}_i),$$

and let $I = \begin{cases} 1: & i = j \\ 0: & i \neq j \end{cases}$, $\tilde{K}$ can be expressed by K:

$$\tilde{K}(x_i, x_j) = K + I/C.$$

The dual problem becomes to

$$\begin{aligned} &\min \ \tilde{L}_D = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (K + \frac{I}{C}) - \sum_i \alpha_i, \\ &s.t. \ \sum_i \alpha_i y_i = 0, \ \alpha_i \geqslant 0. \end{aligned} \tag{2.1}$$

2.2 Calculating Radius Margin Bound

SVM is based on the Statistical Learning Theory (SLT) [11], its generalization bound is $R^2 \|w\|^2$, where R is the radius of the hypersphere that contains all training points. According to Burges [12],

$$R = \sum_{i,j} \beta_i \beta_j K(x_i, x_j) - 2 \sum_{i,j} \beta_i K(x_i, x_j) + \sum_i K(x_i, x_i). \tag{2.2}$$

That is, R is determined by kernel function as well as margin Δ. It is not hard to get the theorem bellow.

Theorem 1. *The generalization bound is smooth in σ. (Cristianini,1998. See [13])*

Formula 2.2 is not efficient in practice, as it requires expensive matrix operations. We need a new expression of R^2. It is obviously that the diameter of the hypersphere that encloses all the samples in feature space is determined almost by the farthest two sample points. So let D be the diameter [1], $D^2 = \left\|\Phi(x_p) - \Phi(x_q)\right\|^2$, where $\{p, q\} = \underset{\{i,j\} \ i,j \in l}{\operatorname{argmax}} \left\|\Phi(x_i) - \Phi(x_j)\right\|^2$.

Theorem 2. *$\{p, q\}$ can be calculated in the input space when the RBF kernel is a monotone function of $\|x_i - x_j\|$.*

Proof. Since

$$\left\|\Phi(x_i) - \Phi(x_j)\right\|^2 = 2r(0) - 2K(x_i, x_j),$$

[1] These expressions of D and R^2 are not precise. There is a tradeoff between computational complexity and learning accuracy. You could calculate them accurately using 2.2. We will discuss the issue deeply in Section 4.

$\{p, q\}$ can be calculated according to the monotonicity of $K(x_i, x_j)$ in the input space:

$$\{p, q\} = \begin{cases} \underset{\{i,j\}\ i,j \in l}{\operatorname{argmin}} \|x_i - x_j\|^2 & : \quad \text{increasing}, \\ \underset{\{i,j\}\ i,j \in l}{\operatorname{argmax}} \|x_i - x_j\|^2 & : \quad \text{decreasing}. \end{cases}$$

□

Then $R^2 = (r(0) - K(x_p, x_q))/2$, if *Gauss* kernel is used, we have

$$R^2 = \frac{1 - K(x_p, x_q)}{2}. \tag{2.3}$$

R^2 is determined only by kernel function, needing not to know the feature map $\Phi : \mathbb{R}^n \to \mathbb{R}^m$, $m > n$. Obviously, formula 2.3 is more efficient than formula 2.2, as p, q usually come from different classes, the computation overheads can be cut down a half on average.

In soft margin SVM, $\tilde{K} = K + I/C$, then

$$\tilde{R}^2 = \frac{\left(1 - K_{pq}\right)}{2} + \frac{1}{2C},$$

i.e.

$$\tilde{R}^2 = R^2 + \frac{1}{2C}. \tag{2.4}$$

The radius margin bound changes to

$$(1 - K_{pq} + \frac{1}{C}) \|\tilde{w}\|^2 . \tag{2.5}$$

It is easier to compute than the original one. Shölkopf discussed the relationship between kernel and the bound in [14], and Chung has proved some properties of this bound in [15].

2.3 The New Tuning Approach

From formula 2.5 we have:

$$\begin{aligned} &\min \quad f = \tilde{R}^2 \times \tilde{L}_D, \\ &s.t. \quad \sum_i \alpha_i y_i = 0,\ \alpha_i \geqslant 0, C > 0, \sigma^2 > 0. \end{aligned}$$

Combining C, σ^2, and α into one vector $X = (C, \sigma^2, \alpha^T)^T$ and constructing vector $\tilde{Y} = (0, 0, Y^T)^T$, where $Y = (y_1, \cdots, y_l)^T$, we can describe the new tuning approach as follows:

$$\begin{aligned} &\min \quad f(X) = \left(1 - K_{pq} + \frac{1}{C}\right)\left[\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j (K + \frac{I}{C}) - \sum_i \alpha_i\right], \\ &s.t. \quad X^T \tilde{Y} = 0,\ X \geqslant 0. \end{aligned} \tag{2.6}$$

The approach is derived from the transformed radius margin bound given by formula 2.5. It is an augmented radius margin bound method that takes into account both the radius and the margin. Since there is only one optimization process in this approach, we can obtain the optimal classifier and the hyperparameter simultaneously.

3 Implementation

In this section we address the implementation issue of the new tuning approach.

3.1 A Synthetic Schema

To solve formula 2.6, we first transform it to an unconstrained minimization problem with SUMT (Sequential Unconstrained Minimization Technique)[9], then search for X^* with a gradient descent based approach VMM (Variable Metric Method)[10].

Given a constrained minimization problem:

$$\begin{array}{ll} \min & f(X), \\ s.t. & g_i(X) = 0, \quad i = 1, \cdots, m, \\ & g_j(X) \geqslant 0, \quad j = 1, \cdots, p, \end{array}$$

SUMT reconstructs a new object function without constrain:

$$J(X, r^k) = f(X) + \frac{1}{r^k} \sum_{i=1}^{m} g_i(X)^2 + r^k \sum_{j=1}^{p} \frac{1}{g_j(X)},$$

where $r^0 > r^1 > r^2 > \cdots > r^k > \cdots > 0$, $r^k \to 0$, $\frac{1}{r^k} \to \infty$.

Accordingly formula 2.6 is converted to

$$\begin{aligned} \min J(X, r^k) = & \left(1 - K_{pq} + \frac{1}{C}\right) \left[\frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \left(K + \frac{I}{C}\right) - \sum_i \alpha_i\right] \\ & + \frac{1}{r^k} \left(X^T \tilde{Y}\right)^2 + r^k \sum_i \frac{1}{e_i X}. \end{aligned} \tag{3.1}$$

The solution of formula 3.1 viz. X^* is also the solution of formula 2.6 [9].

Chapelle et al. present a very useful result about calculating gradient and searching direction in [7](see Lemma 2). The details can be found in Appendix.

3.2 Choosing Start Points

Here we also take *Gauss* kernel $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/2\sigma^2)$ for discussion.

Theorem 3. *All training points become support vectors as* $\sigma \to 0$.

Proof. When $\sigma \to 0$,

$$\lim_{\sigma \to 0} K\left(x_i, x_j\right) = \begin{cases} 1: & x_i = x_j, \\ 0: & x_i \neq x_j. \end{cases}$$

Suppose the training set has l samples, in which the number of samples whose label are $y_i = +1$ or $y_i = -1$ is l^+ or l^- respectively. Assume the *Lagrange* multiplier λ_i is

$$\lambda_i = \begin{cases} \lambda^+: & y_i = +1 \\ \lambda^-: & y_i = -1 \end{cases} \quad 0 < \lambda^+, \lambda^- < C.$$

Because $\forall i \in l\ \lambda_i > 0$, all the sample points become support vectors. Below we want to find a solution to λ_i. From the dual problem solution of SVM we have

$$\sum_{i=1}^{l} \lambda_i y_i = 0,$$

hence,

$$\lambda^+ l^+ - \lambda^- l^- = 0. \tag{3.2}$$

From the KKT condition we get

$$\lambda_i \left[y_i \left(w^T \Phi(x_i) + b \right) - 1 \right] = 0 \text{ and } w^T \Phi(x_i) + b = y_i,$$

which can be transformed to

$$\sum_{j=1}^{l} \lambda_j y_j K\left(x_j, x_i\right) + b = y_i, \quad i = 1, 2, \ldots, n.$$

Let $\sigma \to 0$ we get the simultaneous equations:

$$\begin{cases} \lambda^+ + b = 1, \\ -\lambda^- + b = -1. \end{cases} \tag{3.3}$$

Combining with formula 3.2, we can finally obtain the value of λ_i and b:

$$\lambda^+ = 2l^-/l, \quad \lambda^- = 2l^+/l, \quad b = (l^+ - l^-)/l.$$

Let $C > \max\{2l^-/l, 2l^+/l\}$, the claim holds. □

Theorem 4. *SVM ranks all the test samples to the same class when* $\sigma \to \infty$.

Proof. When $\sigma \to \infty, \lim_{\sigma \to \infty} K(x_i, x_j) = 1$, the determine function is

$$f(x) = \sum_{i=1}^{l} \lambda_i y_i K(x_i, x) + b = b.$$

□

Therefore, the proper value of σ must be neither too large nor too small. See Figure 1. It shows the relationship between classification accuracy and σ. We can see clearly that when $\sigma \to 0$ the training rate is close to 100%, and the testing rate is close to a constant G:

$$G = \begin{cases} l'^+/l' : & b > 0, \\ l'^-/l' : & b < 0. \end{cases}$$

When $\sigma \gg \|x_i - x_j\|$ the testing rate also converges to G. Our experiment reveals that the recommended region of σ is $(0, \|x_i - x_j\|_{\max})$. We prefer $X'^0 = (C^0, \sigma^{2^0}, \alpha^0, r^0)$ starting at $(1, 1, 0, 100)$. Algorithm 1 is the algorithm for formula 3.1.

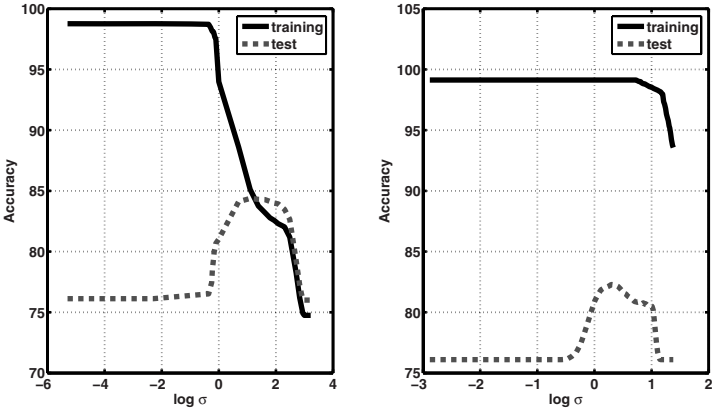


Fig. 1. Relationship between classification accuracy and log–σ

Algorithm 1. SimulTuning

Initialize $X'^0 \in R^n$, $\varepsilon > 0$, $H^0 = I$, $k = 0$.
$\{p, q\} = \underset{\{i,j\}\ i,j \in l}{\operatorname{argmax}} \|x_i - x_j\|^2$.
while $g^k = \nabla J(X'^k) > \varepsilon$ **do**
  $P^k = -H^k g^k$, $\lambda^k = \arg\min_{\lambda} J(X'^k + \lambda P^k)$, $X'^{k+1} = X'^k + \lambda P^k$.
  if $k = n$ **then**
    $X'^0 = X'^n$.
  else
    Compute g^{k+1}, $\Delta X'^k$,Z^k, B^k, C^k, $H^{k+1} = H^k + B^k - C^k$, $k = k + 1$.
  end if
end while
return $X^* = X'^k$.

4 Experiment Results

In our experiments, we assess the convergence properties, number of iteration steps, and classification accuracy, comparing our SimulTuning (Algorithm 1, adopting *Gauss* kernel) with gradient based search and cross validation approaches on five benchmark data sets: *Heart*,*Diabetes*,and *A2a* [16] [2]; *W1a* [17]; and *German.Numer* [18] [3]. Figure 2 shows the track of hyperparameter during searching for the optimal value. We can see clearly that hyperparameter point moves along the ridge line, which ensures the convergence of our algorithm. Figure 3 depicts a comparison of iteration efficiencies, illustrating that our algorithm is superior to 5-fold cross validation in four out of five training data sets. Finally, table 1 lists the experimental results about

[2] Available at http://www.ics.uci.edu/~mlearn/MLRepository.html
[3] Available at http://www.liacc.up.pt/ML/old/statlog/datasets.html

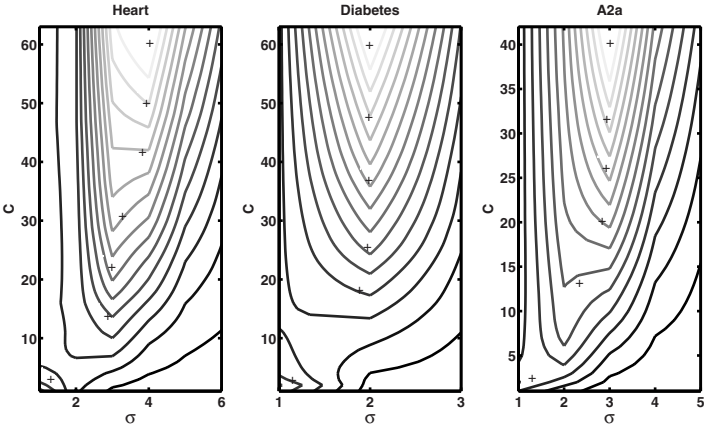


Fig. 2. Isoline of convergence, where '+' denotes the track of (C, σ)

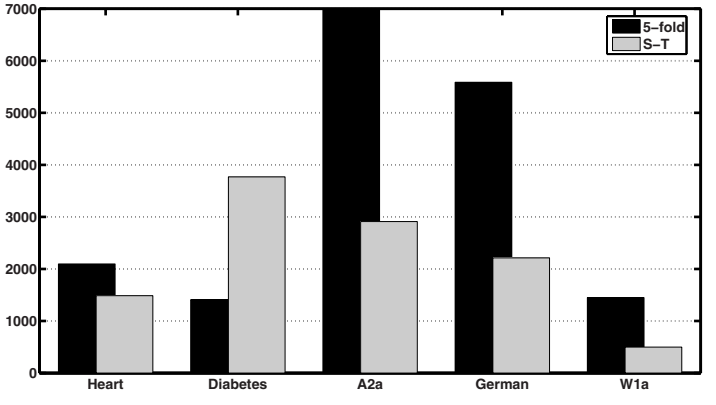


Fig. 3. Compareson of the number of iteration steps. S-T: our SimulTuning, 5-fold: 5-fold cross validation.

Table 1. Classification accuracies of SimulTuning (S-T), gradient based approach (Grad) and cross validation (5-fold). n = dimension of training samples, l = number of training samples, l_{test} = number of test samples, $\log C$ and σ are the optimal value calculated by SimulTuning. We tag the best results in bold.

Attributes / Data Sets	n	l	l_{test}	$J(X'^*)$	$\log C$	σ	Accuracy		
							S-T	Grad	5-fold
A2a	83	2265	30296	176.42	5.6	3.54	**84.470**	83.978	81.766
German	24	300	700	104.2	3	1.5	88.032	**88.113**	75.156
Heart	13	100	170	1.53	6	4.15	**91.852**	89.672	86.741
W1a	299	2477	47272	4.34	2.3	2.67	**97.830**	97.024	97.039
Diabetes	8	300	468	25.9	6	2.1	**80.078**	78.125	76.823

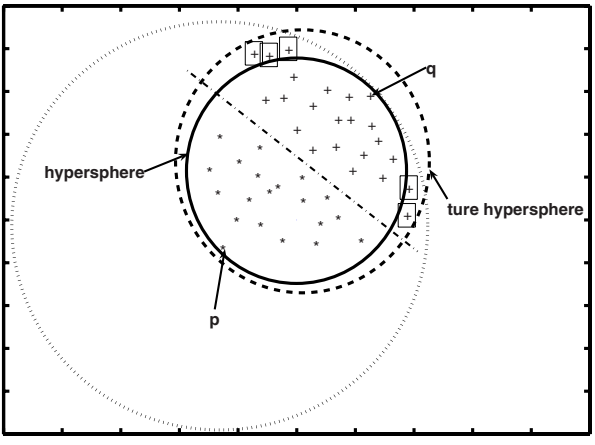


Fig. 4. Distribution pattern of *German*. '*' p and '+' q denote the farthest two points . The boxed '+'s denote samples out of the hypersphere we calculated. The small dashed cycle is the true hypersphere and the large one is the hypersphere centered at p that contains all training samples.

classification accuracies of our algorithm SimulTuning, gradient decent based approach and 5-fold cross validation performed on the five benchmark data sets. The data of accuracies and various attributes demonstrate that our new tuning algorithm SimulTuning surpasses cross validation in all five benchmark tests and gradient based approach in 4 tests.

SimulTuning has a little inferior performance to gradient based approach on *German* data set (88.052:88.113). This phenomenon can be explained as follows. $\tilde{R}^2$ is calculated approximately in SimulTuning, some sample points with a boxed '+' as shown in figure 4 are not enclosed in the hypersphere determined by $\tilde{R}^2$. But this kind of sample distribution pattern of *German* data set is rare, the other four data sets do not have these 'boxed' points.

5 Conclusions and Future Works

It is said an advantage of SVM over ANN is that SVM elegantly separates things to different stages where the innermost one solves simple convex problems. However, this will make sense only when proper hyperparameters are given. In this paper, we present a new approach to tuning hyperparameter and parameter simultaneously for SVM. Experiments on benchmark data sets demonstrate that the tuning algorithm SimulTuning based on this approach outperforms classical approaches without sacrifice of efficiency. Our work also illustrates that combination approach to model selection for SVM is promising.

Although the tuning approach only deals with model selection problem on benchmark data sets and with *Gauss* kernel, the benefits of approximate estimation, the feasibility of combination of hyperparameter and parameter, and the interesting phenomenon relating to *German* data set reveal some focuses of future works. First, a thorough in-

vestigation of approximation in complex sample distribution is needed. Second, combination feasibilities of parameters for other kernels should be demonstrated. Third, attention to heuristic strategies for model selection on large scale data set is deserved.

References

1. Anguita, D., Boni, A., Ridella, S., Rivieccio, F., Sterpi, D.: Theoretical and Practical Model Selection Methods for Support Vector Classifiers. In: Support Vector Machines: Theory and Applications. Springer 71 (2005) 159–181
2. Ayat, N., Cheriet, M., Suen, C.: Automatic model selection for the optimization of the SVM kernels. Pattern Recognition Journal (2005) (Avaliable at http://www.livia.etsmtl.ca/publications/2005/Ayat_pr2005.pdf)
3. Chapelle, O., Vapnik, V.: Model Selection for Support Vector Machines. In: Advances in Neural Information Processing Systems 12. Cambridge, Mass: MIT Press. (1999)
4. Charles, A., Pontil, M.: Learning the kernel function via regularization. Journal of Machine Learing Reasearch **6** (2005) 1099–1125
5. Frauke, F., Christian, I.: Evolutionary tuning of multiple SVM parameters. Neurocomputing **64**(1-4 SPEC ISS) (2005) 107 – 117
6. Soares, C., Brazdil, P.: A meta-learning method to select the kernel width in support vector regression. Machine Learning **54** (2004) 195–209
7. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning **46** (2002) 131–159
8. Keerthi, S.: Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. IEEE Transactions on Neural Networks **13**(5) (2002) 1225–1229
9. McCormick, G.: The projective SUMT method for convex programming. Math.Oper.Res. **14** (1989) 203–223
10. Davidon, W.: Variable metric algorithms for minimization. Technical Report ANL-5990, Argonne National Lab (1959)
11. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
12. Burges, C.: A tutorial on support vector machine for pattern recgnition. Data Mining and Knowledge Discovery **2** (1998) 121–167
13. Cristianini, N., Campbell, C., Shawe-Taylor, J.: Dynamically adapting kernel in support vector machines. Advances in Neural Information Processing Systems **11** (1998) 204–210
14. Shölkopf, B., Shawe-Taylor, J., Smola, A., Willianson, R.: Kernel-dependent support vector error bounds. Artifical Neural Networks **7** (1999) 103–108
15. Chung, K., Kao, W., Sun, C., Lin, C.: Radius margin bounds for support vector machines with RBF kernel. Neural Comput **15** (2003) 2643–2681
16. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI Repository of Machine Learning Databases. Dept. of Information and Computer Sciences, University of California, Irvine (1998)
17. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods - Support Vector Learning. Cambridge, MA, MIT Press. (1998)
18. King, R.: Statlog Databases. Dept. of Statist. Modeling Scie., University of Strathclyde, Glasgow, U.K. (1992)

Appendix

A Computing Gradient

Let $\nabla J(X^k, r^k)$ be the gradient of formula 3.1. It can be computed as follows:

$$\nabla J(X^k, r^k) = \left(\frac{\partial J}{\partial C}, \frac{\partial J}{\partial \sigma^2}, \cdots, \frac{\partial J}{\partial \alpha_i}, \cdots, \frac{\partial J}{\partial r^k}\right)^T,$$

$$\begin{aligned}
\frac{\partial J}{\partial C} &= \tfrac{1}{C^2}\Big(2\tilde{R}^2 \sum_i \alpha_i^2 - \tfrac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j (K + \tfrac{I}{C}) - \sum_i \alpha_i\Big) + r^k, \\
\frac{\partial J}{\partial r^k} &= \sum_i \tfrac{1}{e_i X^k} - \frac{\left(X^{k^T} Y\right)^2}{\left(r^k\right)^2}, \\
\frac{\partial J}{\partial \sigma^2} &= \tfrac{1}{4\sigma^2}\Big[\, \|x_p - x_q\|^2 K_{pq}\Big(\sum_{i,j}\alpha_i\alpha_j y_i y_j (K + \tfrac{I}{C}) - \sum_i \alpha_i\Big) \\
&\quad + 2\tilde{R}^2 \sum_{i,j}\alpha_i\alpha_j y_i y_j \|x_i - x_j\|^2 K \,\Big] + r^k, \\
\frac{\partial J}{\partial \alpha_i} &= 2\tilde{R}^2\Big(\tfrac{1}{2}\sum_j \alpha_j y_i y_j (K + \tfrac{I}{C}) - 1\Big) + \tfrac{y_i}{r^k} + r^k.
\end{aligned}$$

Where $\tilde{R}^2 = \frac{1-K_{pq}}{2} + \frac{1}{2C}$ (cf formula 2.4).

B Computing Searching Direction

Let P^k be the updating direction of $X'^k = (X^{k^T}, r^k)^T$. In the *kth* interaction, let

$$\begin{aligned}
\Delta g^k &= \nabla J(X'^{k+1}) - \nabla J(X'^k), \\
Z^k &= H^k \Delta g^k, \\
\beta^k &= 1/(\Delta X'^k)^T \Delta g^k, \\
\mu^k &= 1/(Z^k)^T \Delta g^k, \\
B^k &= \beta^k \Delta X'^k (\Delta X'^k)^T, \\
C^k &= \mu^k Z^k (Z^k)^T, \\
H^{k+1} &= H^k + B^k - C^k = H^k + \Delta H^k,
\end{aligned}$$

then

$$P^k = -H^k \nabla J\left(X'^k\right).$$

Entropy Regularization, Automatic Model Selection, and Unsupervised Image Segmentation

Zhiwu Lu, Xiaoqing Lu, and Zhiyuan Ye

Institute of Computer Science and Technology, Peking University,
Beijing 100871, China
zhiwu.lu@yahoo.com.cn

Abstract. In region-based image retrieval, the key problem of unsupervised image segmentation is to automatically determine the number of regions for each image in a database. Though we can solve this kind of model selection problem with some statistical criteria such as the minimum description length (MDL) through implementing the EM algorithm, the process of evaluating these criteria may incur a large computational cost. From competitive learning perspective, some more efficient approaches such as rival penalized competitive learning (RPCL) have also been developed for unsupervised image segmentation. However, the segmentation results are not satisfactory and the object of interest may be merged with other regions, since the RPCL algorithm is sensitive to the rival learning rate. In order to solve such problems, we then propose an iterative entropy regularized likelihood (ERL) learning algorithm for unsupervised image segmentation based on the finite mixture model, which can make automatic model selection through introducing entropy regularization into maximum likelihood (ML) estimation. Some segmentation experiments on the Corel image database further demonstrate that the iterative ERL learning algorithm outperforms the MDL based EM (MDL-EM) algorithm and the RPCL algorithm, and leads to some promising results.

1 Introduction

Image segmentation is one of the basic problems of image processing. In general, there are two approaches to do such a task, i.e., region growing [1] and boundary detection [2]. For the region growing approach, each pixel is assigned to one homogeneous region with respect to some features such as gray level, color and texture, while for boundary detection, discontinuity of those features is regarded as an edge and a boundary consists of such edges. In this paper, only the first kind of image segmentation is considered.

Our study on unsupervised image segmentation was motivated by requirements and constraints in the context of image retrieval by content [3,4]. Most approaches use the query-by-example principle, performing queries such as "show

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 173–182, 2007.

me more images that look like this one". However, the user is often more particularly interested in specifying an object (or region) and in retrieving more images with similar objects (or regions), which is opposed to similar images as a whole. Our aim is to allow the user to perform a query on some parts (objects of interest) of an image. In this paper, we focus on the problem of clustering based segmentation of each image in the database to allow partial queries.

Though there have been various clustering methods, such as the EM algorithm [5] for maximum likelihood (ML) [6] and k-means algorithm [7], the number k of clusters in the data set is usually assumed to be pre-known. However, since the image databases for image retrieval are often huge, the prior setting of cluster number for each image is no longer feasible. Such requirement then motivates our interest to the idea of selecting cluster number automatically before or during clustering. Actually, we can solve this model selection problem with some statistical criteria such as the minimum description length (MDL) [8] through implementing the EM algorithm [9], but the process of evaluating these criteria may incur a large computational cost. Some more efficient approaches such as rival penalized competitive learning (RPCL) [10] have also been proposed to make automatical model selection during clustering. Though great improvement can be made as compared with k-means algorithm, the segmentation results are not satisfactory and the object of interest may be merged with other regions, since the RPCL algorithm is sensitive to the rival learning rate.

Under regularization theory [11], we present an iterative algorithm for entropy regularized likelihood (ERL) learning [12,13] to solve such problems, through introducing entropy regularization into ML estimation on finite mixture model for clustering based unsupervised image segmentation. This kind of entropy regularization [14] has already been successfully applied to parameter estimation on mixtures of experts for time series prediction and curve detection, and some promising results have been obtained due to automatic model selection for mixtures of experts. In this paper, we further utilize entropy regularization to make model selection on finite mixture for unsupervised image segmentation, that is, to determine the number of regions of an image automatically.

Finally, we conducted image segmentation experiments to test our algorithm on the Corel image database used as benchmark in [15]. Several experiments have demonstrated that the iterative ERL learning algorithm can automatically select the number of regions for each image in the databases during parameter learning. Moreover, since the object of interest even can be successfully detected from the confusing background, the iterative ERL learning algorithm then performs much better than the MDL based EM (MDL-EM) algorithm and the RPCL algorithm with much less computational cost in the mean time.

2 Entropy Regularization for Automatic Model Selection

We consider the following finite mixture model for cluster analysis:

$$p(x \mid \Theta) = \sum_{l=1}^{k} \alpha_l p(x \mid \theta_l), \sum_{l=1}^{k} \alpha_l = 1, \ \alpha_l \geq 0, \tag{1}$$

where $p(x \mid \theta_l)(l = 1, ..., k)$ are densities from the same parametric family, and k is the number of mixture components.

Given a sample data set $S = \{x_t\}_{t=1}^N$ generated from a finite mixture model with k^* true clusters and $k \geq k^*$, the negative log-likelihood function on the finite mixture model $p(x \mid \Theta)$ is given by

$$L(\Theta) = -\frac{1}{N} \sum_{t=1}^{N} \ln(\sum_{l=1}^{k} (p(x_t \mid \theta_l)\alpha_l)). \tag{2}$$

The well-known ML learning is just implemented by minimizing $L(\Theta)$.

With the posterior probability that x_t arises from the l-th component in the finite mixture

$$P(l \mid x_t) = p(x_t \mid \theta_l)\alpha_l / \sum_{j=1}^{k} p(x_t \mid \theta_j)\alpha_j, \tag{3}$$

we have the discrete Shannon entropy of these posterior probabilities for the sample x_t

$$E(P(l \mid x_t)) = -\sum_{l=1}^{k} P(l \mid x_t) \ln P(l \mid x_t), \tag{4}$$

which can be globally minimized by

$$P(l_0 \mid x_t) = 1, P(l \mid x_t) = 0 (l \neq l_0), \tag{5}$$

that is, the sample x_t is classified into the l_0-th cluster.

When we consider the mean entropy over the sample set S:

$$E(\Theta) = -\frac{1}{N} \sum_{t=1}^{N} \sum_{l=1}^{k} P(l \mid x_t) \ln P(l \mid x_t), \tag{6}$$

all the samples can be classified into some cluster determinedly by minimizing $E(\Theta)$, and some extra clusters are then discarded with mixing proportions reduced to zero.

Hence, the parameter learning on the finite mixture model $p(x \mid \Theta)$ can then be implemented by minimizing the following entropy regularized likelihood function

$$H(\Theta) = L(\Theta) + \gamma E(\Theta), \tag{7}$$

where $\gamma > 0$ is the regularization factor. Here, $E(\Theta)$ is the regularization term which determines the model complexity, and the mixture model can be made as simple as possible by minimizing $E(\Theta)$. Moreover, $L(\Theta)$ is the empirical error of learning on the data set S, and the ML learning by minimizing $L(\Theta)$ is only a special case of the ERL learning with no regularization term.

3 An Iterative Algorithm for Unsupervised Image Segmentation

In this section, we apply the above ERL learning to unsupervised image segmentation via developing an iterative algorithm. For a $N_1 \times N_2$ color image to

be segmented, we consider an 8-dimensional vector consisting of color, texture, and position features for each pixel just the same as [9]. The three color features are the coordinates in the L*a*b* color space, and we smooth these features of the image to avoid over-segmentation arising from local color variations due to texture. The three texture features are contrast, anisotropy, and polarity, which are extracted at an automatically selected scale. The position features are simply the (x, y) position of the pixel, and including the position generally decreases over-segmentation and leads to smoother regions. Finally, we can get a sample set S of $N = N_1 \cdot N_2$ samples for each image in the database.

In the following, we only consider the well-known Gaussian mixture model for unsupervised image segmentation, that is,

$$p(x \mid \theta_l) = \frac{1}{(2\pi)^{n/2}|\Sigma_l|^{1/2}} \exp\{-(1/2)(x - m_l)^T \Sigma_l^{-1}(x - m_l)\}, \tag{8}$$

where n is the dimensionality of x, and $\theta_l = (m_l, \Sigma_l), l = 1, ..., k$ are the mean vectors and covariance matrices of the Gaussian distributions.

We now derive an iterative algorithm to solve the minimum of $H(\Theta)$ as follows. Firstly, we aim to make the above minimum problem without constraint conditions by implementing a substitution: $\alpha_l = exp(\beta_l)/\sum_{j=1}^{k} exp(\beta_j)$, where $-\infty < \beta_l < \infty, l = 1, ..., k$. Using the general methods for matrix derivatives, we are then led to the following series of equations:

$$\frac{\partial H(\Theta)}{\partial m_l} = -\frac{1}{N}\sum_{t=1}^{N} U(l \mid x_t)\Sigma_l^{-1}(x_t - m_l) = 0, \tag{9}$$

$$\frac{\partial H(\Theta)}{\partial \Sigma_l} = -\frac{1}{2N}\sum_{t=1}^{N} U(l \mid x_t)\Sigma_l^{-1}[(x_t - m_l)(x_t - m_l)^T - \Sigma_l]\Sigma_l^{-1} = 0, \tag{10}$$

$$\frac{\partial H(\Theta)}{\partial \beta_l} = -\frac{1}{N}\sum_{t=1}^{N}\sum_{j=1}^{k} U(j \mid x_t)(\delta_{jl} - \alpha_l) = 0, \tag{11}$$

$$U(l \mid x_t) = P(l \mid x_t)(1 + \gamma\sum_{j=1}^{k}(\delta_{jl} - P(j \mid x_t)) \ln(p(x_t \mid \theta_j)\alpha_j)), \tag{12}$$

where δ_{jl} is the Kronecker function. Then, the solution of those equations can be given explicitly as follows:

$$\hat{m}_l = \frac{1}{\sum_{t=1}^{N} U(l \mid x_t)}\sum_{t=1}^{N} U(l \mid x_t)x_t, \tag{13}$$

$$\hat{\Sigma}_l = \frac{1}{\sum_{t=1}^{N} U(l \mid x_t)}\sum_{t=1}^{N} U(l \mid x_t)(x_t - m_l)(x_t - m_l)^T, \tag{14}$$

$$\hat{\alpha}_l = \frac{1}{\sum_{j=1}^{k}\sum_{t=1}^{N} U(j \mid x_t)}\sum_{t=1}^{N} U(l \mid x_t). \tag{15}$$

These explicit expressions give us an iterative algorithm for minimum $H(\Theta)$: during each iteration, we first update P and U according to (3) and (12), respectively, and then update Θ with newly estimated U according to (13)–(15). Hence, this iterative algorithm seems very similar to the EM algorithm on Gaussian mixture. Actually, the iterative ERL learning algorithm just degrades into the EM algorithm when the regularization factor γ is reduced to zero. However, it is different from the EM algorithm in that the mechanism of entropy regularization is implemented on the mixing proportions during the iterations, which leads to the automatic model selection.

Once the iterative ERL learning algorithm has converged to a reasonable solution Θ^*, all the samples (i.e., pixels) from a color image can then be divided into k clusters or regions by

$$C[l] = \{x_t : P(l|x_t) = \max_{j=1,\ldots,k} P(j|x_t)\}. \tag{16}$$

Due to the regularization mechanism introduced in the iteration process, some clusters may be forced to have no samples and then the desired k^*, that is, the true number of regions in an image, can be selected automatically.

As compared with the gradient implementations for the ERL learning in [12,13], the above iterative algorithm has the following two advantages. On one hand, there is no need to select so many parameters for the iterative algorithm, which makes the implementation much more easy. In fact, for the gradient algorithm, we must select an appropriate learning rate on a sample data set, which is generally a difficult task. On the other hand, just like the EM algorithm, the iterative algorithm is generally faster than the gradient algorithm, which is specially appropriate for image processing.

Though we originally introduce entropy regularization into the maximum likelihood estimation (implemented by EM algorithm) for automatic model selection on the Gaussian mixture, it can also be observed that the minimization of the ERL function $H(\Theta)$ is robust with respect to initialization and the drawbacks of EM algorithm may be avoided. That is, when local minima of the negative likelihood $L(\Theta)$ arise during minimizing the ERL function, the average entropy $E(\Theta)$ may still keep large and we can then go across these local minima. Hence, some better segmentation results may be obtained by minimum $H(\Theta)$.

For example, the standard EM algorithm may not escape one type of local minima of the negative likelihood when two or more components in the Gaussian mixture have similar parameters, and then share the same data. As for image segmentation, it means that the object of interest in an image may be split into two or more regions. However, the iterative ERL learning algorithm can promote the competition among these components by minimum $E(\Theta_k)$ as shown in [12], and then only one of them will "win" and the other will be discarded.

4 Experimental Results

We further applied the iterative ERL learning algorithm to unsupervised image segmentation, and also made comparison with MDL-EM algorithm and RPCL

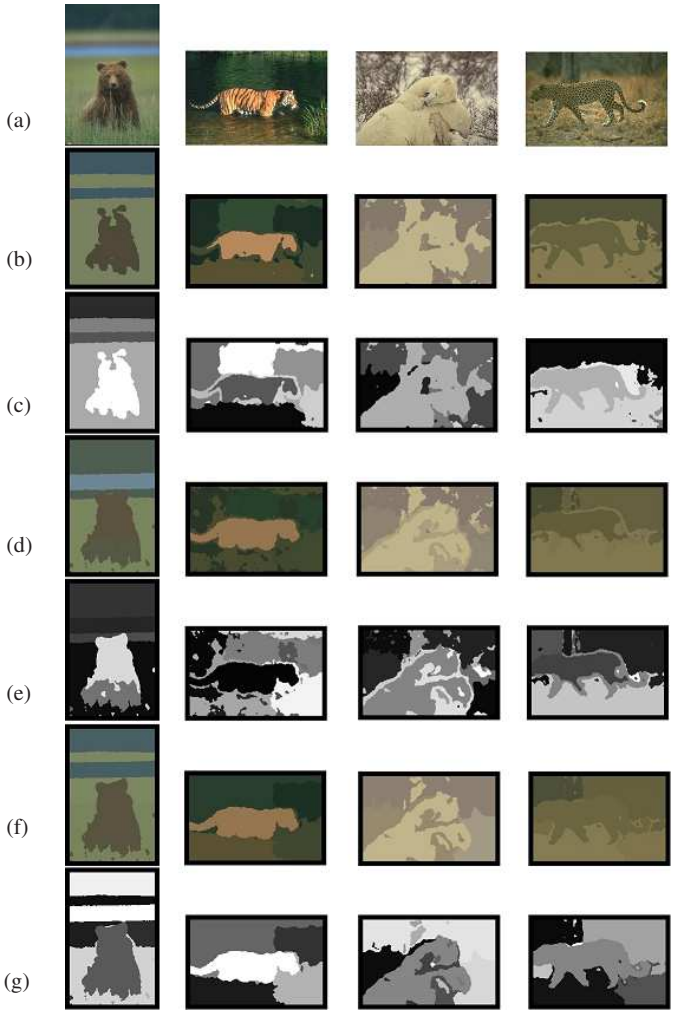


Fig. 1. The segmentation results on one set of randomly selected images by the three learning algorithms: (a) The original images; (b)&(c) The results by the RPCL algorithm; (d)&(e) The results by the MDL-EM algorithm; (f)&(g) The results by the iterative ERL learning algorithm. The gray segments for each algorithm are just the connected components of the color segments.

algorithm on the Corel image database used as benchmark in [15]. We carried out a large number of trials on the database, and only eight images were randomly selected (see Fig. 1(a) and Fig. 2(a)) to show the segmentation results.

In all the segmentation experiments, the parameters of the three learning algorithms can be set as follows. The iterative ERL learning algorithm is always

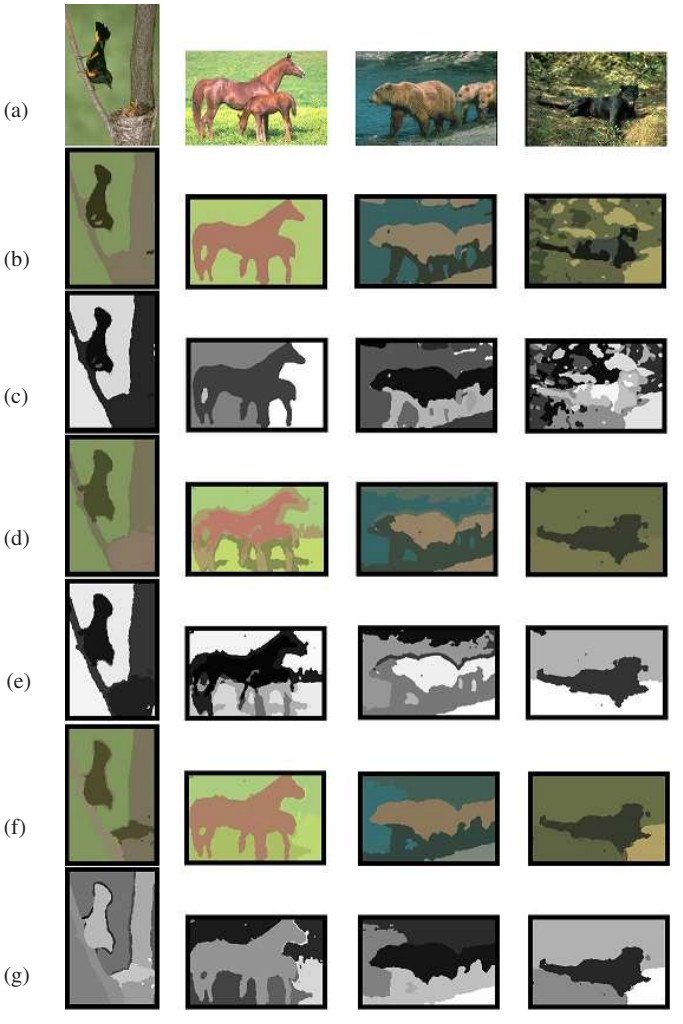


Fig. 2. The segmentation results on another set of randomly selected images by the three learning algorithms: (a) The original images; (b)&(c) The results by the RPCL algorithm; (d)&(e) The results by the MDL-EM algorithm; (f)&(g) The results by the iterative ERL learning algorithm. The gray segments for each algorithm are just the connected components of the color segments.

implemented with $k \geq k^*$ and $\gamma \in [0.2, 0.5]$, while the centers and widths of the Gaussian units are initialized by some clustering algorithms such as the k-means algorithm. In the segmentation, we actually set k a relatively larger value (e.g., $k = 6$), and select γ in the empirical range which is obtained by a large number of segmentation trials. Since we can not adaptively select this model selection parameter for each image in the database, we simply set $\gamma = 0.4$ for all the images uniformly. Moreover, the ERL learning is always stopped

when $|(H(\hat{\Theta}) - H(\Theta))/H(\Theta)| < 10^{-5}$. Just the same as the ERL learning, the RPCL algorithm also fixes k at 6, while the MDL-EM algorithm selects k in the range $[2, 5]$. Finally, the learning rates for winner and rival units during RPCL clustering are set as $\eta_w = 0.05$ and $\eta_r = 0.005$, respectively.

Once a segmentation model is selected after the clustering is stopped, the next step is to perform spatial grouping of those pixels belonging to the same color/texture cluster. We first produce a k^*-level image (i.e., the color vision of segmentation results for each algorithm in Fig. 1 and Fig. 2) which encodes pixel-cluster memberships by replacing each pixel with the label of the cluster for which it attains the highest likelihood, and then run a connected-components algorithm to find image regions (i.e., the gray vision of segmentation results for each algorithm in Fig. 1 and Fig. 2). Note that there may be more than k^* of these regions for each image. Finally, to enforce a minimal amount of spatial smoothness in the final segmentation, we apply a 3×3 maximum-vote filter to the output of each clustering algorithm. This filter assigns its output value as the value that occurs most often in the 3×3 window.

From the segmentation results shown in Fig. 1 and Fig. 2, we can find that the iterative ERL learning algorithm successfully detects the object of interest from the confusing background and performs generally better than the other two algorithms. That is, the MDL-EM algorithm may converge at local minima and the object of interest may be split into two regions (see brown bear and horse) when two or more Gaussian centers are initialized in the region of it, while the RPCL algorithm is sensitive to the rival learning rate η_r and the object of interest may be merged with other regions (see brown bear and sparrow).

Moreover, the average seconds per image taken by the three learning algorithms for segmentation of the eight randomly selected images are also listed in Table 1. Note that we just recorded the computational cost by the clustering for grouping pixels into regions, and the postprocessing of the segmentation results such as searching connected components is not included. In all the segmentations, we process the images by the three learning algorithms offline on a 3.0GHz Pentium IV computer. As expected, the iterative ERL learning algorithm runs much faster than the MDL-EM algorithm since the process of evaluating the MDL criterion incurs a larger computational cost. As compared with the RPCL algorithm, the iterative ERL learning algorithm also keeps more efficient to make unsupervised image segmentation.

The further experiments on the other images in the database have also been made successfully for segmentation in the similar cases. Actually, in many experiments, the iterative ERL learning algorithm can automatically detect the number of regions for a color image in the database and maintain the edges of

Table 1. The average seconds per image taken by the three learning algorithms for segmentation of the randomly selected images

RPCL	MDL-EM	ERL
57.7	80.6	38.5

the object of interest well even in the confusing background. Note that the iterative ERL learning algorithm is just compared with the MDL-EM algorithm, and the comparison results should be the same when some other model selection criteria are taken into account to determine the model scale k for the EM algorithm. Additionally, once the color/texture features are assigned to those connected components of the color images, we can then implement region-based image retrieval on the Corel image database. In the future work, we will evaluate the iterative ERL learning algorithm using the precision-recall measure in the context of image retrieval.

5 Conclusions

We have proposed an iterative ERL learning algorithm for unsupervised image segmentation with application to content based image retrieval. Through introducing a mechanism of entropy regularization into the likelihood learning on the finite mixture model, the iterative ERL learning algorithm can make model selection automatically with a good estimation of the true parameters in the mixture. When applied to unsupervised image segmentation, the iterative ERL learning algorithm can even successfully detect the object of interest from the confusing background, and then performs much better than the MDL based EM (MDL-EM) algorithm and the RPCL algorithm with much less computational cost in the mean time.

References

1. Shih, F.Y., Cheng, S.X.: Automatic Seeded Region Growing for Color Image Segmentation. Image and Vision Computing **23** (10) (2005) 877–886
2. Dai, M., Baylou, P., Humbert, L., Najim, M.: Image Segmentation by a Dynamic Thresholding Using Edge Detection Based on Cascaded Uniform Filters. Signal Processing **52** (1) (1996) 49–63
3. Banerjee, M., Kundu, M.K.: Edge Based Features for Content Based Image Retrieval. Pattern Recognition **36** (11) (2003) 2649–2661
4. Yap, K.-H., Wu, K.: A Soft Relevance Framework in Content-Based Image Retrieval Systems. IEEE Transactions on Circuits and Systems for Video Technology **15** (12) (2005) 1557–1568
5. Render, R.A., Walker, H.F.: Mixture Densities, Maximum Likelihood and the EM Algorithm. SIAM Review **26** (2) (1984) 195–239
6. Govaert, G., Nadif, M.: Comparison of the Mixture and the Classification Maximum Likelihood in Cluster Analysis with Binary Data. Computational Statistics & Data Analysis **23** (1) (1996) 65–81
7. Chinrungrueng, C., Sequin, C.H.: Optimal Adaptive K-means Algorithm with Dynamic Adjustment of Learning Rate. IEEE Transactions on Neural Networks **6** (1) (1995) 157–169
8. Rissanen, J.: Modeling by Shortest Data Description. Automatica **14** (1978) 465–471

9. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (8) (2002) 1026–1038
10. Xu, L., Krzyzak, A., Oja, E.: Rival Penalized Competitive Learning for Clustering Analysis, RBF Net, and Curve Detection. IEEE Transactions on Neural networks **4** (4) (1993) 636–648
11. Dennis, D.C., Finbarr, O.S.: Asymptotic Analysis of Penalized Likelihood and Related Estimators. The Annals of Statistics **18** (6) (1990) 1676–1695
12. Lu, Z.: Entropy Regularized Likelihood Learning on Gaussian Mixture: Two Gradient Implementations for Automatic Model Selection. Neural Processing Letters **25** (1) (2007) 17–30
13. Lu, Z., Ma, J.: A Gradient Entropy Regularized Likelihood Learning Algorithm on Gaussian Mixture with Automatic Model Selection. Lecture Notes in Computer Science 3971 (2006) 464–469
14. Lu, Z.: A Regularized Minimum Cross-Entropy Algorithm on Mixtures of Experts for Time Series Prediction and Curve Detection. Pattern Recognition Letters **27** (9) (2006) 947–955
15. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Agorithms and Measuring Ecological Statistics. In: Proceedings of the 8th International Conference on Computer Vision (2001) 416–423

A Timing Analysis Model for Ontology Evolutions Based on Distributed Environments

Yinglong Ma[1,2], Beihong Jin[2], Yuancheng Li[1], and Kehe Wu[1]

[1] Computer Sciences and Technology Department,
North China Electric Power University, Beijing 102206, P.R. China
[2] Technology Center of Software Engineering, Institute of Software,
Chinese Academy of Sciences, P.O.Box 8718, Beijing 100080, P.R. China
{m_y_long, jbh}@otcaix.iscas.ac.cn

Abstract. In order to reuse and assess ontologies, it is critical for ontology engineers to represent and manage ontology versioning and evolutions. In this paper, we propose a timing analysis model for ontology evolution management with more expressive time constraints in a distributed environment. In the model, a timing change operation sequence is called a timing evolution behavior that must satisfy all of the time constraints in a distributed environment. Using this timing analysis model, we can detect whether ontology evolutions are timing consistent in the distributed environment. Given a timing change operation sequence, we also can detect whether it is a timing evolution behavior of the distributed environment. All of these detections can be reduced to detecting whether the group of inequations has solutions. This enables us to better manage dynamic versioning and evolutions of distributed ontologies. We also developed a prototype system called TEAM that can perform our timing analysis task of distributed ontology evolutions.

1 Introduction

It is an important goal for ontology engineers to reuse knowledge-based systems by building and sharing domain ontologies, which are semantically sound specifications of domain conceptualizations [1]. Especially within Semantic Web environment, many Web ontology languages have been developed such as OWL[1]. Moreover, Semantic Web will not be realized by agreeing on a single global ontology, but rather by weaving together a large collection of partial ontologies that are distributed across the Web [2]. Current research areas had rapidly shifted from applications based on simple ontology to some aspects such as representation, evolutions and management of multiple ontologies in a distributed environment [3]. These aspects can be tackled using mapping, combining and versioning among distributed ontologies [4,5,6].

On one hand, evolution and version management of multiple ontologies in a distributed environment should efficiently represent ontology changes in details. This enables us to trace evolution history of multiple ontologies and assess

[1] www.w3c.org/TR/owl-ref/

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 183–192, 2007.

whether any two ontology versions are based on different conceptualizations or whether they represent the same conceptualization. On the other hand, we also should specify change operations of ontologies and further analyze the different causalities in diverse contexts because of these change operations. Current some ontology versioning approaches such as KAON[2] and Protege[3], cannot jointly address the two aspects [7]. More importantly, considering ontology versioning is changing over time, we need an approach with time constraints for representing ontology versioning and evolution, especially for specifing ontology versioning and evolution with more specific time constraints. For example, the change operation op to ontology o occurs 3 time units earlier than the change operation op' to ontology o'; the change operation op based on ontology o takes place 2 time units, and so on. In our opinion, we urgently need an ontology evolution management approach that not only can specify change details between ontologies and analyze ontology change operations and implications between these operations, but also can represent more expressive time constraints. This paper will work towards this goal.

In this paper, we propose a timing analysis model for ontology evolution management with more expressive time constraints. Using this timing analysis model, we can detect whether ontology evolutions are timing consistent in a distributed environment. Given a timing change operation sequence, we also can detect whether it is a timing evolution behavior of the distributed environment. This enables us to better manage dynamic versioning and evolutions of distributed ontologies.

This paper is organized as follows: Section 2 briefly introduces the motivation of this paper. Section 3 first proposes the timing analysis model of ontology versioning and evolution based on a single context. Then in Section 4, we extend work of Section 3 to a distributed environment. Meanwhile, we analyze some important properties from the timing analysis model. The time analysis model based on a single context is also suitable for distributed environments. Section 5 briefly introduces the architecture and implementation of prototype system called TEAM. Section 6 and Section 7 are the related work and conclusion, respectively.

2 Motivation

In a distributed environment, especially knowledge based system with real-time characteristics, diverse contexts probably own themselves ontologies that are semantically overlapped with each other. In the situation, when an ontology is changed with respect to an ontology change operation in some context, some ontology change operations in other contexts must be triggered to change their ontologies and hence keep semantic consistencies between these distributed ontologies. We use Figure 1 to illustrate our example.

[2] kaon.semanticweb.org
[3] protege.stanford.edu

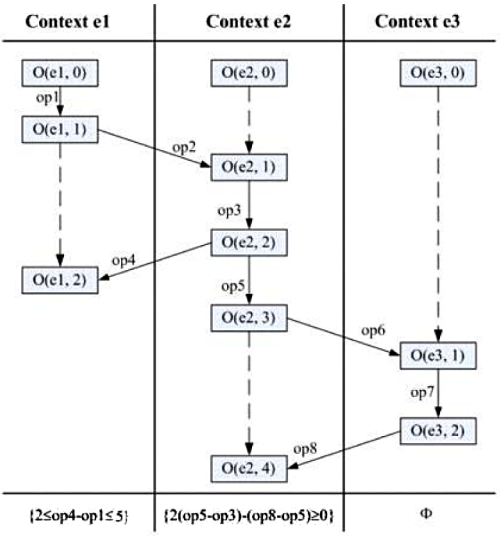


Fig. 1. An example of distributed ontology evolution

In this example, there are three diverse contexts (columns) in the distributed environment. They are denoted $e1$, $e2$ and $e3$, respectively. These distributed contexts own themselves original ontology version. We use $O_{(ei,0)}$ to represent the original ontology from the context ei $(1 \leq i \leq 3)$. $O_{(ei,n)}$ is used for representing the n^{th} ontology version from the context ei. For any context ei $(1 \leq i \leq 3)$, it continually evolves in accord with the time axis from top to bottom. Considering the example in Figure 1, in the context $e1$, because of the ontology change operation $op1$, the original ontology version $O_{(e1,0)}$ is changed to ontology version $O_{(e1,1)}$. Of course, the change probably causes semantic inconsistencies with semantic terms from other contexts. In order to keep semantic consistency among semantic terms from these different contexts, another ontology change operation $op2$ is triggered which revises semantic terms in ontology version $O_{(e2,0)}$ in the context $e2$ to $O_{(e2,1)}$. The eight ontology change operations in the figure have caused version evolutions of multiple ontologies in the whole distributed environment. Compatangelo et al. [7] formally specify three types of operations performed on distributed ontologies using rewriting rules which include *creation of a concept*, *renaming of a concept* and *addition of attribute/value pair to a concept*. In this paper, we will not concentrate on the specific types of these ontology change operations because our time analysis model is independent of specific types of ontology change operations.

The arrows denote the path of ontology changes. The dashed arrows denote the virtual path of ontology changes in a single context. In the bottom of the figure, each context owns a time constraint set which represents some time constraints of the form

$$a \leq c_1(op_1 - op'_1) + c_2(op_2 - op'_2) + \cdots + c_n(op_n - op'_n) \leq b$$

where op_1, op'_1, op_2, op'_2, $\cdots$, op_n, op'_n are ontology change operations, a, b, c_1, c_2, $\cdots$, c_n are real numbers (b may be $\propto$).

In this example, we must address some problems as follows:

1) Given a specific sequence of change operations performed on these distributed environments, how can we detect whether the sequence reflects a correct change behavior of distributed ontology evolutions. That is to say, we need to detect whether the sequence is timing consistent with respect to all the time constraints of the distributed environment.

2) Given all the time constraints of the distributed environment, how will we know whether there exists some timing consistent evolution behavior? That is, we need to know whether there exists a sequence of ontology change operations that satisfies all the time constraints of the whole distributed environment.

In order to specify ontology changes and evolution with exact time constraints in a distributed environment, we must answer the two questions. Regarding to the first question, if the given operation sequence cannot satisfy all the time constraints, then we know that these change operations performed on distributed ontologies are meaningless because the sequence is not a correct change behavior w.r.t all the time constraints. As for the second problem, if there does not exist any sequence σ such that σ satisfies all the time constraints, then we know that the time constraints are unreasonable and they should be reset.

3 Time Analysis Model on Single Context

Definition 1. *In context e, $O_{(e,0)}$ is the original ontology. The set of evolving ontology versions based on context e is denoted as $S_e = \{O_{(e,0)}, O_{(e,1)}, \cdots, O_{(e,n)}\}$.*

Definition 2. *In context e, a timing analysis model for ontologies based on S_e is denoted as $TAM_e = (S_e, OP_e, C_e, VS_e, T)$, where*

- *S_e is the evolving ontology set*
- *OP_e is the set of ontology change operations performed on ontologies*
- *C_e is a set of time constraint marks of the form:*
  $a \leq c_1(op_1 - op'_1) + c_2(op_2 - op'_2) + \cdots + c_n(op_n - op'_n) \leq b$
  where $op_1, op'_1, op_2, op'_2, \cdots, op_n, op'_n \in OP_e$
- *VS_e is the version space based on ontology set S_e and $VS_e \subseteq S_e \times OP_e \times S_e$*
- *T is a linear ordering about change operation triggering $T \subseteq OP_e \times OP_e$ where T is irreflective, transitive, asymmetry and comparable.*

As for the example in Figure 1, we construct TAM_{e2} with respect to the context $e2$, we find that ontology set $S_{e2} = \{O_{(e2,0)}, O_{(e2,1)}, O_{(e2,2)}, O_{(e2,3)}, O_{(e2,4)}\}$. Because of the sequence of change operations from $OP_{e2} = \{op_2, op_3, op_5, op_8\}$, ontologies continually change from one version to another version. The version space $VS_{e2} = \{(O_{(e2,0)}, op_2, O_{(e2,1)}), (O_{(e2,1)}, op_3, O_{(e2,2)}), (O_{(e2,2)}, op_5, O_{(e2,3)}), (O_{(e2,3)}, op_8, O_{(e2,4)})\}$, the time constraint set $C_{e2} = \{0 \leq 2(op5 - op3) - (op8 - op5)\}$. The linear ordering set $T = \{(op2, op3), (op3, op5), (op5, op8)\}$.

We use a change operation sequence for representing and modeling an untiming evolution behavior of ontologies. Any operation sequence is of the form $op_0 - op_1 - \cdots - op_n$, which represents op_{i+1} takes place after op_i for any $0 \leq i \leq n-1$, where $op_0, op_1, \cdots, op_n$ are the change operation names that are triggered because of the causalities of them.

Definition 3. *As for the timing analysis of ontology evolutions based on e, any operation sequence $\sigma = op_0 - op_1 - \cdots - op_n$ is untiming evolution behavior of untiming analysis model TAM_e if and only if the following conditions hold:*

— *all operations op_i in OP_e occur in the sequence σ, and $op_i \neq op_j$ for any i, j $(i \neq j, 0 \leq i, j \leq n)$*

— *for any op_i, op_j such that $(op_i, op_j) \in T$, then $0 \leq i \leq j \leq n$.*

Considering the example in Figure 1, we say the sequence $op_2 - op_3 - op_5 - op_8$ can be regarded as an untiming evolution behavior of ontologies based the context $e2$. It is not difficult to give an algorithm to check if there is an untiming evolution behavior for a given time analysis model. As mentioned above, however, an untiming evolution behavior cannot fully specify dynamic evolution behavior of ontologies. We concentrate on the timing evolution behavior. A timing change operation sequence is of the form $(op_0, t_0) - (op_1, t_1) - \cdots - (op_n, t_n)$, where op_i is a change operation name, and for any i $(0 \leq i \leq n)$, t_i is a nonnegative real number that represents the performed time finishing operation op_i. The sequence represents that op_0 takes place t_0 time units after ontology changes start, op_1 takes place t_1 time units after op_0 takes place, $\cdots$, op_n takes place t_n time units after op_{n-1} takes place. The needed time performing the whole sequence σ is called performed time of σ and denoted as $PT(\sigma)$. The occurrence time of an operation op_i in σ is denoted as $OT(op_i)$. It is obvious to obtain the following lemma.

Lemma 1. *For any timing change sequence $\gamma = (op_0, t_0) - (op_1, t_1) - \cdots - (op_n, t_n)$ of TAM_e, its performing time is denoted as $PT(\gamma) = \Sigma_{j=0}^{n} t_j$ and $OT(op_i) = \Sigma_{j=0}^{i} t_j$ for any i $(0 \leq i \leq n)$.*

Definition 4. *A timing change operation sequence $\gamma = (op_0, t_0) - (op_1, t_1) - \cdots - (op_n, t_n)$ is a timing evolution behavior of the timing analysis model TAM_e if and only if the following conditions hold:*

— *$op_0 - op_1 - \cdots - op_n$ is an untiming evolution behavior of TAM_e, and*

— *the different time units $t_0, t_1, \cdots, t_n$ must satisfy the time constraints in C_e, i.e., for any time constraint $a \leq \Sigma_{i=0}^{n} c_i(f_i - f_i') \leq b$ in C_e such that $a \leq c_0\delta_0 + c_1\delta_1 + \cdots + c_n\delta_n \leq b$, where for each $i(0 \leq i \leq n)$,*

1) if $f_i = op_j$ and $f_i' = op_k$ then if $j \leq k$ then $\delta_i = -(OT(op_k) - OT(op_j))$,

2) otherwise, if $k < j$ then $\delta_i = OT(op_j) - OT(op_k)$

From the previous definitions, we can detect if a timing change operation sequence is a timing evolution behavior of a given time analysis model. That is to say, we can judge whether the sequence correctly reflects versioning and evolution under specific time constraints. We have resolved the first problem put forward in Section 2. For example, we continue to consider the context $e2$ in Figure 1,

a operation sequence $\gamma = (op2, 1) - (op3, 1) - (op5, 1) - (op8, 1)$ is a timing evolution behavior of the given TAM_{e2} because γ satisfies every time constraint in C_{e2}, whereas the sequence $\sigma = (op2, 1) - (op3, 2) - (op5, 1) - (op8, 3)$ cannot be regarded as a timing evolution behavior of TAM_{e2} because it doesn't satisfies the time constraint $\{0 \leq 2(op5 - op3) - (op8 - op5)\}$.

We use the notation $TEBS(TAM_e)$ for representing the set of all timing evolution behaviors which are the change operation sequences satisfying all the time constraints in TAM_e. In order to better represent time operation sequence, we define a special change operation ε which represents the start of ontology changes.

Definition 5. *If there is a change operation sequence* $\gamma = (\varepsilon, 0) - (op_1, t_1) - \cdots - (op_n, t_n)$ *such that* γ *is a timing evolution behavior of* TAM_e*, then we say ontology versioning and evolution based on* TAM_e *is timing consistent.*

Considering the problem put forward in Section 2, we will discuss whether a given TAM_e is timing consistent. We briefly give the following theorem.

Theorem 1. *Ontology versioning and evolution based on* TAM_e *is timing consistent if and only if* $TEBS(TAM_e) \neq \emptyset$.

Proof. According to the previous definitions, we know that $TEBS(TAM_e)$ is the set of all of timing evolution behaviors of the time analysis mode TAM_e based on the context e. Next, according to definition 5, if ontology versioning and evolution based on TAM_e is timing consistent, then there at least exists a timing evolution behavior of TAM_e. Of course, $TEBS(TAM_e) \neq \emptyset$.

If $TEBS(TAM_e) \neq \emptyset$, this means that at least exists a timing evolution behavior of TAM_e, hence we know that ontology versioning and evolution based on TAM_e is timing consistent. □

In the following, we will extend the time analysis approach to a distributed environment, we find that this approach will still work well in diverse contexts.

4 Extending Timing Analysis Model to Distributed Environments

Definition 6. *A timing analysis model base on a distribute environment* ENV *is denoted as* TAM*, which is a six-tuple,* $TAM = (ENV, S, OP, C, VS, T)$*, where*

- $ENV = \{e_1, e_2, \cdots, e_m\}$ *is the set of diverse contexts*
- $S = \bigcup_{e \in ENV} S_e$ *represents the set of all changing ontlogies in* ENV
- $OP = \bigcup_{e \in ENV} OP_e$ *represents the set of all change operations in* ENV
- $C = \bigcup_{e \in ENV} C_e$ *represents the set of all time constraints in* ENV
- VS *is the version space in* ENV*, and* $VS \subseteq S \times OP \times S$
- T *is a linear ordering for representing triggering relations between change operations,* $T \subseteq OP \times OP$*, and* T *is irreflective, transitive, asymmetry and comparable.*

We consider the example in Figure 1. In the distributed environment,
$ENV = \{e1, e2, e3\}$,
$S = \{O_{(e1,0)}, O_{(e1,2)}, O_{(e1,3)}, O_{(e2,0)}, O_{(e2,1)}, O_{(e2,2)}, O_{(e2,3)}, O_{(e2,4)}, O_{(e3,0)}, O_{(e3,1)}, O_{(e3,2)}\}$,
$OP = \{op1, op2, op3, \cdots, op8\}$,
$C = \{2 \leq op4 - op1 \leq 5, 0 \leq 2(op5 - op3) - (op8 - op5)\}$,
$VS = \{(O_{(e1,0)}, op_1, O_{(e1,1)}), (O_{(e1,1)}, op_2, O_{(e2,1)}), (O_{(e2,1)}, op_3, O_{(e2,2)}), (O_{(e2,2)}, op_5, O_{(e2,3)}), (O_{(e2,3)}, op_6, O_{(e3,1)}), (O_{(e3,1)}, op_7, O_{(e3,2)}), (O_{(e3,2)}, op_8, O_{(e2,4)})\}$.
$T = \{(op1, op2), (op2, op3), (op3, op4), (op4, op5), (op6, op7), (op7, op8)\}$.

Definition 7. *Based on the time analysis model TAM in a distributed environment ENV, if there is a change operation sequence $\gamma = (\varepsilon, 0) - (op_1, t_1) - \cdots - (op_n, t_n)$ such that γ is a timing evolution behavior of TAM, then we say ontology versioning and evolution based on TAM is timing consistent.*

From the theorem 1, we can easily obtain the following extended theorem.

Theorem 2. *Ontology versioning and evolution based on TAM is timing consistent if and only if $TEBS(TAM) \neq \emptyset$.*

The theorem gives a framework for detecting whether a given time analysis model is timing consistent based on a distributed environment. It can answer the second problem put forward in Section 2. Although theorem 1 provides a framework for detecting whether a given time analysis model is timing consistent, the framework is difficult to be operated well because we have no any straightforward solution to check whether the set of the timing evolution behaviors is empty. Therefore, we need to deeply exploit a framework and obtain a feasible and well-operated algorithm.

According to the previous definitions, a timing change operation sequence $\sigma = (\varepsilon, 0) - (op_1, t_1) - \cdots - (op_n, t_n)$ is a timing evolution behavior of TAM, then all $t_1, t_2, \cdots, t_n$ in σ must satisfy all time constraints $r_1, r_2, \cdots, r_k$ in the time constraint set C of TAM, where for any $1 \leq i \leq k$,

$$r_i = a_i \leq \Sigma_{j=1}^{n} c_{(i,j)}(f_j - f'_j) \leq b_i \tag{1}$$

If $f_j = op_p$, and $f'_j = op_q$ for any $p \neq q$, and $1 \leq p, q \leq n$,

$$r_i = a_i \leq \Sigma_{j=1}^{n} c_{(i,j)}\delta_j \leq b_i \tag{2}$$

where

$$\delta_j = \begin{cases} -(t_{p+1} + t_{p+2}, \cdots, t_q), & p < q, \quad (3) \\ t_{q+1} + t_{q+2}, \cdots, t_p, & q < p. \quad (3') \end{cases}$$

From equations (2),(3) and (3'), we can find that all $t_1, t_2, \cdots, t_n$, *per se*, must satisfy a group of inequalities consisting of $r_1, r_2, \cdots, r_k$. We denote the group of inequalities as $GI(TAM)$. We immediately will find a solution to judge whether a given sequence γ is a timing evolution behavior of TAM. That is, if γ is a

timing evolution behavior of TAM, then $t_1, t_2, \cdots, t_n$ must be a solution of the group of inequalities in accord with the time constraints in TAM. If there is no any sequence that can satisfy the group of inequalities, i.e, the group has no solution, the ontology versioning based on TAM is not timing consistent. Then we will obtain the following theorem.

Theorem 3. *Ontology versioning and evolution based on TAM is timing consistent if and only if $GI(TAM)$ has at least a solution.*

We will find that theorem 3 gives me a feasible and well-operated solution to dynamically detect evolution behaviors of distributed ontologies. The specific algorithm can be reduced to evaluating the group of inequalities corresponding to the TAM. This can easily be solved by linear programming.

5 Architecture of TEAM

We developed a prototype system call TEAM (TimE Analysis Model) for simply simulating the specific time consistency analysis applications in distributed environments. This system provides a *GUI interface* for interacting with users. Each local ontology repository is used for storing partial ontology and its evolution versions. Each ontology from each repository can use *time consistency checker* for detecting their time consistency. Using *main control component*, the whole distributed system also can be configured and further detect time consistency based evolutions of distributed ontologies. Time Consistency Checker is, *per se*, a solution resolver for an inequation set. We use LINDO API 4.1 [10] for our *time consistency checker*. It can be used to define and resolve a group of inequalities in accord with our time analysis model. Figure 2 gives an example of time consistency checking of an version evolving sequence.

One of key problems of the prototype system is how we can label the time marks for each evolution version. We use OWL language for describing ontology

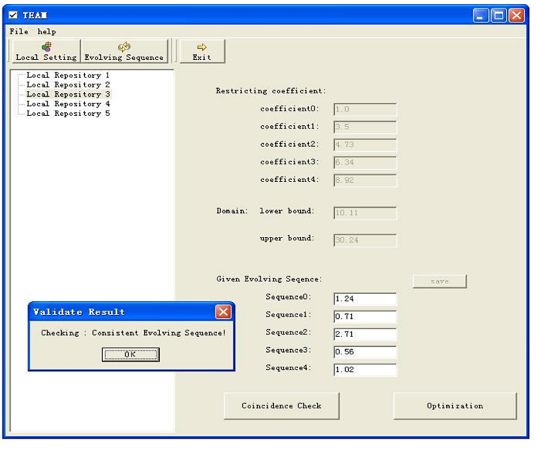


Fig. 2. Time consistency checking of an version evolving sequence

information. We know that each ontology description document includes some ontology version information such as <owl:versionInfo>. We can add some labels such as <dc:revisionBegin> and <dc:revisionEnd> into the version information description of ontology documents, their literal values are defined as built-in datatype `time`. When we want to extract information of performed time of version evolution operations, we can only find these labels and further calculate subtraction of corresponding values of these labels.

6 Related Work and Discussion

Current research areas had rapidly shifted from applications based on simple ontology to some aspects such as representation, evolutions and management of multiple ontologies in a distributed environment. Some research approaches concentrate on ontology versioning and evolution within a distribute environment with diverse but overlapped semantic contexts. These semantically overlapped research work includes some ontology versioning approaches such as KAON, Protege. Compatangelo et al [7] propose a blackboard architecture that also allows the centralized management of different ontology versions in distributed environments. They also formally specify three types of operations performed on distributed ontologies using rewriting rules which include *creation of a concept*, *renaming of a concept* and *addition of attribute/value pair to a concept*. It seems that these work above didn't deal with ontology versioning and evolutions with explicit time constraints. Huang et al. [8] developed a temporal logic approach for reasoning multi-version ontologies based on **LTLm**, the standard model checking algorithm [9] for evaluating a query in the temporal logic **LTLm**. This approach can better express simple linear time constraints between ontology versions, e.g., ontology o occurs prior to ontology o'. It is rather difficult for ontology engineers to specify ontology versioning and evolution with more specific time constraints. Compared with these works, our work is based on the work about different kinds of change operations specified in [7] and extends the work into ontology evolutions with real-time characteristics. It can represent multiple ontology versioning and evolutions with more explicit time constraints in distributed environments. Especially, it can detect valid dynamic evolution behavior of ontologies by the group of inequations. In the case, our model also is feasible and well-operated, and can easily be solved by linear programming. The prototype system can easily be implemented as plugins for some current ontology versioning and evolution management systems. We concentrate on timing versioning of multiple and distributed ontologies because of their operation changes. Therefore, we do not consider semantic change relations between terminologies inside ontologies. In fact, semantic change relations have been partially addressed in above related work such as [8]. In future work, we will apply our distributed timing analysis model to real-time applications of electric power information and provide real-time information maintenance and decision supports.

7 Conclusion

In this paper, we propose a timing analysis model for ontology evolutions with more expressive time constraints in distributed environments. Dynamic versioning and evolutions in the distributed environment can be represented and detected using this timing analysis model. All of these detections can be reduced to detecting whether a group of inequalities has solutions. This enables us to better manage dynamic versioning and evolutions of distributed ontologies.

Acknowledgements

This work is supported by the Chinese National "863" High-Tech Program under Grant No.2006AA01Z231. This work also is supported by Doctor Degree Teacher Science Development Foundation in North China Electric Power University.

References

1. Neches R., et al. Enabling Technology for Knowledge Sharing. In *AI Magazine*, **12(3)**, (1991) 36–56
2. Hendler J. Agents and the Semantic Web. In *IEEE Intelligent Systems*, **16**, (2001) 30–37
3. Klein M., et al. Ontology Versioning and Change Detection on the Web. In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02). Ontologies and the Semantic Web, **2473**, (2002) 197 – 212
4. Noy N.F. and Musen M.A. Ontology Versioning in an Ontology Management Framework. In *IEEE Intelligent System*, (2004) **19(4)** 6–13
5. Noy N., et al. The Prompt Suite: Interactive Tools for Ontology Mergin and Mapping. In *International Journal of Human-Computer Studies*, **59(6)**, (2003) 983–1024
6. Ehrig M and Staab, S. QOM - Quick Ontology Mapping. In the Proceedings of International Semantic Web Conference (ISWC2004), (2004) 683–696
7. Compatangelo E., Vasconcelos W. and Scharlau B. Managing Ontology Versions with A Distributed Blackboard Architecture. In Proceedings of the 24th Int Conf. of the British Computer Societys Specialist Group on Artificial Intelligence (AI2004), Springer-Verlag, 2004
8. Huang Z., and Stuckenschmidt H. Reasoning with Multiversion Ontologies – A Temporal Logic Approach. In Proceedings of the 2005 International Semantic Web Conference (ISWC05), 2005
9. Edmund M. Clarke, et al. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 1999.
10. LINDO System INC. LINDO System API 4.1. 2006. http://lindo.com/products/api/dllm.html

An Optimum Random Forest Model for Prediction of Genetic Susceptibility to Complex Diseases

Weidong Mao and Shannon Kelly

Department of Computer Science, Shippensburg University, Shippensburg, PA 17257
{wmao,sk7776}@ship.edu

Abstract. High-throughput single nucleotide polymorphism (SNP) genotyping technologies make massive genotype data, with a large number of individuals, publicly available. Accessibility of genetic data makes genome-wide association studies for complex diseases possible. One of the most challenging issues in genome-wide association studies is to search and analyze genetic risk factors resulting from interactions of multiple genes. The integrated risk factor usually have a higher risk rate than single SNPs. This paper explores the possibility of applying random forest to search disease-associated factors for given case/control samples. An optimum random forest based algorithm is proposed for the disease susceptibility prediction problem. The proposed method has been applied to publicly available genotype data on Crohn's disease and autoimmune disorders for predicting susceptibility to these diseases. The achieved accuracy of prediction is higher than those achieved by universal prediction methods such as Support Vector Machine (SVM) and previous known methods.

Keywords: random forest, association study, complex diseases, susceptibility, risk factor,prediction.

1 Introduction

Assessing the association between DNA variants and disease has been used widely to identify regions of the genome and candidate genes that contribute to disease [1]. 99.9% of one individual's DNA sequences are identical to that of another person. Over 80% of this 0.1% difference will be Single Nucleotide Polymorphisms, and they promise to significantly advance our ability to understand and treat human disease. In a short, an **SNP** is a single base substitution of one nucleotide with another. Each individual has many single nucleotide polymorphisms that together create a unique DNA pattern for that person. It is important to study SNPs because they represent genetic differences among humans. Genome-wide association studies require knowledge about common genetic variation and the ability to genotype a sufficiently comprehensive set of variants in a large patient sample [2]. High-throughput SNP genotyping technologies make massive genotype data, with a large number of individuals, publicly available.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 193–204, 2007.

Accessibility of genetic data make genome-wide association studies for complex diseases possible.

Success stories when dealing with diseases caused by a single SNP or gene, sometimes called monogenic diseases, were reported. However, most complex diseases, such as psychiatric disorders, are characterized by a non mendelian, multifactorial genetic contribution with a number of susceptible genes interacting with each other [3,4]. A fundamental issue in the analysis of SNP data is to define the unit of genetic function that influences disease risk. Is it a single SNP, a regulatory motif, an encoded protein subunit, a combination of SNPs in a combination of genes, an interacting protein complex, or a metabolic or physiological pathway [5]? In general, a single SNP or gene may be impossible to associate because a disease may be caused by completely different modifications of alternative pathways, and each gene only makes a small contribution. This makes the identifying genetic factors difficult. Multi-SNP interaction analysis is more reliable but it's computationally infeasible. In fact, a 2-SNP interaction analysis for a genome-wide scan with 1 million SNPs requires 10^{12} pairwise tests. An exhaustive search among multi-SNP combination is computationally infeasible even for a small number of SNPs. Furthermore, there are no reliable tools applicable to large genome ranges that could rule out or confirm association with a disease.

It's important to search for informative SNPs among a huge number of SNPs. These informative SNPs are assumed to be associated with genetic diseases. Tag SNPs generated by multiple the linear regression based method [6] are good informative SNPs, but they are reconstruction-oriented instead of disease-oriented. Although the combinatorial search method [7] for finding disease-associated multi-SNP combinations has a better result, the exhaustive search is still very slow.

Multivariate adaptive regression splines models [8,9] are related methods used to detect associations between diseases and SNPs with some degree of success. However, the number of selected predictors is limited and the type of possible interactions must be specified in advance. Multifactor dimensionality reduction methods [10,11] are developed specifically to find gene-gene interactions among SNPs, but they are not applicable to a large sets of SNPs.

The random forest model has been explored in disease association studies [12]. It was applied on simulated case-control data in which the interacting model among SNPs and the number of associated SNPs were specified, thus making the association model simple and the association relatively easier to detect. For real data, such as Crohn's disease [13], multi-SNP interaction is much more complex which involves more SNPs.

In this paper, we first propose an optimum random forest model for searching the most disease-associated multi-SNP combination for given case-control data. In the optimum random forest model, we generate a forest for each variable (e.g. SNP) instead of randomly selecting some variables to grow the classification tree. We can find the best classifier (a combination of SNPs) for each SNP, then we may have M classifiers if the length of the genotype is M. We rank classifiers

according to their prediction rate and assign a weight for each SNP, and in such way we can decide which SNP is more disease-associated.

We next address the disease susceptibility prediction problem [14,15,16,17]. This problem is to assess accumulated information targeted to predicting genotype susceptibility to complex diseases with significantly high accuracy and statistical power. We use the most disease-associated multi-SNPs (generated from the optimum random forest) to grow a random forest and make the genetic susceptibility prediction. The proposed method is applied to two publicly available data: Crohn's disease [13] and autoimmune disorder [18]. The accuracy of the prediction based on the optimum random forest is higher than the accuracy of all previously known methods implemented in [14,15].

In the next section we will overview the random forest tree and classification tree, describe the genetic model, address the problem of searching of most disease-associated multi-SNP and propose the optimum random forest algorithm to find the combination of SNPs which are most associated with diseases. In Section 3 we give the disease susceptibility prediction problem formulation and describe the random forest algorithm. In Section 4 we describe the two real case/control population samples and discuss the results of our experiments.

2 Search for the Most Disease-Associated Multi-SNPs

In this section we first give an overview of the random forest tree and classification tree, then we will describe the genetic model. Next we formulate the problem of searching for the most disease-associated multi-SNP and propose the optimum random forest algorithm.

2.1 Classification Trees and Random Forests

In machine learning, a Random Forest is a classifier that consists of many classification trees. Each tree is grown as follows:

1 If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
2 If there are M input variables, a number $m \ll M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
3 Each tree is grown to the largest extent possible. There is no pruning [19].

A different bootstrap sample from the original data is used to construct a tree. Therefore, about one-third of the cases are left out of the bootstrap sample and not used in the construction of the tree. Cross-validation is not required because the one-third **oob (out-of-bag) data** is used to get an unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance. After each tree is built, we compute the **proximities** or each terminal node.

In every classification tree in the forest, put down the **oob** samples and compute the importance score for each tree based on the number of votes cast for the correct class. This is the importance score for variable m. All variables can be ranked and those important variables can be found in this way.

Random forest is a sophisticated method in data mining to solve classification problems, and it can be used efficiently in disease association studies to find most disease-associated variables such as SNPs that may be responsible for diseases.

2.2 Genetic Model

Recent work has suggested that SNP's in human population are not inherited independently; rather, sets of adjacent SNP's are present on alleles in a block pattern, so called **haplotype**. Many haplotype blocks in human have been transmitted through many generations without recombination. This means although a block may contain many SNP's, it takes only a few SNP's to identify or tag each haplotype in the block. A genome-wide haplotype would comprise half of a diploid genome, including one allele from each allelic gene pair. The **genotype** is the descriptor of the genome which is the set of physical DNA molecules inherited from the organism's parents. A pair of haplotype consists a genotype.

SNP's are bi-allelic and can be referred as 0 for majority allel and 1, otherwise. If both haplotypes are the same allele, then the corresponding genotype is homogeneous, and can be represented as 0 or 1. If the two haplotypes are different, then the genotype is represented as 2.

The case-control sample populations consist of N individuals which are represented in genotype with M SNPs. Each SNP attains one of the three values 0,1, or 2. The sample G is an $(0, 1, 2)$-valued $N \times M$ matrix, where each row corresponds to an individual, each column corresponds to a SNP.

The sample G has 2 classes, **case** and **control**, and M variables, and each of them is represented as a SNP. To construct a classification tree, we split the sample S into 3 child nodes, depending on the value $(0, 1, 2)$ of the variable (SNP) on split site (loci). We grow the tree to the largest possible extent. The construction of the classification tree for case-control sample is illustrated in Fig. 1 . The relationship of a leaf to the tree on which it grows can be described by the hierarchy of splits of branches (starting from the trunk) leading to the last branch from which the leaf hangs. The collection of split site is a Multi-SNPs combination (MSC), which can be viewed as a multi-SNP interaction for the disease. In this example, $MSC = \{5, 9, 3\}$ and $m = 3$, which is a collection of 3 SNPs, represented as its loci.

2.3 Most Disease-Associated Multi-SNPs Problem

To fully understand the basis of complex diseases, it is important to identify the critical genetic factors involved, which is a combination of multiple SNPs. For a given sample G, S is the set of all SNPs (denoted by loci) for the sample, and a *multi-SNPs combination* (MSC) is a subset of S. In disease associations, we need to find a MSC which consists of a combination of SNPs that are well associated with the disease. The problem can be formulated as follows:

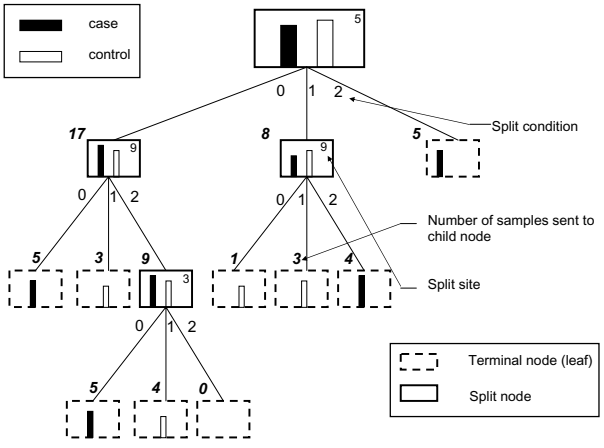


Fig. 1. Classification tree for case-control sample

Most Disease-associated Multi-SNPs Problem. Given a sample G with a set of SNPs S, find the most disease-associated multi-SNPs combination (MDMSC) such that the sample is classified correctly by the MDMSC with high confidence.

Although there are many statistical methods to detect the most disease associated SNPs, such as odds ratio or risk rates, the result is not satisfactory [15]. We decide to search the MDMSC based on the random forest.

2.4 Optimum Random Forest

We generate a MSC for each SNP, the size of the MSC should be much less than the number of SNPs set S ($m \ll M$). The MSC_i ($i = 1, 2, \ldots, M$) must include the i^{th} SNP and the other $(m-1)$ SNPs can be randomly chosen from S. In this way, the M MSCs will cover all SNPs in the sample.

For each MSC, we grow many trees, by permuting the order of the MSC and the training sample. We run all the testing samples down these trees to get the classifier for each sample in the training set, then we can get a classification rate for each tree of a MSC. The classification rate for the MSC is that of the tree whose rate is the highest, and this tree is the best tree. Each member(SNP) of the MSC_i is assigned a weight $w_{i,j}$ ($j \in MSC$) based on the accuracy. The weights for SNPs in the same MSC are the same. All SNPs in all other MSCs will be assigned a weight too. If a SNP is not a member of MSC_i, then $w_{i,j} = 0$.

The weight for each SNP W_j ($j = 1, 2, \ldots, M$) in M is the sum of weights from all MSCs.

$$W_j = \sum_{i=1}^{M} w_{i,j} \tag{1}$$

In the general random forest (GRF) algorithm, the MSC is selected completely randomly and $m \ll M$. It may miss some important SNPs if they are not chosen for any MSC. In our optimum random forest (ORF) algorithm, this scenario is avoided because we generate at least one MSC for each SNP. On the other hand, in GRF, the classifier (forest) consists of trees where there is a correlation between any two trees in the forest, and the correlation will decrease the rate of the classifier. But in ORF, we generate a cluster by permuting the order of the MSC and samples for each tree and the prediction for testing samples is on this cluster only, which is completely independent from the other trees. In this way, we extinguish the correlation among trees.

All SNPs are sorted according to their weights. The most disease-associated SNP is the one with the highest weight. The contribution to diseases of each SNP is quantified by its weight, but in GRF there is no way tell the difference of contribution among SNPs.

3 Disease Susceptibility Prediction

In this section we first describe the input and the output of prediction algorithms and then we show how to apply the optimum random forest to the disease susceptibility prediction.

Data sets have n genotypes and each has m SNPs. The input for a prediction algorithm includes:

(G1) Training genotype set $g_i = (g_{i,j}), i = 0, \ldots, n, j = 1, \ldots m$, $g_{i,j} \in \{0, 1, 2\}$
(G2) Disease status $s(g_i) \in \{0, 1\}$, indicating if $g_i, i = 0, \ldots, n$, is in case (1) or in control (0) , and
(G3) Testing genotype g_t without any disease status.

We will refer to the parts (G1-G2) of the input as the *training set* and to the part (G3) as the test case. The output of prediction algorithms is the disease status of the genotype g_n, i.e., $s(g_t)$.

We use leave-one-out cross-validation to measure the quality of the algorithm. In the leave-one-out cross-validation, the disease status of each genotype in the data set is predicted while the rest of the data is regarded as the training set.

Below we describe several universal prediction methods. These methods are adaptations of general computer-intelligence classifying techniques.

Closest Genotype Neighbor (CN). For the test genotype g_t, find the closest (with respect to Hamming distance) genotype g_i in the training set, and set the status $s(g_t)$ equal to $s(g_i)$.

Support Vector Machine Algorithm (SVM). Support Vector Machine (SVM) is a generation learning system based on recent advances in statistical learning theory. SVMs deliver state-of-the-art performance in real-world applications and have been used in case/control studies [17,21]. There are some SVM softwares available and we decide to use libsvm-2.71 [22] with the following radial basis function:

$$\exp(-\gamma * |u - v|^2)$$

This is the kernel function, where γ is 0.0078125.

General Random Forest (GRF). We use Leo Breiman and Adele Cutler's original implementation of RF version 5.1 [19]. This version of RF handles unbalanced data to predict accurately. RF tries to perform regression on the specified variables to produce the suitable model. RF uses bootstrapping to produce random trees and it has its own cross-validation technique to validate the model for prediction/classification.

Most Reliable 2 SNP Prediction (MR2) [23]**.** This method chooses a pair of adjacent SNPs (site of s_i and s_{i+1}) to predict the disease status of the test genotype g_t by voting among genotypes from the training set which have the same SNP values as g_t at the chosen sites s_i and s_{i+1}. They choose the 2 adjacent SNPs with the highest prediction rate in the training set.

LP-based Prediction Algorithm (LP). This method is based on a graph $X = \{H, G\}$, where the vertices H correspond to distinct haplotypes and the edges G correspond to genotypes connecting its two haplotypes. The density of X is increased by dropping SNPs which do not collapse edges with opposite status. The linear program assigns weights to haplotypes such that for any non-diseased genotype the sum of weights of its haploptypes is less than 0.5 and greater than 0.5 otherwise. We maximize the sum of absolute values of weights over all genotypes. The status of the testing genotype is predicted as sum of its endpoints [14].

Optimum Random Forest(ORF). In the training set, we use random forest to choose those most disease-associated SNPs . The selected disease-associated multi-SNPs combination (MDMSC) (a collection of m SNPs) is used to build the optimum random forest. The m variables are used to split the sample. We may use the same MDMSC to grow many different trees (Tree $T_{3,5,8}$ is different from Tree $T_{8,3,5}$) and choose the best tree (classifier) to predict the disease status of the testing genotype. The best tree has the highest prediction rate in the training set. Since the training set is different when the testing individual is left out, the MSC and the best classifier are different too. The Optimum Random Forest algorithm is illustrated in Fig.2.

4 Results and Discussion

In this section we describe the two real case/control population samples and the results of optimum random forest based susceptibility prediction method on these sets.

4.1 Data Sets

The data set Daly *et al* [13] is derived from the 616 kilobase region of human Chromosome 5q31 that may contain a genetic variant responsible for Crohn's

Input: Training genotype set $G^{n,m} = (g_{i,j} | \ g_{i,j} \in \{0, 1, 2\}, \ i = 1..n, \ j = 1..m)$,
Disease status of $G^{n,m}$, $s^{n,m}$,
The corresponding most disease-associated multi-SNPs combination (MDMSC) for $G^{n,m}$,
The number of trees generated from MDMSC t,
Testing genotype g_t.

For $i = 1$ to t,
  Permute MDMSC, generate a tree T_i,
  For $j = 1$ to 1000,
    Randomly generate a bootstrapped sample S_j from G,
    Run S_j down the tree T_i to get the classification tree,
    Predict testing sample G'_j ($G'_j = G - S_j$) to get the prediction rate $p_{i,j}$,
  Compute the average prediction rate $\bar{p}_i$ for T_i,
Find the best tree T_b which has the highest $\bar{p}$,
Run g_t down the best tree T_b to get the disease status.

Output: Disease status of the test genotype $s(g_t)$.

Fig. 2. Optimum Random Forest Prediction Algorithm

disease by genotyping 103 SNPs for 129 trios. All offspring belong to the case population, while almost all parents belong to the control population. In the entire data, there are 144 case and 243 control individuals. The missing genotype data and haplotypes have been inferred using the 2SNP phasing method [20].

The data set of Ueda *et al* [18] are sequenced from 330kb of human DNA containing genes CD28, CTLA4 and ICONS which are proved related to autoimmune disorder. A total of 108 SNPs were genotyped in 384 cases of autoimmune disorder and 652 controls. Similarly, the missing genotype data and haplotypes have been inferred.

4.2 Measures of Prediction Quality

To measure the quality of prediction methods, we need to measure the deviation between the true disease status and the result of predicted susceptibility, which can be regarded as measurement error. We will present the basic measures used in epidemiology to quantify accuracy of our methods.

The basic measures are:

Sensitivity: the proportion of persons who have the disease who are correctly identified as cases.

Specificity: the proportion of people who do not have the disease who are correctly classified as controls.

The definitions of these two measures of validity are illustrated in the following contingency table.

Table 1. Classification contingency table

		True Status +	True Status -	
Classified	+	a	b	a + b Positive tests
status	-	c	d	c + d Negative tests
Total		a + c	b + d	
		Cases	Controls	

In this table:

a = True positive, people with the disease who test positive
b = False positive, people without the disease who test positive
c = False negative, people with the disease who test negative
d = True negative, people without the disease who test negative

From the table, we can compute Sensitivity (accuracy in classification of cases, Specificity (accuracy in classification of controls) and accuracy:

$$sensitivity = \frac{a}{a+c} \tag{2}$$

$$specificity = \frac{d}{b+d} \tag{3}$$

$$accuracy = \frac{a+d}{a+b+c+d} \tag{4}$$

Sensitivity is the ability to correctly detect a disease. Specificity is the ability to avoid calling normal as disease. Accuracy is the percent of the population that are correctly predicted.

4.3 Results and Discussion

In Table 2 We compare the **optimum random forest (ORF)** method with the other 5 methods we described above. The best accuracy is achieved by ORF

Table 2. The comparison of the prediction rates of 6 prediction methods for Crohn's Disease (Daly *et al*)[13] and autoimmune disorder (Ueda *et al*) [18]

Data Set	Measures	Prediction Methods CN	SVM	GRF	MR2	LP	ORF
(Daly *et al*)	Sensitivity	45.5	x 20.8	34.0	30.6	37.5	70.1
	Specificity	63.3	88.8	85.2	85.2	88.5	76.9
	Accuracy	**54.6**	**63.6**	**66.1**	**65.5**	**69.5**	**74.4**
(Ueda *et al*)	Sensitivity	37.7	14.3	18.0	6.9	7.1	59.4
	Specificity	64.5	88.2	92.8	97.2	91.2	79.6
	Accuracy	**54.8**	**60.9**	**65.1**	**63.9**	**61.3**	**72.1**

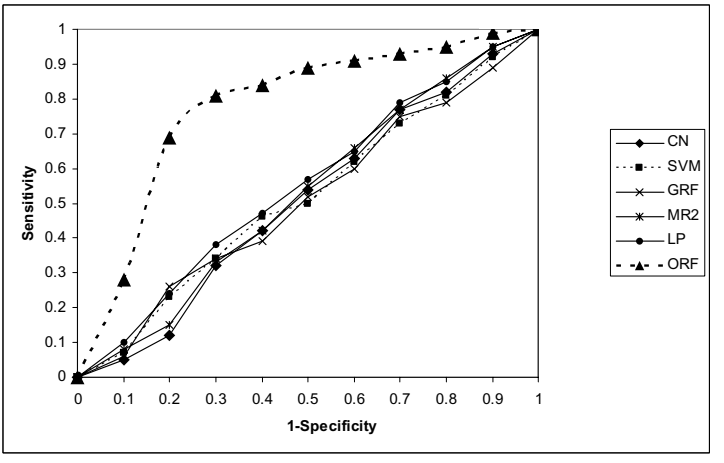


Fig. 3. ROC curve for 6 predictionmethods

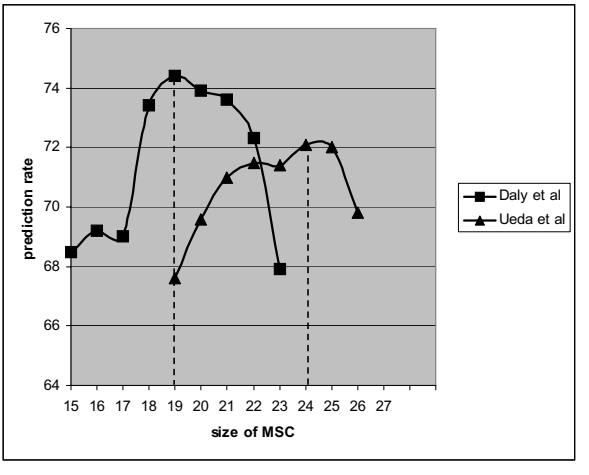


Fig. 4. Best MDMSC size for 2 data sets

- 74.4% and 72.1%, respectively. From the results we can find that the ORF has the best result since we select the most disease-associated multi-SNPs to build the random forest for prediction. Because these SNPs are well associated with the disease, the random forest may produce a good classifier to reflect the association.

Fig. 3 shows the receiver operating characteristics (ROC) curve for 6 methods. A ROC curve represents the tradeoffs between sensitivity and specificity. The ROC curve also illustrates the advantage of ORF over all previous methods.

If the size of MDMSC is m, and the total number of SNPs is M, to get a good classifier, then m should be much less than M. The prediction rate depends on the size of MDMSC, as shown in Fig. 4. In our experiment, we found that the best size of MDMSC is 19 for Crohn's Disease (103 SNPs) and 24 for autoimmune disorder (108 SNPs), respectively.

5 Conclusions

In this paper, we discuss the potential of applying random forest on disease association studies. The proposed genetic susceptibility prediction method based on the optimum random forest is shown to have a high prediction rate and the multi-SNPs being selected to build the random forest are well associated with diseases. In our future work we are going to continue validation of the proposed method.

References

1. Cardon, L.R., Bell, J.I.: Association Study Designs for Complex Diseases, Vol.2. Nature Reviews: Gentics (2001), 91-98.
2. Hirschhorn, J.N.,Daly, M.J.: Genome-wide Association Studies for Common Diseases and Complex Diseases, Vol.6. Nature Reviews: Gentics (2005), 95-108.
3. Merikangas, KR., Risch, N. : Will the Genomics Revolution Revolutionize Psychiatry, The American Journal of Psychiatry, (2003),160:625-635.
4. Botstein, D., Risch, N.:Discovering Genotypes Underlying Human Phenotypes: Past Successes for Mendelian Disease, Future Approaches for Complex Disease, Nature Genetics (2003), 33:228-237.
5. Clark, A.G., Boerwinkle E., Hixson J. and Sing C.F.: Determinants of the success of whole-genome association testing, Genome Res.(2005) 15, 1463-1467.
6. He, J. and Zelikovsky, A.: Tag SNP Selection Based on Multivariate Linear Regression, Proc. of Intl Conf on Computational Science (2006), LNCS 3992, 750-757.
7. Brinza, D., He, J. and Zelikovsky, A.: Combinatorial Search Methods for Multi-SNP Disease Association, Proc. of Intl. Conf. of the IEEE Engineering in Medicine and Biology (2006), to appear.
8. York T.P., Eaves L.J.: Common Disease Analysis using Multivariate Adaptive Regression Splines (MARS): Genetic AnalysisWorkshop 12 simulated sequence data. Genet Epidemiology (2001), 21 Suppl I: S649-54.
9. Cook N.R., Zee R.Y., Ridker P.M.: Tree and Spline Based Association Analysis of gene-gene interaction models for ischemic stroke. Stat Med (2004), 23(9):I439-I453:
10. Ritchie M.D., Hahn L.W., Roodi N., Bailey L.R., Dupont W.D., Parl F.F., Moore J.H. : Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am J Hum Genet. 2001;69:138-147.
11. Hahn L.W., Ritchie M.D., Moore J.H.: Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. Bioinformatics. 2003;19:376-382.
12. Lunetta, K., Hayward, L., Segal, J., Van Eerdewegh, P.: Screening Large-scale Association Study Data: Exploiting Interactions Using Random Forests, BMC Genet. (2004); 5:32.

13. Daly, M., Rioux, J., Schaffner, S., Hudson, T. and Lander, E.: High resolution haplotype structure in the human genome. Nature Genetics (2001) 29, 229-232.
14. Mao, W., He, J., Brinza, D. and Zelikovsky, A.: A Combinatorial Method for Predicting Genetic Susceptibility to Complex Diseases, Proc. Intl. Conf. of the IEEE Engineering In Medicine and Biology Society (EMBC 2005), pp. 224-227.
15. Mao, W., Brinza, D., Hundewale, N., Gremalschi, S. and Zelikovsky, A.: Genotype Susceptibility and Integrated Risk Factors for Complex Diseases, Proc. IEEE Intl. Conf. on Granular Computing (GRC 2006), pp. 754-757.
16. Kimmel, G. and Shamir R.: A Block-Free Hidden Markov Model for Genotypes and Its Application to Disease Association. J. of Computational Biology (2005), Vol. 12, No. 10: 1243-1260.
17. Listgarten, J., Damaraju, S., Poulin B., Cook, L., Dufour, J., Driga, A., Mackey, J., Wishart, D., Greiner,R., and Zanke, B.: Predictive Models for Breast Cancer Susceptibility from Multiple Single Nucleotide Polymorphisms. Clinical Cancer Research (2004), Vol. 10, 2725-2737.
18. Ueda, H., Howson, J.M.M., Esposito, L. et al.: Association of the T Cell Regulatory Gene CTLA4 with Susceptibility to Autoimmune Disease, Nature (2003), 423:506-511.
19. Breiman, L. and Cutler, A. http://www.stat.berkeley.edu/users/breiman/RF
20. Brinza, D. and Zelikovsky, A. :2SNP: Scalable Phasing Based on 2-SNP Haplotypes, Bioinformatics (2006), 22(3), 371-373.
21. Waddell, M., Page,D., Zhan, F., Barlogie, B., and Shaughnessy, J.: Predicting Cancer Susceptibility from SingleNucleotide Polymorphism Data: A Case Study in Multiple Myeloma. Proceddings of BIOKDD (2005), 05.
22. Chang, C. and Lin, C. http://www.csie.ntu.edu.tw/ cjlin/libsvm
23. Kimmel, G. and Shamir R.: A Block-Free Hidden Markov Model for Genotypes and Its Application to Disease Association. J. of Computational Biology (2005), Vol. 12, No. 10: 1243-1260.

Feature Based Techniques for Auto-Detection of Novel Email Worms

Mohammad M Masud, Latifur Khan, and Bhavani Thuraisingham

Department of Computer Science
The University of Texas at Dallas
Richardson, Texas-75083
{mehedy, lkhan, bhavani.thuraisingham}@utdallas.edu

Abstract. This work focuses on applying data mining techniques to detect email worms. We apply a feature-based detection technique. These features are extracted using different statistical and behavioral analysis of emails sent over a certain period of time. The number of features thus extracted is too large. So, our goal is to select the best set of features that can efficiently distinguish between normal and viral emails using classification techniques. First, we apply Principal Component Analysis (PCA) to reduce the high dimensionality of data and to find a projected, optimal set of attributes. We observe that the application of PCA on a benchmark dataset improves the accuracy of detecting novel worms. Second, we apply J48 decision tree algorithm to determine the relative importance of features based on information gain. We are able to identify a subset of features, along with a set of classification rules that have a better performance in detecting novel worms than the original set of features or PCA-reduced features. Finally, we compare our results with published results and discuss our future plans to extend this work.

Keywords: Email worm, data mining, feature selection, Principal Component Analysis, classification technique.

1 Introduction

Worms are malicious code that infect a host machine and spread copies of itself through the network to infect other hosts. There are different kinds of worms such as: Email worms, Instant Messaging worms, Internet worms, IRC worms and File-sharing Networks worms. Email worm, as the name implies, spreads through infected email messages. The worm may be carried by attachment, or the email may contain links to an infected website. When the user opens the attachment, or clicks the link, the host is immediately infected. Email worms use the vulnerability of the email software at the host machine and sends infected emails to the addresses stored in the address book. In this way, new machines get infected. Examples of email worms are "W32.mydoom.M@mm", "W32.Zafi.d", "W32.LoveGate.w", "W32.Mytob.c", and so on. Worms do a lot of harm to computers and people. They can clog the network traffic, cause damage to the system and make the system unstable or even unusable.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 205–216, 2007.

There has been a significant amount of research going on to combat worms. The traditional way of dealing with a known worm is to apply signature based detection. Once a new worm appears, researchers work hard to find a unique pattern in the code that can identify it as a particular type of worm. This unique pattern is called the *signature* of the worm. Thus, a worm can be detected from its signature. But the problem with this approach is that it involves significant amount of human intervention and it may take long time (from days to weeks) to discover the signature. Since worms can propagate very fast, there should be a much faster way to detect them before any damage is done.

We are concerned with the problem of detecting new email worms without knowing their signatures. Thus, our work is directed towards automatic (i.e., without any human intervention) and efficient detection of novel worms. Our work is inspired by [1], which does not require signature detection; rather, it extracts different features from the email (to be explained shortly) and tries to classify the email as clean or infected. They employ two classifiers in series, the first one is Support Vector Machine (SVM) and the next one is Naïve Bayes (NB). We will refer to this two-layer approach as 'SVM + NB series' approach. They report a very high accuracy, as well as very low false positive and false negative rate. But we address several issues with their approach. First, they do not apply a balanced dataset to test classification accuracy. Second, they do not apply cross validation on the dataset. Third, our experimental results indicate that a two-layer approach is less efficient than a single layer approach in terms of cross validation accuracy on a balanced dataset. Fourth, they apply too many features, most of which is found to be redundant by our study. We deal with all these problems and provide efficient solutions.

Our contributions to this research work are as follows: First, we compare individual performances of NB and SVM with their SVM + NB series counterpart. We show that the series approach is less effective than either NB or SVM alone. So, we claim that one of the layers of the two-layer approach is redundant. Second, we rearrange the dataset so that it becomes more balanced. We divide the dataset into two portions: one containing only known worms (and some clean emails), the other containing only a novel worm. Then we apply a three-fold cross validation on the 'known worm' dataset, and we test the accuracy of each of the learned classifiers on the 'novel worm' dataset. We report the cross validation accuracy, and novel detection accuracy for each of the datasets. Third, we apply PCA on the dataset to improve the efficiency of classification task. PCA is commonly used to extract patterns from high dimensional data, especially when the data is noisy. Besides, it is a simple and nonparametric method. Since the original dataset contains a total of 94 attributes, it is very likely that some of these attributes are redundant, while some others add noise into the data, and PCA could be effectively applied to reduce this data to a lower dimension; revealing the underlying simple pattern hidden in the data. Fourth, we build decision tree, using the WEKA [2] implementation of C4.5 [3] called the J48, from the dataset to identify the most important features according to information gain. We find that only a few features are sufficient to obtain similar or better classification accuracy. We report the features as well as the classification rules obtained from the decision tree.

The rest of this paper is organized as follows: section 2 describes related work in automatic email worm detection, section 3 describes the feature reduction and

selection techniques, section 4 describes the dataset, section 5 describes the experiments, section 6 discusses the results, and section 7 concludes with future guidelines for research.

2 Related Work

There are different approaches to automate the detection of worms. These approaches are mainly of two types: behavioral and content-based. Behavioral approaches analyze the behavior of messages like source-destination addresses, attachment types, message frequency etc. Content-based approaches look into the content of the message, and try to detect signature automatically. There are also combined methods that take advantage of both techniques.

An example of behavioral detection is social network analysis [4, 5]. It detects worm infected emails by creating graphs of network, where users are represented as nodes, and communications between users are represented as edges. A social network is a group of nodes among which there exists edges. Emails that propagate beyond the group boundary are considered to be infected. But the drawback of this system is that worms can easily bypass social networks by intelligently choosing the recipient lists, by looking at recent emails in the user's outbox.

Statistical analysis of outgoing emails is another behavioral approach [6, 7]. Statistics collected from frequency of communication between clients and their mail server, byte sequences in the attachment etc. are used to predict anomalies in emails and thus worms are detected.

Example of content based approach is the EarlyBird System [8]. In this system, statistics on highly repetitive packet contents are gathered. These statistics are analyzed to detect possible infection of host or server machines. This method generates content signature of worm without any human intervention. Results reported by this system indicated very low false positive rate of detection. Other examples are the Autograph [9], and the Polygraph [10], developed at Carnegie Mellon University.

There are other approaches to detect early spreading of worms, such as employing honeypot" [11]. A honeypot is a closely monitored decoy computer that attracts attacks for early detection and in-depth adversary analysis. The honeypots are designed to not send out email in normal situations. If a honeypot begins to send out emails after running the attachment of an email, it is determined that this email is an email worm.

Martin et al. [12] also report an experiment with email data, where they apply a statistical approach to find an optimum subset of a large set of features to facilitate the classification of outgoing emails, and eventually, detect novel email worms.

Another approach by Sidiroglou et al. [13] employs behavior-based anomaly detection, which is different from the signature based or statistical approaches. Their approach is to open all suspicious attachments inside an instrumented virtual machine looking for dangerous actions, such as writing to the Windows registry, and flag suspicious messages.

Although our approach is feature-based, it is different from the above feature-based detection approaches in that we apply PCA, and decision tree to reduce the dimension

of data. Rather than choosing a subset of features, PCA finds a linear combination of the features and projects them to a lower dimension, reducing noise in data. On the other hand, we apply decision tree to identify the most important features, thereby removing redundant or noisy features. Both these approaches achieve higher accuracy.

3 Feature Reduction and Classification Techniques

Firstly, we briefly describe the features that are used in email worm detection. These features are extracted from a repository of outgoing emails collected over a period of two years [1]. These features are categorized into two different groups: i) per-email feature and ii) per-window feature. Per-email features are features of a single email, while per-window features are features of a collection of emails sent within a window of time. Secondly, we describe our feature reduction techniques, namely, PCA and J48. Finally, we briefly describe the two-layer approach and its limitations.

3.1 Feature Description

For a detailed description of the features please refer to [12]. Each of these features are either continuous valued or binary. Value of a binary feature is either 0 or 1, depending on the presence or absence of this feature in a data point. There are a total of 94 features. Here we describe some of them.

3.1.1 Per Email Features

i. *HTML in body*: Whether there is HTML in the email body. This feature is used because a bug in the HTML parser of the email client is a vulnerability that may be exploited by worm writers. It is a binary feature.
ii. *Embedded image*: Whether there is any embedded image. This is used because a buggy image processor of the email client is also vulnerable to attacks. It is a binary feature.
iii. *Hyperlinks*: Whether there are hyperlinks in the email body. Clicking an infected link causes the host to be infected. It is also a binary feature.
iv. *Binary Attachment*: Whether there are any binary attachments. Worms are mainly propagated by binary attachments. This is also a binary feature.
v. *Multipurpose Internet Mail Extensions* (*MIME*) *type of attachment*: There are different MIME types, for example: "application/msword", "application/pdf", "image/gif", "text/plain" etc. Each of these types is used as a binary feature (total 27).
vi. *UNIX "magic number" of file attachments*: Sometimes a different MIME type is assigned by the worm writers to evade detection. Magic numbers can accurately detect the MIME type. Each of these types is used as a binary feature (total 43).
vii. *Number of attachments*: It is a continuous feature.
viii. *Number of words/characters in subject/body*: These features are continuous. Most worms choose random text, whereas a user may have certain writing characteristics. Thus, these features are sometimes useful to detect infected emails.

3.1.2 Per Window Features

i. *Number of emails sent in window*: An infected host is supposed to send emails at a faster rate. This is a continuous feature.
ii. *Number of unique email recipients*, *senders*: These are also important criteria to distinguish between normal and infected host. This is a continuous feature too.
iii. *Average number of words/characters per subject, body; average word length*: These features are also useful in distinguishing between normal and viral activity.
iv. *Variance in number of words/characters per subject, body; variance in word length:* These are also useful properties of email worms.
v. *Ratio of emails with attachments:* usually, normal emails do not contain attachments, whereas most infected emails do contain them.

3.2 Feature Reduction and Selection

The high dimensionality of data always appears to be a major problem for classification tasks because i) it increases the running time of the classification algorithms, ii) it increases chance of overfitting, and iii) large number of instances are required for learning tasks. We apply PCA to obtain a reduced dimensional data and apply decision tree to select a subset of features, in order to eliminate these problems.

3.2.1 Principal Component Analysis: Reducing Data Dimension

PCA finds a reduced set of attributes by projecting the original attributes into a lower dimension. We observe that for some optimal dimension of projection, the reduced dimensional data observes a better accuracy in detecting novel worms. PCA not only reduces the dimension of data to eliminate all these problems, but also discovers hidden patterns in data, thereby increasing classification accuracy of the learned classifiers. As high dimensional data contains redundancies and noises, it is much harder for the learning algorithms to find a hypothesis consistent with the training instances. The learned hypothesis is likely to be too complex and susceptible to overfitting. PCA reduces the dimension, without losing much information, and thus allows the learning algorithms to find a simpler hypothesis that is consistent with the training examples, and thereby reduces the chance of overfitting. But it should be noted that PCA projects data into a lower dimension in the direction of maximum dispersion. Maximum dispersion of data does not necessarily imply maximum separation of between – class data and/or maximum concentration of within – class data. If this is the case, then PCA reduction may result in poor performance. That is why we apply PCA to reduce the 94-dimensional data into different lower dimensions, ranging from 5 to 90, and select the optimal dimension that achieves the highest classification accuracy.

3.2.2 Decision Tree: Feature Selection Using Information Gain

Feature selection is different from dimension reduction because it selects a subset of the feature set, rather than projecting combination of features onto lower dimension. There are different feature selection techniques available, such as greedy selection, which selects the features one after another until the classification accuracy deteriorates. But the problem with this selection approach is that it takes a lot of time and depending on the order of selection, results vary significantly. We apply the

decision tree approach for feature selection because of three reasons. First, it is fast, second, it applies information gain to select best features, and finally, we can extract a set of rules from the decision tree that reveals the true nature of the 'positive' or the 'negative' class. Thus, we are not only aware of the essential attributes but also get an overall idea about the infected emails. It may also be possible to generalize two or more rules to obtain a generalized characteristic of different types of worms.

Information gain is a very effective metric in selecting features. Information gain can be defined as a measure of the effectiveness of an attribute (i.e., feature) in classifying the training data [14]. If we split the training data on this attribute values, then information gain gives the measurement of the expected reduction in entropy after the split. The more an attribute can reduce entropy in the training data, the better the attribute in classifying the data. Information Gain of a binary attribute A on a collection of examples S is given by (1):

$$Gain\ (S, A) \equiv Entropy\ (S) - \sum_{V \in Values\ (A)} \frac{|\ S_v\ |}{|\ S\ |} Entropy\ (S_v) \tag{1}$$

Where $Values(A)$ is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has value v. In our case, each binary attribute has only two possible values (0, 1). Entropy of subset S is computed using the following equation:

$$Entropy(S) = -\frac{p(s)}{n(s)+p(s)} \log_2\left(\frac{p(s)}{n(s)+p(s)}\right) - \frac{n(s)}{n(s)+p(s)} \log_2\left(\frac{n(s)}{n(s)+p(s)}\right) \tag{2}$$

Where $p(S)$ is the number of positive examples in S and $n(S)$ is the total number of negative examples in S. Computation of information gain of a continuous attribute is a little tricky, because it has infinite number of possible values. One approach followed by Quinlan [3] is to find an optimal threshold, and split the data into two halves. The optimal threshold is found by searching a threshold value with highest information gain within the range of values of this attribute in the dataset.

We use J48 for building decision tree, which is an implementation of C4.5. Decision tree algorithms choose the best attribute based on information gain criterion at each level of recursion. Thus, the final tree actually consists of the most important attributes that can distinguish between the positive and negative instances. The tree is further pruned to reduce chances of overfitting. Thus, we are able to identify the features that are necessary and the features that are redundant, and use only the necessary features. Surprisingly enough, in our experiments we find that only four/five features are necessary among the ninety-four features. The decision trees generated in our experiments have better classification accuracies than both original and PCA reduced data.

3.3 Classification Techniques

We apply the NB [15] and SVM [16] classifiers in our experiments. We also apply the SVM+NB series classifier and J48 Decision Tree classifier. NB, SVM and the series classifiers are applied on the original data and the PCA-reduced data, while the J48 is applied on the original data only, because the classifier itself selects a subset of features, discarding redundant ones. The SVM+NB series is implemented as per [1].

We do not recommend using the series classifier because of the following reasons. First, it is not practical. Because we must come up with a set of parameter values such that the false positive of SVM becomes zero. Given a set of continuous parameters, the problem of finding this optimal point is computationally intractable. Second, the assumption in this approach is wrong. Because, even if we happen to find an optimal point luckily, there is no guarantee that these set of values will work on a test data, since this optimal point is obtained from the training data. Third, if NB performs poorly on a particular test set, the output would also be poor. Because, both NB and SVM must perform well to produce a good series result, if any one fails, the combined approach would also fail. In our experimental results, we have indicated the effect of all these problems.

4 Dataset

We have collected the worm dataset used in the experiment by Martin et al. [1]. They have accumulated several hundreds of clean and worm emails over a period of two years. All these emails are outgoing emails. Several features are extracted from these emails as explained in section 3.1.

There are six types of worms contained in the dataset: VBS.BubbleBoy, W32.Mydoom.M, W32.Sobig.F, W32.Netsky.D, W32.Mydoom.U, and W32.Bagle.F. But the classification task is binary: {clean, infected}. The original dataset contains six training and six test sets. Each training set is made up of 400 clean emails and 1000 worm emails. The worm emails are made up of 200 examples from each of the five different worms. The sixth virus is then included in the test set, which contains 1200 clean emails and 200 infected messages.

5 Experiments

As we have mentioned earlier, the dataset is imbalanced. So we apply both cross validation and novel worm detection in our experiments. In our distribution, each balanced set contains 1600 clean email messages, which are the combination of all the clean messages in the original dataset (400 from training set, 1200 from test set). Also, each balanced set contains 1000 viral messages (from original training set), marked as “known worms” and 200 viral messages (the sixth worm from the original test set), and marked as “novel worm”. The cross validation is done as follows: we randomly divide the set of 2600 (1600 clean + 1000 viral) messages into three equal sized subsets. We take two subsets as training set and the remaining set as the test set. This is done three times by rotating the testing and training sets. We take the average accuracy of these three runs. This accuracy is shown under the column *accuracy* in following tables. Besides testing the accuracy of the test set, we also test the detection accuracy of the learned classifier on the “novel worm” set. This accuracy is also averaged over all runs and shown as *novel detection accuracy*.

For SVM, we use libsvm [17] package, and apply C-Support Vector Classification (C-SVC) with the radial basis function using “gamma” = 0.2 and “C”=1. We use our

own C++ implementation of NB. We use the WEKA [2] implementation of J48, with pruning applied. We extract rules from the decision trees generated using J48.

6 Results

We discuss the results in three separate subsections. In section 6.1 we discuss the results found from the original data. In section 6.2, we discuss the results found from the reduced dimensional data using PCA. In section 6.3, we discuss the results obtained using J48.

6.1 Results from the Original Dataset

The results are shown in table 1. Table 1 reports the accuracy of the cross validation and novel detection for each dataset. The cross validation accuracy is shown under the column '*Acc'* and the accuracy of detecting novel worms is shown under the column '*novel detection acc'*. Each worm at the row heading is the novel worm for that dataset. In table 1, we report accuracy and novel detection accuracy for each of the six worm types. From the results reported in table 1, we see that SVM observes the best accuracy among all classifiers. The best accuracy observed by SVM is 99.77%, on the sobig.f dataset, while the worst accuracy observed by the same is 99.58%, on the mydoom.m dataset.

Table 1. Comparison of accuracy (%) of different classifiers on the worm dataset

Worm Type	**NB**		**SVM**		**SVM+NB**	
	Acc (%)	Novel detection Acc (%)	Acc (%)	Novel detection Acc (%)	Acc (%)	Novel detection Acc (%)
Mydoom.m	99.42	21.72	99.58	30.03	99.38	21.06
sobig.f	99.11	97.01	99.77	97.01	99.27	96.52
Netsky.d	99.15	97.01	99.69	65.01	99.19	64.02
Mydoom.u	99.11	97.01	99.69	96.19	99.19	96.19
Bagle.f	99.27	97.01	99.61	98.01	99.31	95.52
Bubbleboy	99.19	0	99.65	0	99.31	0
Average	**99.21**	**68.29**	**99.67**	**64.38**	**99.28**	**62.22**

6.2 Results from the Reduced Dimensional Data (Reduced by PCA)

The following chart (Fig. 1) shows the results of applying PCA on the original data. The *X* axis represents the dimension of the reduced dimensional data, which has been varied from 5 to 90, with step 5 increments. The last point on the *X* axis is the unreduced or original dimension. Fig. 1 shows the cross validation accuracy for different dimensions. The data from the chart should be read as follows: a point (x, y) on a given line, say the line for SVM, indicates the cross validation accuracy y of

SVM, averaged over all six datasets, where each dataset has been reduced to x dimension using PCA.

Fig. 1 indicates that at lower dimensions, cross validation accuracy is lower, for each of the three classifiers. But SVM is found to have achieved its near maximum accuracy when data dimension is 30. NB and SERIES reaches within 2% of maximum accuracy at dimension 30 and onwards. All classifiers attain their maximum at the highest dimension 94, which is actually the unreduced data. So, from this observation, we may conclude that PCA is not effective on this dataset, in terms of cross validation accuracy. The reason behind this poorer performance on the reduced dimensional data is possibly the one that we have mentioned earlier in section 3.2. The reduction by PCA is not producing a lower dimensional data where dissimilar class instances are maximally dispersed and similar class instances are maximally concentrated. So, the classification accuracy is lower at lower dimensions.

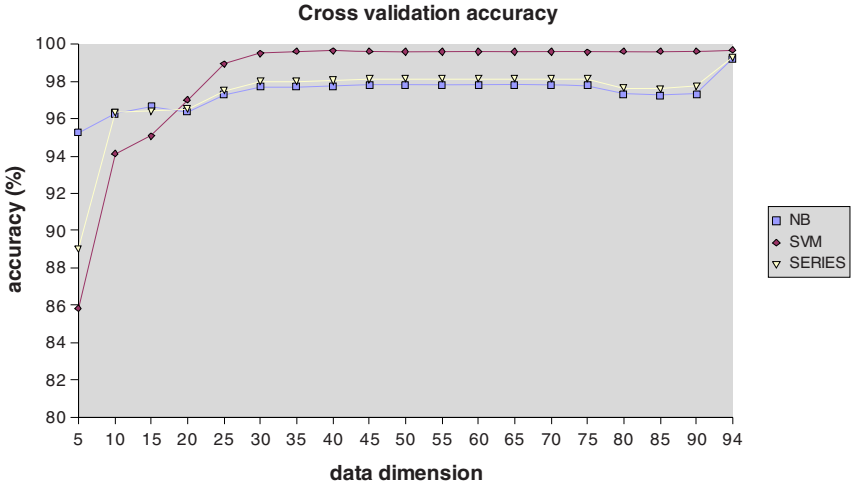


Fig. 1. Average cross validation accuracy of the three classifiers on lower dimensional data, reduced by PCA

We now present the results, at dimension 25, similar to the results presented in previous section. Table 2 compares the novel detection accuracy and the cross validation accuracy of different classifiers. We choose this particular dimension because, at this dimension all the classifiers seem to be the most balanced in all aspects: cross validation accuracy, false positive and false negative rate and novel detection accuracy. We conclude that this dimension is the optimal dimension of projection by PCA.

Results in Table 2 indicate that accuracy and novel detection accuracy of SVM are higher than NB, respectively. Also, as mentioned in previous section, we again observe that accuracy and novel detection accuracy of SVM+NB is worse than both NB and SVM. Thus, SVM is found to be the best among these three classifiers, in both unreduced and reduced dimensions.

Table 2. Comparison of accuracy among different classifiers on the PCA reduced worm dataset at dimension 25

Worm Type	NB		SVM		SVM+NB	
	Acc (%)	Novel detection Acc (%)	Acc (%)	Novel detection Acc (%)	Acc (%)	Novel Detection Acc (%)
Mydoom.m	99.08	25.0	99.46	30.02	99.15	24.7
sobig.f	97.31	97.01	99.19	97.01	97.77	97.01
Netsky.d	96.61	97.51	98.62	97.01	96.73	97.01
Mydoom.u	96.92	97.51	98.46	97.34	97.15	97.34
bagle.f	96.92	97.51	98.93	97.01	97.07	97.01
Bubbleboy	96.96	0	98.88	0	97.08	0
Average	**97.3**	**69.09**	**98.92**	**69.73**	**97.49**	**68.85**

6.3 Results Obtained from J48

Table 3 reports the accuracy, novel detection accuracy, #of selected features and tree size for different worm types. Comparing with the previous results, we find that the average novel detection accuracy of J48 is the highest (70.9%) among all the classifiers both in the original and PCA-reduced dataset. Besides, the average accuracy (99.35%) is also better than all other classifiers in the reduced dataset and very close to the best accuracy (SVM, 99.67%) in the original dataset. Surprisingly enough, on average only 4.5 features have been selected by the decision tree algorithm, which means almost 90 other features are redundant. It is interesting to see which features have been selected by the decision tree algorithm. First, we describe the rules in disjunctive normal form (DNF) that we have extracted from each of the decision trees. Each rule is expressed as a disjunction of one or more conditions. We use the symbol '$\wedge$' to denote conjunction and '$\vee$' to denote disjunction. We are able to detect the reason (explained later) behind the poor performance of all the classifiers in **Bubbleboy** dataset, where all of them have 0% novel detection accuracy.

Table 3. Accuracy (%), novel detection accuracy (%), #of selected features, and tree size as obtained by applying J48 on the original dataset

Worm Type	Acc (%)	Novel detection Acc (%)	Total features selected	Tree size (total nodes)
Mydoom.m	99.3	32.0	4	11
sobig.f	99.4	97.5	4	11
Netsky.d	99.2	99.0	4	11
Mydoom.u	99.2	97.0	6	15
bagle.f	99.4	99.5	6	15
Bubbleboy	99.6	0.5	3	7
Average	**99.35**	**70.92**	**4.5**	**11.67**

Worm rules: if any of the following rules is satisfied then it is a worm

Rule I (from Mydoom.m dataset):
[(VarWordsInBody<=457)∧(RatioAttach<=0.9)∧(MeanWordsInBody<=22.1)]
∨ [(VarWordsInBody<=457)∧(RatioAttach<=0.9)]

Rule II (from sobig.f dataset): [(RatioAttach > 0.7) ∧ (VarAttachSize <= 7799173)]
∨ [(RatioAttach>=0.7)∧(VarAttachSize>7799173)∧(NumAttachments>0)]

Rule III (from Netsky.d dataset): [(RatioAttach > 0.6) ∧ (VarAttachSize <= 10229122)]
∨ [(RatioAttach>=0.7)∧(VarAttachSize>10229122)∧(NumAttachments>0)]

Rule IV (from Mydoom.u dataset):
[(FreqEmailSentInWindow <= 0.067) ∧ (MeanWordsInBody <= 60.6)]
∨ [(FreqEmailSentInWindow<=0.067)∧(MeanWordsInBody>60.6)∧(NumAttachments>0)]

Rule V (from bagle.f dataset): [(RatioAttach > .6) ∧ (VarAttachSize <= 7799173)]
∨[(RatioAttach>.6)∧(VarAttachSize>7799173)∧NumAttachments>0)∧(AvgWordLength<45)]

Rule VI (from Bubbleboy dataset):
[(NumFromAddrInWindow>1)∧(AttachmentIsText0= 1)]

By looking at the above rules, we can easily find some important features such as:

VarWordsInBody, RatioAttach, MeanWordsInBody, NumAttachments, VarAttachSize, and so on. Using the above rules, we can also summarize general characteristics of worm. For example, it is noticeable that for most of the worms, *RatioAttach* >= 0.7, as well as NumAttachments > 0. These generalizations may lead to a generalized set of rules that would be effective against a new attack.

The rule VI above is obtained from the 'Bubbleboy' dataset. But only one of the 200 test cases satisfies this rule, so the novel detection accuracy is only 0.5%. Other classifier results also show that novel detection accuracy on the dataset is also 0%. This indicates that this worm has completely different characteristics, and cannot be detected by the generalizations obtained on other five worm types.

7 Conclusion

In this work, we explore three different data mining approaches to automatically detect email worms. The first approach is to apply either NB or SVM on the original dataset, without any feature reduction, and train a classifier. The second approach is to reduce data dimension using PCA and apply NB or SVM on the reduced data and train a classifier. The third approach is to select best features using decision tree such as J48 and obtain classification rules. Results obtained from our experiments indicate that J48 achieves the best performance. Looking at the rules extracted from the decision tree, we conclude that the feature space is actually very small, and the classifiers are quite simple. That is why the tree based selection performs better than PCA in this dataset. It might not have been the case if the feature space had been more complex. In that case, the second approach would have been more effective. The first approach would be suitable if we have only a few features in the original data. In summary, all the approaches are possible to apply, depending on the characteristic of the dataset.

In future, we would like to continue our research in detecting worms by combining feature-based approach with content-based approach to make it more robust and

efficient. Besides, rather than relying entirely on features, we are willing to focus on the statistical property of the contents of the messages for possible contamination of worms. Finally, we would also like to obtain a larger and richer collection of dataset.

References

1. Martin, S., Sewani, A., Nelson, B., Chen, K., Joseph, A.D. A Two-Layer Approach for Novel Email Worm Detection. Submitted to USENIX SRUTI (Steps on Reducing Unwanted Traffic on the Internet) 2005.
2. WEKA: Data Mining Software in Java. http://www.cs.waikato.ac.nz/~ml/weka/
3. Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
4. Golbeck, J., and Hendler, J. Reputation network analysis for email filtering. In CEAS (2004).
5. Newman, M. E. J., Forrest, S., and Balthrop, J. Email networks and the spread of computer viruses. Physical Review E 66, 035101 (2002).
6. Symantec Corporation. W32.Beagle.BG. Online, 2005. http://www.sarc.com/avcenter/venc/data/w32.beagle. bg@mm.html.
7. Schultz, M., Eskin, E., and Zadok, E. MEF: Malicious email filter, a UNIX mail filter that detects malicious windows executables. In USENIX Annual Technical Conference - FREENIX Track (June 2001).
8. Singh, S., Estan, C., Varghese, G., and Savage, S. The EarlyBird System for Real-time Detection of Unknown Worms. Technical report - cs2003-0761, UCSD, 2003.
9. Kim, H.-A. and Karp, B., Autograph: Toward Automated, Distributed Worm Signature Detection, in the Proceedings of the 13th Usenix Security Symposium (Security 2004), San Diego, CA, August, 2004.
10. J. Newsome, B. Karp, and D. Song. Polygraph: Automatically Generating Signatures for Polymorphic Worms. In Proceedings of the IEEE Symposium on Security and Privacy, May 2005.
11. Honeypot. http://www.honeypots.net/.
12. Martin, S., Sewani, A., Nelson, B., Chen, K., Joseph, A.D. Analyzing Behavioral Features for Email Classification, In the Proceedings of the IEEE Second Conference on Email and Anti-Spam (CEAS 2005), July 21 & 22, 2005, Stanford University.
13. Sidiroglou, S., Ioannidis, J., Keromytis, A.D., Stolfo, S.J. An Email Worm Vaccine Architecture. Proceedings of the First International Conference on Information Security Practice and Experience (ISPEC 2005), Singapore, April 11-14, 2005: 97-108
14. Mitchell, T. Machine Learning. McGraw Hill, 1997.
15. John, G.H., Langley, P.: Estimating Continuous Distributions in Bayesian Classifiers. In the Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan KaufMann Publishers, San Mateo, (1995) 338-345.
16. Boser, B. E., Guyon, I. M. and Vapnik, V. N. A training algorithm for optimal margin classifiers. In D. Haussler, editor, 5th Annual ACM Workshop on COLT, pages 144-152, Pittsburgh, PA, 1992. ACM Press.
17. A library for Support Vector Machine: http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Multiresolution-Based BiLinear Recurrent Neural Network

Byung-Jae Min, Dong-Chul Park, and Hwan-Soo Choi

Dept. of Information Eng. and Myongji IT Eng. Research Inst.
Myong Ji University, Korea
{mbj2000,parkd,hschoi}@mju.ac.kr

Abstract. A Multiresolution-based BiLinear Recurrent Neural Network (MBLRNN) is proposed in this paper. The proposed M-BLRNN is based on the BLRNN that has been proven to have robust abilities in modeling and predicting time series. The learning process is further improved by using a multiresolution-based learning algorithm for training the BLRNN so as to make it more robust for long-term prediction of the time series. The proposed M-BLRNN is applied to long-term prediction of network traffic. Experiments and results on Ethernet network traffic data show that the proposed M-BLRNN outperforms both the traditional Multi-Layer Perceptron Type Neural Network (MLPNN) and the BLRNN in terms of the normalized mean square error (NMSE).

Keywords: Wavelet, Recurrent Neural Networks.

1 Introduction

Predicting a chaotic time series is equivalent to approximating an unknown nonlinear function mapping of a chaotic signal. Various models have been proposed to model and predict the future behavior of time series. Statical models such as moving average and exponential smoothing methods, linear regression models, autoregressive models (AR), autoregressive moving average (ARMA) models, and Kalman filtering-based methods have been widely used in practice [1].

In recent years, various nonlinear models have been proposed for time series prediction [2,3]. One group of models that has garnered strong interest is neural networks(NN)-based models, because of their universal approximation capabilities [4,5]. As shown in a wide range of engineering applications, NN-based models have been successfully applied and well accepted in numerous practical problems. Among these NN-based models, the feed-forward neural network, also known as the MultiLayer Perceptron Type Neural Network (MLPNN), is the most popularly used, and has been applied to solve many difficult and diverse problems. A recurrent neural network (RNN) model with consideration of the internal feed-back was proposed to overcome the inherent limitations of the MLPNN. The RNN has been proven to be more efficient than the MLPNN in modeling dynamical systems and has been widely used for time series prediction [5].

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 217–223, 2007.

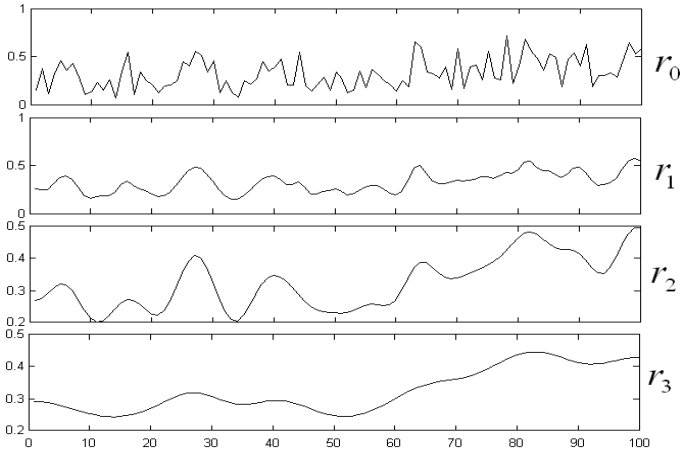


Fig. 1. Example of different representations obtained by the wavelet transform

In this paper, we propose a Multiresolution-based BiLinear Recurrent Neural Network (M-BLRNN) for time series prediction. The proposed M-BLRNN is based on the BLRNN that has been proven to have robust abilities in modeling and predicting time series [6,7]. The proposed M-BLRNN is verified through application to network traffic prediction. The experiments and results show that the proposed M-BLRNN is more efficient than the traditional MLPNN and the BLRNN with respect to long-term prediction of network traffic.

The remainder of this paper is organized as follows: Section 2 presents a review of multiresolution analysis with a wavelet transform. A brief review of the BLRNN is given in Section 3. The proposed M-BLRNN is presented in Section 4. Section 5 presents some experiments and results on a network traffic data set including a performance comparison with the traditional MLPNN and BLRNN models. Section 6 concludes the paper.

2 Multiresolution Wavelet Analysis

The wavelet transform [8], a novel technology developed in the signal processing community, has received much attention from neural network researchers in recent years. Several NN models based on a multiresolution analysis using a wavelet transform have been proposed for time series prediction [9] and signal filtering [10]. The aim of the multiresolution analysis is to analyze a signal at different frequencies with different resolutions. It produces a high quality local representation of a signal in both the time domain and the frequency domain.

The calculation of the *à* trous wavelet transform can be described as follows: First, a low-pass filter is used to suppress the high frequency components of a signal while allowing the low frequency components to pass through. The scaling

function associated with the low-pass filter is then used to calculate the average of elements, which results in a smoother signal.

The smoothed data $c_j(t)$ at given resolution j can be obtained by performing successive convolutions with the discrete low-pass filter h,

$$c_j(t) = \sum_k h(k) c_{j-1}(t + 2^{j-1}k) \tag{1}$$

where h is a discrete low-pass filter associated with the scaling function and $c_0(t)$ is the original signal. A suitable low-pass filter h is the B_3 spline, defined as $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$.

From the sequence of the smoothing of the signal, the wavelet coefficients are obtained by calculating the difference between successive smoothed versions:

$$w_j(t) = c_{j-1}(t) - c_j(t) \tag{2}$$

By consequently expanding the original signal from the coarsest resolution level to the finest resolution level, the original signal can be expressed in terms of the wavelet coefficients and the scaling coefficients as follow:

$$c_0(t) = c_J(t) + \sum_{j=1}^{J} w_j(t) \tag{3}$$

where J is the number of resolutions and $c_J(t)$ is the finest version of the signal. Eq.(3) also provides a reconstruction formula for the original signal.

3 BiLinear Recurrent Neural Networks

The BLRNN is a simple recurrent neural network, which has a robust ability in modeling dynamically nonlinear systems and is especially suitable for time-series data. The model was initially proposed by Park and Zhu [6]. It has been successfully applied in modeling time-series data [6,7].

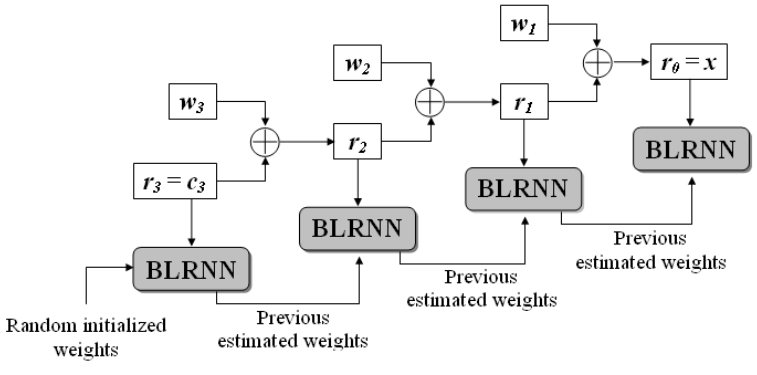


Fig. 2. Learning process using the multiresolution-based learning algorithm

In the following, we explain about a simple BLRNN that has N input neurons, M hidden neurons and where $K = N-1$ degree polynomials is given. The input signal and the nonlinear integration of the input signal to hidden neurons are:

$$\boldsymbol{X}[n] = [x[n], x[n-1], ..., x[n-K]]^T$$
$$\boldsymbol{O}[n] = [o_1[n], o_2[n], ..., o_M[n]]^T$$

where T denotes the transpose of a vector or matrix and the recurrent term is a $M \times K$ matrix.

$$\boldsymbol{Z}_p[n] = [o_p[n-1], o_p[n-2], ..., o_p[n-K]] \tag{4}$$

$$\begin{aligned} s_p[n] &= w_p + \sum_{k_1=0}^{N-1} a_{pk_1} o_p[n-k_1] \\ &\quad + \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} b_{pk_1k_2} o_p[n-k_1]x[n-k_2] \\ &\quad + \sum_{k_2=0}^{N-1} c_{pk_2} x[n-k_2] \\ &= w_p + \boldsymbol{A}_p^T \boldsymbol{Z}_p^T[n] + \boldsymbol{Z}_p[n]\boldsymbol{B}_p^T \boldsymbol{X}[n] + \boldsymbol{C}_p^T \boldsymbol{X}[n] \end{aligned} \tag{5}$$

where w_p is the weight of bias neuron. $\boldsymbol{A}_p$ is the weight vector for the recurrent portion, $\boldsymbol{B}_p$ is the weight matrix for the bilinear recurrent portion, and $\boldsymbol{C}_p$ is the weight vector for the feedforward portion and $p = 1, 2..., M$. More detailed information on the BLRNN and its learning algorithm can be found in [6,7].

4 Multiresolution-Based Learning Algorithm

The multiresolution-based learning algorithm attempts to improve the learning process by decomposing the original signal into a multiresolution representation. As stated in Section 2, the original signal is decomposed into a multiresolution representation using the wavelet transform. The representation of the signal at a resolution level j can be calculated as follows:

$$r_j = \begin{cases} x, & \text{if} \quad j = 0 \\ c_J + w_J + w_{J-1} + \cdots + w_j, & \text{if} j < J \\ c_J, & \text{if} \quad j = J \end{cases} \tag{6}$$

where r_j is the representation of the signal at resolution level j, J is the number of resolution levels, c_J is the scaling coefficients at resolution level J, x is the original signal, and w_j is the wavelet coefficients at resolution level j. Fig. 1 shows an example of different representations of a signal obtained using the wavelet transform, where r_0 is the original signal, and r_1, r_2, and r_3 are representations of the signal at resolution levels $1, 2$, and 3, respectively. The figure plots 100 samples from each representation of the signal for easy visualization.

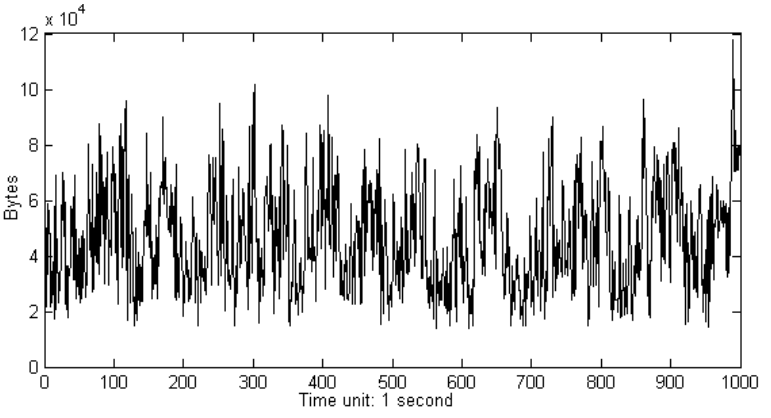


Fig. 3. Ethernet network traffic data over 1,000 seconds

The learning process is performed through J learning phases according to the number of resolution levels J:

$$L_J(r_J) \rightarrow L_{J-1}(r_{J-1}) \rightarrow \cdots \rightarrow L_0(r_0) \tag{7}$$

where $L_j(r_j)$ denotes the learning phase at resolution level j using representation r_j.

The learning phase at each resolution level j, $L_j(r_j)$, uses the representation signal r_j to update the network weights. The first learning phase $L_J(r_J)$ begins with randomly initialized network weights and the subsequent learning phase $L_j(r_j)$ begins with the updated weights from the previous learning phase $L_{j-1}(r_{j-1})$. It should be noted that only a single BLRNN model is employed to learn the information from the representation at different resolution levels. Fig. 2 shows the learning process using the multiresolution-based learning algorithm.

5 Experiments and Results

The proposed M-BLRNN is examined and evaluated in terms of its application to the long-term prediction of network traffic. Real-world Ethernet traffic data sets collected at Bellcore in August 1989 [11] are used to conduct experiments. The Ethernet traffic data set is network traffic data measured at each 0.01(s) over two normal hours of traffic corresponding to 10^6 samples of data. The data were downsampled with a time scale of 1(s), resulting in 10,000 data samples. Fig. 3 shows the first 1,000 samples from the network traffic data and Fig. 1 shows an example of different representations of the network traffic data used in the experiments.

In order to measure the long-term prediction performance, the normalized mean square error (NMSE) is employed. The NMSE is calculated by the following

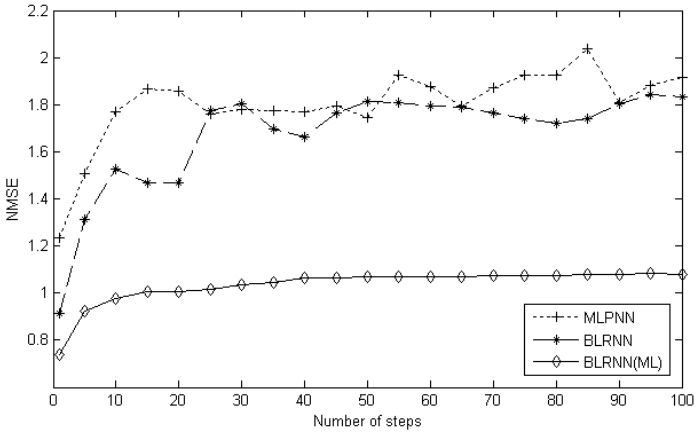


Fig. 4. Prediction performance for Ethernet network traffic data

formula:

$$NMSE = \frac{1}{\sigma^2 N} \sum_{n=1}^{N} (x_n - \hat{x}_n)^2$$

where x_n represents the true value of the sequence, $\hat{x}_n$ represents the predicted value, and σ represents the variance of the original sequence over the prediction duration N.

Fig. 4 shows the prediction performance over 100 steps of prediction for the traditional MLPNN, the BLRNN, and the proposed M-BLRNN in terms of the NMSE. The performance of the MLPNN is obtained using a MLPNN model with a structure of 24-10-1, where 24, 10, and 1 are the number of input neurons, hidden neurons, and output neurons, respectively. The performance of the BLRNN is obtained using a BLRNN model with a structure of 24-10-1 and 3 recursion lines. The result of the proposed M-BLRNN is obtained using a M-BLRNN model with a structure of 24-10-1, 3 recursion lines, and 3 resolution levels. The MLPNN and the BLRNN are trained with 3,000 iterations while the M-BLRNN is trained with 1,000 iterations at each learning phase. As can be seen from Fig. 4, the proposed M-BLRNN outperforms both the traditional MLPNN and the BLRNN. Moreover, when the number of steps increases, the performance of the traditional MLPNN and the BLRNN degrades significantly while the proposed M-BLRNN suffers from a minor degradation of performance. This implies that the proposed M-BLRNN is more robust than the traditional MLPNN and the BLRNN for long-term prediction of time series.

6 Conclusion

A Multiresolution-based BiLinear Recurrent Neural Network (M-BLRNN) for time series prediction is proposed in this paper. The proposed M-BLRNN employed the wavelet transform to decompose the signal into a multiresolution

representation. The learning process based on the multiresolution-based learning algorithm is performed by learning from each representation of the signal at each level of resolution. The proposed M-BLRNN is applied to the network traffic prediction. The experiments and results verified that the proposed M-BLRNN is more efficient than the traditional MLPNN and the BLRNN with respect to long-term prediction of time series. The promising results from this paper provide motivation to utilize the proposed M-BLRNN in other practical applications.

References

1. Kiruluta, A., Eizenman, M., Pasupathy, S.: Predictive Head Movement Tracking using a Kalman Filter. IEEE Trans. on Systems, Man and Cybernetics 27(2) (1997) 326-331
2. Han,M., Xi, J., Xu, S., Yin, F.-L.: Prediction of chaotic time series based on the recurrent predictor neural network. IEEE Trans. Signal Processing, 52 (2004) 3409-3416
3. Wang, L.P., Fu, X.J.: Data Mining with Computational Intelligence, Springer, Berlin (2005)
4. Leung, H., Lo, T., Wang, S.: Prediction of Noisy Chaotic Time Series using an Optimal Radial Basis Function Neural Network. IEEE Trans. on Neural Networks 12(5) (2001) 1163-1172
5. Connor, J.T., Martin, R.D., Atlas, L.E.: Recurrent Neural Networks and Robust Time Series Prediction. IEEE Trans. on Neural Networks 5(2) (1994) 240-254
6. Park, D.C., Zhu, Y.: Bilinear Recurrent Neural Network. IEEE ICNN, Vol. 3, (1994) 1459-1464.
7. Park, D.C., Jeong, T.K.: Complex Bilinear Recurrent Neural Network for Equalization of a Satellite Channel. IEEE Trans on Neural Network 13 (2002) 711-725.
8. Mallat, S.G.: A Theory for Multiresolution Signal Decomposition: the Wavelet Representation. IEEE Trans. Pattern Anal. Machine Intell. 11 (1989) 674-693.
9. Alarcon-Aquino, V., Barria, J.A.: Multiresolution FIR Neural-Network-Based Learning Algorithm Applied to Network Traffic Prediction. IEEE Trans. Sys. Man. and Cyber. PP(99) (2005) 1-13.
10. Renaud, O., Starck, J.L., Murtagh, F.: Wavelet-Based Combined Signal Filtering and Prediction. IEEE Trans. on Sys., Man, and Cyber. 35 (2005) 1241-251.
11. Leland,W.E.,Wilson,D.V.: High time-resolution measurement and analysis of LAN traffic: Implications for LAN interconnection, Proc. IEEE INFOCOM '91, (1991) 1360-1366

Query Expansion Using a Collection Dependent Probabilistic Latent Semantic Thesaurus

Laurence A.F. Park and Kotagiri Ramamohanarao

ARC Centre for Perceptive and Intelligent Machines in Complex Environments,
Department of Computer Science and Software Engineering,
The University of Melbourne, 3010, Australia
{lapark,rao}@csse.unimelb.edu.au

Abstract. Many queries on collections of text documents are too short to produce informative results. Automatic query expansion is a method of adding terms to the query without interaction from the user in order to obtain more refined results. In this investigation, we examine our novel automatic query expansion method using the probabilistic latent semantic thesaurus, which is based on probabilistic latent semantic analysis. We show how to construct the thesaurus by mining text documents for probabilistic term relationships, and we show that by using the latent semantic thesaurus, we can overcome many of the problems associated to latent semantic analysis on large document sets which were previously identified. Experiments using TREC document sets show that our term expansion method out performs the popular probabilistic pseudo-relevance feedback method by 7.3%.

1 Introduction

Short queries, consisting of only a few terms can be vague and hence cause an information retrieval system to return documents covering a broad number of topics which are not specific to the users information need. To assist the user, methods of query expansion have been developed, where the retrieval system adds terms to the short query in order to improve the precision of the results provided. This can be done with user interaction [6] or automatically without user interaction [8].

In this article, we will describe and examine our new method of automatic query expansion using a probabilistic latent semantic thesaurus. Our main contributions are: a generalisation model for ranking; and showing the probabilistic latent semantic thesaurus method is both efficient in storage and memory while yielding high precision. We show that our method of query expansion outperforms the popular BM25 pseudo-relevance feedback method by an average of 7.3% in terms of average reciprocal rank and also improves on our baseline BM25 by an average 8%.

This article will proceed as follows: In section 2, we briefly explain the information retrieval process and describes how the retrieval system can assist the user by adding to the query, this can be done with the local relevance feedback

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 224–235, 2007.

methods or the global thesaurus methods. In section 3, we further explain our new collection dependent thesaurus using probabilistic latent semantic analysis and show how to construct it. Experimental procedures and results are provided in section 4.

2 Query Term Expansion

Information retrieval systems are used to quickly assess whether a collection of unexamined text documents contains the information we desire. To the retrieval system, each text document is simply a sequence of terms. The query process requires a set of key terms to be supplied to the retrieval system, which are judged by the user as best describing their information need. Once the query is given, the retrieval system compares the query to the text documents and returns the top matching documents to the user. The user then evaluates whether the documents suit the information need. If the information need is met, the process is finished, but if it is not met, the user must ponder how to reword the query in order to obtain better results.

The problem with this process lies in the guess work involved in converting the information need into a set of query terms. The query can be expressed in many ways due to the versatility of our language. Unfortunately information retrieval systems use term matching to identify the documents relevant to the query, therefore to obtain the best results, the query must be expressed using the terms found in the document set. This implies that the user would require some knowledge of the content of the document collection to formulate a query. But as we mentioned, information retrieval systems are used to quickly assess whether a collection of unexamined text documents contains the information we desire, therefore we should not have to examine the document set in order to construct a query.

Rather than let the user manually examine the document set before querying, methods of term expansion have been derived that place the document analysis within the retrieval system. Term expansion is the process of adding terms to a query in order to create a query which is closer to the information need relative to the document set. The terms are chosen based on a similarity analysis of the query terms within the document set. To perform term expansion, we need a document-query scoring function ($S_q(d, Q)$) to rank documents based on the users query, a term scoring function ($S_t(t, Q)$) to select terms for the query expansion, and a document-expansion scoring function ($S_e(d, E)$) to rank documents based on the expansion terms. The final document score is a combination of the query and expansion term document scoring functions:

$$S(d, Q) = (1 - \alpha)S_q(d, Q) + \alpha S_e(d, E) \tag{1}$$

Note that using $\alpha = 0$ implies that there is no feedback used and $\alpha = 1$ implies that purely feedback is used. Typically, an α value of less than one is used to

put less emphasis on the expansion terms, so that they don't dominate the query. Each method we present will be based on the BM25 document scoring function, therefore they will all use the same document-query scoring function:

$$S_q(d,Q) = \sum_{t \in Q} w_{d,t} w_t \tag{2}$$

where

$$w_{d,t} = \frac{f_{d,t}(k_1+1)}{K + f_{d,t}} \qquad w_t = \log\left(\frac{N - f_t + 0.5}{f_t + 0.5}\right) \tag{3}$$

where $S_q(d,Q)$ is the score of document d based on the query Q, $f_{d,t}$ is the frequency of term t in document d, N is the number of documents in the collection, f_t is the number of documents containing term t, $K = k_1((1-b)+b\ dl/avdl)$, k_1 and b are constants, dl is the document length, and $avdl$ is the average document length.

In this section we will describe the two major methods of automatic term expansion called Pseudo-relevance feedback and Thesaurus expansion.

2.1 Pseudo-relevance Feedback

Interactive relevance feedback extends the involvement of the retrieval system in the information retrieval process to rely on user feedback. After the query has been supplied and the top matching documents have been calculated by the retrieval system, the system then proceeds by presenting the user with the matching documents and asks which are relevant to the query. Once the system receives the relevance information, it then continues by extracting terms from the set of relevant documents that will be included in the expanded query. After the query is formed, the retrieval system retrieves the documents that best match the new expanded query.

Pseudo-relevance feedback is a non-interactive version of the mentioned relevance feedback method. To remove the user interaction and hence speed up the query process, the retrieval system does not question the user about the relevance of the top matching documents to the query, but instead assumes that the documents that match the query are relevant. Terms are then extracted from this set of documents and used to build the expanded query.

A popular and effective pseudo-relevance feedback system comes from Robertson [4]. Given a query Q, the ten documents with the greatest $S_q(d,Q)$ are chosen for feedback. Using pseudo-relevance feedback, we assume that all ten documents are relevant to the query. Using terms from these ten documents, we must select a subset of terms to include in our expanded query. The selection process involves scoring each term and selecting the top terms to be included in our expanded query. The term scoring function used is:

$$S_t(t,Q) = f_{R,t} w_t \tag{4}$$

	dog	puppy	cat	kitten
dog	0.7	0.2	0.07	0.03
puppy	0.3	0.6	0.01	0.09
cat	0.06	0.04	0.7	0.2
kitten	0.02	0.08	0.6	0.3

Fig. 1. An example of a collection dependent thesaurus. All of the words found within the collection are listed with the probability of their relationship to each word. We can see that the words *dog* and *puppy* have a higher probability of being related to *dog* than the words *cat* and *kitten*.

where $f_{R,t}$ is the frequency of term t in the set of pseudo-relevant documents R. The terms with the greatest scores are then used to query the document collection using the document-expansion scoring function:

$$S_e(d, E) = S_q(d, E)$$

where E is the set of expansion terms. The final document score is calculated using equation 1 where $\alpha = 1/3$.

2.2 Collection Dependent Thesaurus Expansion

Relevance feedback is a local query expansion because it uses only a subset of the document set to calculate the set of expansion terms. Thesaurus expansion is a global query expansion because it makes use of the whole document set when calculating the set of expansion terms.

A thesaurus is a collection of words, where each word has an associated set of words that are related to it. Many thesauruses have been constructed manually, and can be used by those that have an understanding of the language. A typical entry in a manually built thesaurus contains the desired word and sets of related words grouped into different senses. To effectively find related words, we must know which sense to choose. Unfortunately, this is not an easy task for a machine and hence machine word-sense disambiguation is an active field in the area of computer science and linguistics.

A collection dependent thesaurus is one that is automatically built using the term frequencies found with a document collection. Since the thesaurus is document collection dependent, any word relationships found will be based on documents that can be retrieved. A collection dependent thesaurus is a square table that contains all of the words found within the collection and their relationship to each other. Each element of the table contains the probability of a word being related to another. A thesaurus built using a very small document set is shown in figure 1.

The information retrieval process using a collection dependent thesaurus is very similar to that of pseudo-relevance feedback. The difference being that the initial document retrieval step to obtain the candidates for the query expansion does not have to be performed since we already have a table of term relationships.

To obtain our set of expansion terms, we must choose terms that obtain the greatest score using the score function:

$$S_t(\tau, Q) = \sum_{t \in Q} P(\tau|t) w_t \tag{5}$$

where the thesaurus element $P(\tau|t)$ is the probability of term τ being related to term t and Q is the set of query terms. We can see that equation 5 gives higher scores to those terms having a higher probability of being related to the query. The document-expansion scores are calculated using:

$$S_e(d, E) = \sum_{t \in E} w_{d,t} S_t(t, Q) \tag{6}$$

where E is the set of terms with the greatest term score. The document-expansion score is the same as the document-query score except for the term weight (w_t) being replaced by the term score.

3 Probabilistic Latent Semantic Thesaurus

The previous section showed us how to use term expansion within the information retrieval process. We described how we can use a collection dependent thesaurus to obtain term relationships for our query expansion, but we did not explain how the thesaurus was built and the probabilistic values were obtained. In this section we will examine how to construct the thesaurus using probabilistic latent semantic analysis; but before we do, we will explain the concept of latent semantic analysis and show how it can be used to obtain term relationships.

3.1 Latent Semantic Analysis

Probabilistic [4] and vector space methods [1] of information retrieval base the document score on the occurrence of the query terms within the document. This implies that if the query terms do not appear within the document, it obtains a score of zero and is considered irrelevant to the query. Any terms that are not related to the query are ignored, even though their occurrence within a document could infer relevance.

Rather than observing the occurrence of terms, the method of latent semantic analysis [2] observes the occurrence of topics, where a topic is a set of related words. Its use becomes more intuitive once we observe the following document creation models. The document creation model with term based comparisons uses the following sequence:

1. the author starts by having an idea that needs to be conveyed
2. the idea is put to paper by choosing specific words
3. if other words were chosen during the writing process, the written document would not convey the same idea

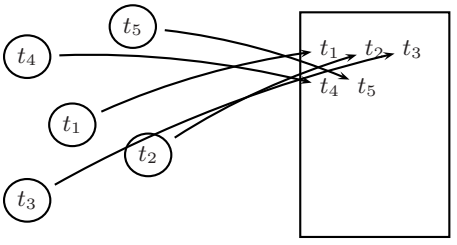


Fig. 2. A naïve document creation model. The author chooses specific terms for the document. If different terms are chosen the document will not convey the same message. This is the model used by retrieval systems that assume all terms are independent of each other.

We can see that this model (shown in figure 2) is not correct, since our language allows us to project the same idea using different words. This document creation model is projected in the probabilistic and vector space methods of information retrieval; if query terms do not appear in a document, the document is considered irrelevant even if it does contain related terms.

A more appropriate document creation model (shown in figure 3) uses the following sequence:

1. the author begins by having an idea that needs to be conveyed
2. the author chooses specific topics to convey the idea
3. the idea is put to paper by choosing words that are associated to each of the chosen topics
4. if different topics were chosen during the writing process, the written document would not convey the same idea

In this case, two documents containing different terms could project the same idea if the terms were associated to the same topics. This more realistic model takes into account the synonymy found in modern day languages by comparing topics rather than terms. Latent semantic analysis is the process of discovering these topics and their relationship to the set of terms.

3.2 Probabilistic Latent Semantic Analysis

Probabilistic latent semantic analysis (PLSA) [3] is the process of discovering the topics within a document set using probabilistic means. In this section we will describe how we can use it in our retrieval system.

The probability of choosing a specific term from a specific document within our document collection is given as:

$$P(d,t) = \frac{f_{d,t}}{\sum_{d \in D} \sum_{t \in T} f_{d,t}} \tag{7}$$

where D and T are the set of documents and terms respectively. Given the set of topics Z, we are able to form the following relationship between the set of documents D and set of terms T using Bayesian analysis:

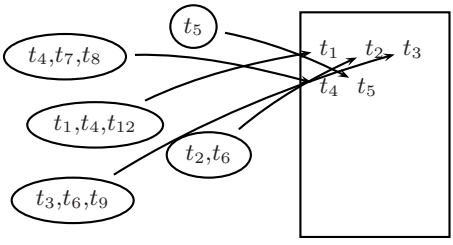


Fig. 3. The latent semantic analysis document model. The author chooses specific topics for the document and then chooses a term from the topic to place in the document. This model implies that documents containing different terms can convey the same message, as long as the replacement terms are associated to the same topic.

$$\begin{aligned} P(d,t) &= P(d|t)P(t) \\ &= \sum_{z \in Z} P(d|z)P(z|t)P(t) \\ &= \sum_{z \in Z} P(d|z)P(t|z)P(z) \end{aligned} \tag{8}$$

where d and t are conditionally independent given z, $d \in D$, and $t \in T$. Since the $P(d|z)$, $P(t|z)$ and $P(z)$ values are unknown, we must obtain the best fit that satisfies equation 8. Using expectation maximisation, we are able to obtain an approximation of the unknown probabilities $P(d|z)$, $P(t|z)$ and $P(z)$ for all d, t and z in D, T and Z respectively.

Before we obtain the probabilities, we must choose the size of Z. By choosing a small number of elements for Z (much less than the number of documents and terms in the document set), we make sure that our model is not over fitted. A small number of $z \in Z$ implies that there will be a small number of topics and hence, the documents and terms will cluster into topic sets.

3.3 Building the Thesaurus

Using probabilistic latent semantic analysis, we have obtained the probabilistic relationships between the topics and terms, and the topics and documents. To build a thesaurus, we need to calculate the probabilistic relationships between each of the terms. To do so, we have derived the following relationship:

$$\begin{aligned} P(t_x|t_y) &= \sum_{z \in Z} P(t_x|z)P(z|t_y) \\ &= \sum_{z \in Z} \frac{P(t_x|z)P(t_y|z)P(z)}{P(t_y)} \end{aligned} \tag{9}$$

where t_x and t_y are conditionally independent given z. To obtain the probabilistic term relationship values, we use the $P(t|z)$ and $P(z)$ probabilities obtained from PLSA, and:

$$P(t) = \frac{\sum_{d \in D} f_{d,t}}{\sum_{d \in D} \sum_{t \in T} f_{d,t}} \tag{10}$$

Once we have obtained the probabilistic term relationships, we store the values in a fast lookup index so that they can easily be accessed at the query time.

3.4 Thesaurus Performance

Every term will have a probabilistic relationship to every other term, therefore our thesaurus will be a $|T| \times |T|$ table of non-zero floating point values, where $|T|$ is the cardinality of T. This implies that the thesaurus will be very large for large document sets. This storage problem also exists in probabilistic latent semantic indexing.

Fortunately, our use of a thesaurus means that we do not need to store all of the terms. Each of the term relationships is based on the term samples found within the document set. We have shown that terms that are under-sampled (found in only a few documents) will not produce proper relationships [5], therefore it seems fitting to ignore the under-sampled terms. If we ignore all terms that are found in no more than N documents, we will remove a significant amount of terms due to the occurrence of terms following the Zipf distribution. By removing these terms, we are choosing to keep the terms that appear in at least N documents, which is directly related to the term weight (w_t). By choosing $N = 50$ [5], we reduce the storage required from a 4 gigabyte index to a 21 megabyte thesaurus.

We stated that there is a relationship between every term in the thesaurus. Therefore if we were to use any number of query terms, the expanded query would contain every term in the thesaurus. The query processing time is proportional to the number of query terms, therefore including every term in the query would lead to a very expensive query process. This query processing speed problem exists in every query expansion method and can be resolved by choosing only those terms that have the greatest term score ($S_t(t, Q)$) for the expansion. By doing so, we receive the top M related terms to the query. From this we can see that the query expansion size leads to a trade off between system precision and query speed.

As for the query speed, the difference between the pseudo-relevance feedback and thesaurus is the term expansion method. The former requires two lookups of a sparse index, while the latter requires one lookup of a dense index. This leads to similar query processing times.

4 Experiments

To examine the performance of our collection dependent thesaurus using probabilistic latent semantic analysis, we have run experiments on two well known document collections. The first document collection is the Associated Press articles from TREC disk-1 (AP1) containing 84,678 documents, the second is the

set of Associated Press articles from TREC disk-2 (AP2) containing 79,919 documents. For each of the document sets, we used the titles of queries 51 to 200 and the associated relevance judgements from TREC-1, 2 and 3.

Our experiments compared the increase in precision due to query expansion at various levels of expansion. We reported results using our probabilistic latent semantic thesaurus (PLST), pseudo-relevance feedback (PRFB) and a term co-occurrence thesaurus (COT). The increase in precision shown is compared to BM25 with no query expansion. The term co-occurrence thesaurus uses the thesaurus method found in section 2.2, where:

$$P(\tau|t) = \frac{\sum_{d \in D} f_{d,\tau} f_{d,t}}{\sum_{\tau \in T} \sum_{d \in D} f_{d,\tau} f_{d,t}} \tag{11}$$

where T is the set of terms and D is the set of documents.

We built a thesaurus for each of the document collections using the following suggested parameters [5]. The thesaurus included all terms that were found in at least 50 documents, the mixing parameter was set to $\alpha = 0.6$, and 100 topics were calculated. To compare our thesaurus method we also ran experiments using pseudo-relevance feedback on the same document sets using the suggested parameters of $\alpha = 0.25$ and using the top ten documents for feedback [4]. Within the BM25 model, we used the parameters $k_1 = 1.2$ and $b = 0.75$ [4].

The precision at 10 documents and average reciprocal rank increases are shown in figures 4 and 5 respectively. The increases are shown with respect to the BM25 (without expansion) ranking function. The two measures reported, measure the system for different uses. The reciprocal rank of a query is the inverse of the rank of the first relevant document (e.g. if the first relevant document is ranked third, the reciprocal rank is 1/3). The average reciprocal rank is the average of all reciprocal ranks from each query. If we are using the retrieval system to find one document, we would use this value to measure the system. Precision at 10 documents is the average number of relevant documents found in those that the system ranks in the top ten. We would use this measure if we wanted a few relevant documents.

The results show that the PLST outperforms the COT for all levels of expansion using both measures. We can see that our PLST method outperforms the pseudo-relevance feedback method in average reciprocal rank. In fact, we can see on both data sets that applying any expansion PRFB reduces the ARR. If we observe the precision at 10 documents, we find that PRFB provides better precision for low expansion sizes and PLST provides better precision for higher expansion sizes. If we use typical expansion sizes of 20 terms to PRFB and 100 terms for PLST, we find that PLST provides an average increase of 7.3% in ARR and 0.2% increase in prec10. This implies that for the typical Web surfer who only wants one relevant document, PLST is the better query expansion method to use, while for someone who wants a many relevant pages in the first ten ranked documents, either PLST or PRFB could be used.

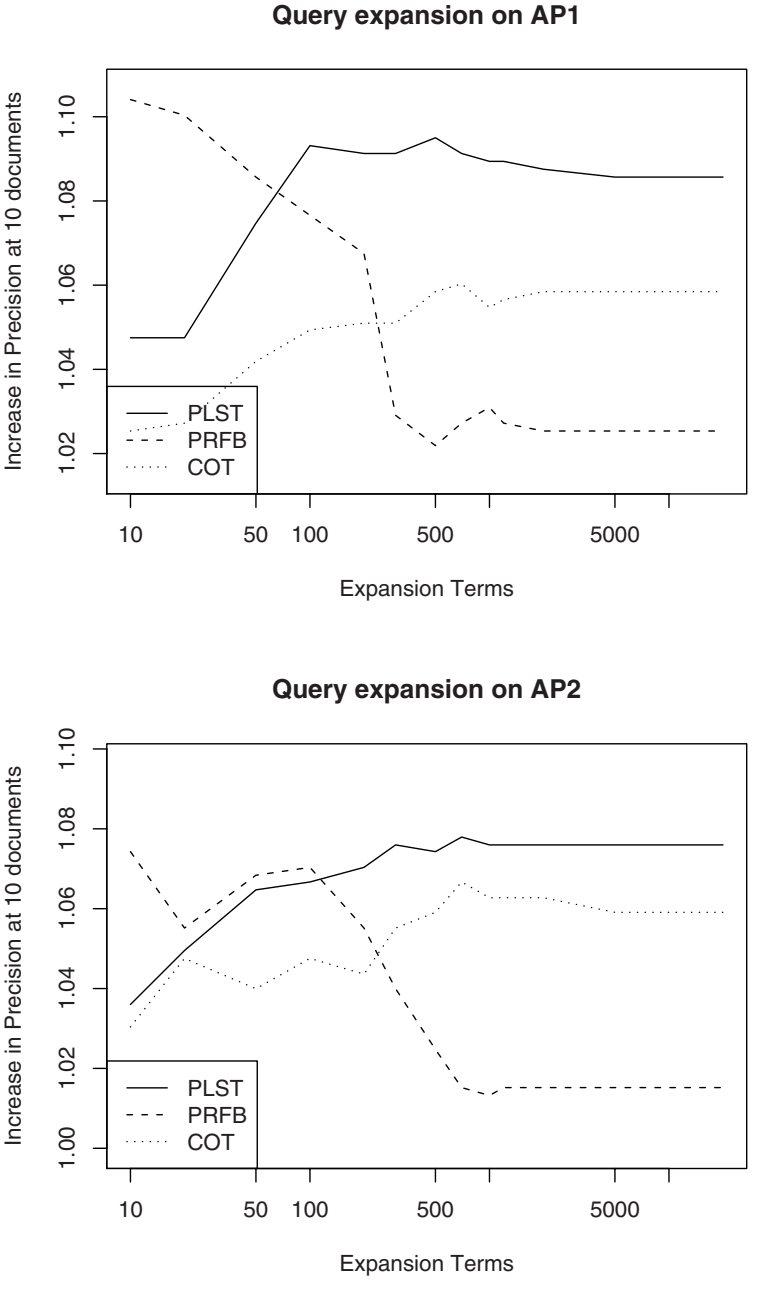


Fig. 4. A comparison of the increase in precision after 10 documents (prec10) due to query expansion of our collection dependent thesaurus using probabilistic latent semantic analysis (PLST) against pseudo-relevance feedback (PRFB) and a term co-occurrence thesaurus (COT) on the AP1 and AP2 document sets. The baseline BM25 (without expansion) precision after 10 documents is 0.3747 for AP1 and 0.3554 for AP2.

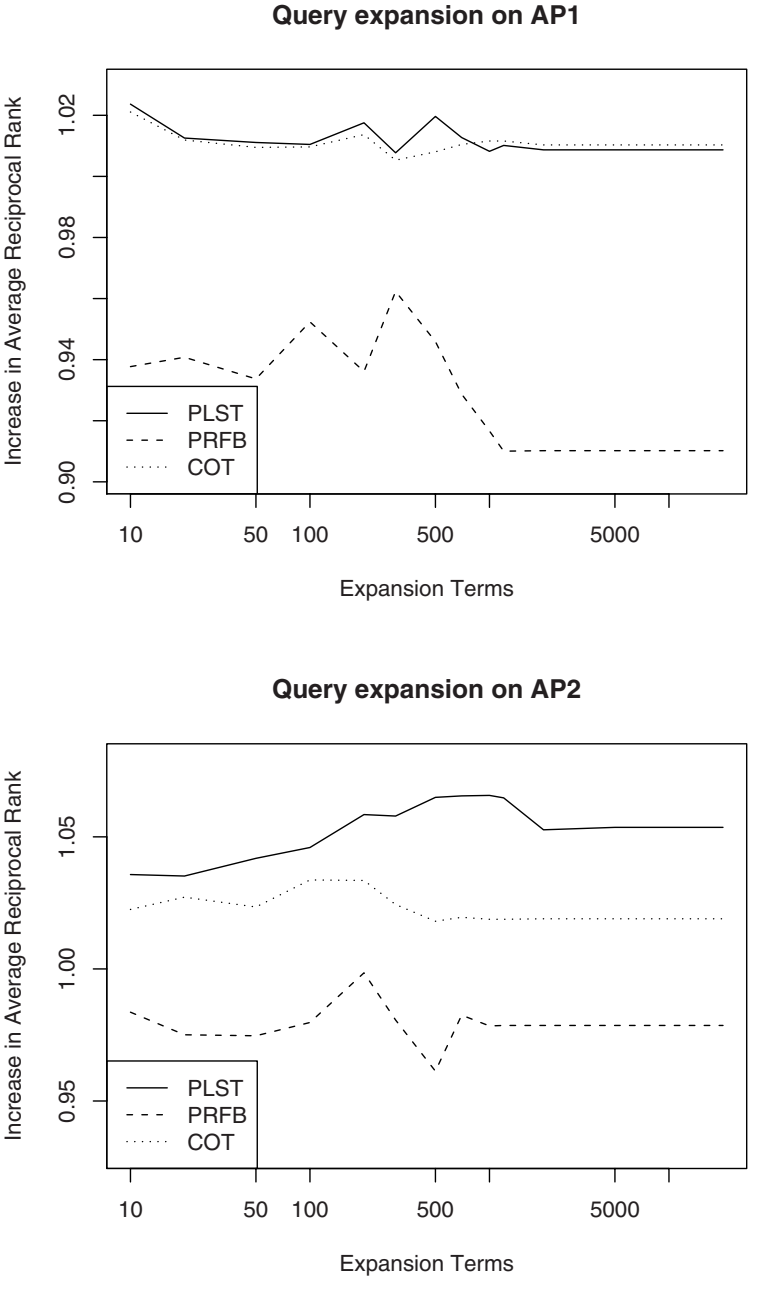


Fig. 5. A comparison of the increase in average reciprocal rank (ARR) due to query expansion of our collection dependent thesaurus using probabilistic latent semantic analysis (PLST) against pseudo-relevance feedback (PRFB) and a term co-occurrence thesaurus (COT) on the AP1 and AP2 document sets. The baseline BM25 (without expansion) average reciprocal rank is 0.6214 for AP1 and 0.5374 for AP2.

5 Conclusion

Automatic query expansion is a method of adding terms to the query without interaction from the user in order to obtain more refined results. We have presented our new method of automatic query expansion using a collection dependent thesaurus built with probabilistic latent semantic analysis. We have shown how to build the thesaurus using probabilistic latent semantic term relationships, and we have shown how to efficiently query the thesaurus in order to expand our query.

Experiments were performed and compared to the popular pseudo-relevance feedback using the BM25 weighting scheme and a term co-occurrence thesaurus. The results showed that our probabilistic latent semantic thesaurus outperformed the term co-occurrence thesaurus for all levels of recall and the pseudo-relevance feedback retrieval by an average 7.3% when one relevant document was desired, and an average 0.2% when observing the top ten ranked documents. This implies that a probabilistic latent semantic thesaurus would be the query expansion choice for the typical Web surfer.

References

1. Chris Buckley and Janet Walz. SMART in TREC 8. In Voorhees and Harman [7], pages 577–582.
2. Susan T. Dumais. Latent semantic indexing (lsi): Trec-3 report. In Donna Harman, editor, *The Third Text REtrieval Conference (TREC-3)*, pages 219–230, Gaithersburg, Md. 20899, March 1994. National Institute of Standards and Technology Special Publication 500-226.
3. Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM Press, 1999.
4. K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments, part 2. *Information Processing and Management*, 36(6):809—840, 2000.
5. Laurence A. F. Park and Kotagiri Ramamohanarao. Hybrid pre-query term expansion using latent semantic analysis. In Rajeev Rastogi, Katharina Morik, Max Bramer, and Xindong Wu, editors, *The Fourth IEEE International Conference on Data Mining*, pages 178–185, Los Alamitos, California, November 2004. IEEE Computer Society.
6. Xuehua Shen and ChengXiang Zhai. Active feedback in ad hoc information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–66, New York, NY, USA, 2005. ACM Press.
7. Ellen M. Voorhees and Donna K. Harman, editors. *The Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Md. 20899, November 1999. National Institute of Standards and Technology Special Publication 500-246, Department of Commerce, National Institute of Standards and Technology.
8. Ryen W. White, Ian Ruthven, and Joemon M. Jose. A study of factors affecting the utility of implicit relevance feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–42, New York, NY, USA, 2005. ACM Press.

Scaling Up Semi-supervised Learning: An Efficient and Effective LLGC Variant

Bernhard Pfahringer[1], Claire Leschi[2], and Peter Reutemann[1]

[1] Department of Computer Science, University of Waikato, Hamilton, New Zealand
[2] INSA Lyon, France

Abstract. Domains like text classification can easily supply large amounts of unlabeled data, but labeling itself is expensive. Semi-supervised learning tries to exploit this abundance of unlabeled training data to improve classification. Unfortunately most of the theoretically well-founded algorithms that have been described in recent years are cubic or worse in the total number of both labeled and unlabeled training examples. In this paper we apply modifications to the standard LLGC algorithm to improve efficiency to a point where we can handle datasets with hundreds of thousands of training data. The modifications are priming of the unlabeled data, and most importantly, sparsification of the similarity matrix. We report promising results on large text classification problems.

1 Introduction

Semi-supervised learning (and transduction) addresses the problem of learning from both labeled and unlabeled data. In recent years, this problem has generated a lot of interest among the Machine Learning community [26,38]. This learning paradigm is motivated by both practical and theoretical issues. Indeed, it provides a very interesting framework to up-to-date application domains such as web categorization (e.g. [35]), text classification (e.g. [22,15,17]), camera image classification (e.g. [3,25]), or computational biology (e.g. [31]). More gerally, it is of high interest in all domains in which one can easily get huge collections of data but labeling this data is expensive and time consuming, needs the availability of human experts, or even is infeasible. Moreover, it has been shown experimentally that, under certains conditions, the use of a small set of labeled data together with a large supplementary of unlabeled data allows the classifiers to learn a better hypothesis, and thus significantly improve the generalization performance of the supervised learning algorithms. Thus, one should sum up transductive learning as "less human effort and better accuracy". However, as has been noted by Seeger [26], issues in semi-supervised learning have to be addressed using (probably) genuinely new ideas.

Most of the semi-supervised learning approaches use the labeled and unlabeled data simultaneously or at least in close collaboration. Roughly speaking, the unlabeled data provides information about the structure of the domain, i.e. helps to capture the underlying distribution of the data, whereas the labeled data identifies the classification task within this structure. The challenge for

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 236–247, 2007.

the algorithms can be viewed as realizing a kind of trade-off between robustness and information gain [26]. To make use of unlabeled data, one must make assumptions, either implicitely or explicitely. As reported in [34], the key to semi-supervised learning is the prior assumption of consistency, that allows for exploiting the geometric structure of the data distribution. This assumption relies on a local and/or global statement(s). The former one (also shared by most of the supervised learning algorithms) means that nearby data points should belong to the same class. The later one, called cluster assumption, states that the decision boundary should lie in regions of low data density. Then, points which are connected by a path through regions of high data density have the same label. A common approach to take into account the assumption of consistency is to design an objective function which is smooth enough w.r.t. the intrinsic structure revealed by known labeled and unlabeled data.

Early methods in transductive learning were using mixture models (in which each mixture component should be associated with a class) and extensions of the EM algorithm [22]. More recent approaches belong to the following categories: self-training, co-training, transductive SVMs, split learning and graph-based methods. In the self-training approach, a classifier is trained on the labeled data and then used to classify the unlabeled ones. The most confident (now labeled) unlabeled points are added to the training set, together with their predictive labels, and the process is repeated until convergence [32,25]. The approaches related to co-training [7,17] build on the hypothesis that the features describing the objects can be divided in two subsets such that each of them is sufficient to train a good classifier, and the two sets are conditionally independent given the classes. Two classifiers are iteratively trained, each on one set, and they teach each other with the few unlabeled data (and their predictive labels) they feel more confident with. The transductive SVMs [29,15] are a "natural" extension of SVMs to the semi-supervised learning scheme. They aim at finding a labeling of the unlabeled data so that the decision boundary has a maximum margin on the original labeled data and on the (newly labeled) unlabeled data. Another category of methods, called split learning algorithms, represent an extreme alternative using the unlabeled and labeled data in two different phases of the learning process [23]. As stated by Ando and Zhang [1], the basic idea is to learn good functional structures using the unlabeled data as a modeling tool, and then the labeled data is used for supervised learning based on these structures. A detailed presentation of all these approaches is beyond the scope of this paper. In the following, we will focus on graph-based methods which are more directly related to the Local and Global Consistency (LLGC) algorithm [34] for which we are proposing some improvements.

Graph-based methods attempt to capture the underlying structure of the data within a graph whose vertices are the available data (both labeled and unlabeled) and whose (possibly weighted) edges encode the pairwise relationships among this data. As noticed in [33], examples of recent work in that direction include Markov random walks [28], cluster kernels [9], regularization on graphs [27,34] and directed graphs [35]. The graph is most often fully connected. Nevertheless, if

sparsity is desired, the pairwise relationships between vertices can reflect a nearest neighbor property, either thresholding the degree(k-NN) or the distance (ϵ-NN). The learning problem on graphs can generally be thought of as estimating a classifying function f which should be close to a given function y on the labeled data and smooth on the whole graph [34]. For most of the graph-based methods, this can be formally expressed in a regularization framework [38] where the first term is a loss function and the second term a regularizer. The so-defined cost (or energy) function should be minimized on the whole graph by means of (iterative) tuning of the edges values. Consequently, different graph-based methods mainly vary by the choice of the loss function and the regularizer [38]. For example, the work on graph cuts [6] minimizes the cost of a cut in the graph for a two-class problem, while [16] minimizes the normalized cut cost and [39,34] minimize a quadratic cost. As noticed in [38], these differences are not actually crucial. What is far more important is the construction and the quality of the graph, which should reflect domain knowledge through the similarity function which is used to assign edges (and their weights). One can find a discussion of that issue in [38,3] Other important issues such as consistency and scalability of semi-supervised learning methods are also discussed in [37].

2 Related Work

The LLGC method of Zhou et al. [34] is a graph-based approach which addresses the semi-supervised learning problem as designing a function f that satisfies both the local and global consistency assumptions. The graph G is fully connected, with no self-loop. The edges of G are weighted with a positive and symmetric function w which represents a pairwise relationships between the vertices. This function is further normalized w.r.t. the conditions of convergernce of the algorithm [21,9]. The goal is to label the unlabeled data. According to Zhou et al., the key point of the method is to let every point iteratively spread its label information to its neighbors until a global state is reached. Thus, looking at LLGC as an iterative process, one can intuitively understand the iteration as the process of information diffusion on graphs [18]. The weights are scaled by a parameter σ for propagation. During each iteration, each point receives the information from its neighbor and also retains its initial information. A parameter α allows to adjust the relative amount of information provided by the neighbors and the initial one. When convergence is reached, each unlabeled point is assigned the label of the class it has received most information for during the iteration process. One can also consider the LLGC method through the regularization framework. Then, the first term of the cost function Q(f) is a fitting constraint that binds f to stay close to the initial label assignment. The second term is a smoothness constraint that maintains local consistency. The global consistency is maintained by using a parameter μ which yields a balance between the two terms.

As stated by Zhou et al., the closest related graph-based approach to LLGC is the method using Gaussian random fields and harmonic functions presented in [39]. In this method, the label propagation is formalized in a probabilistic

framework. The probability distribution assigned to the classification function f is a Gaussian random field defined on the graph. This function is constrained to give their initial labels to labeled data. In terms of regularization network, this approach can be viewed as having a quadratic loss function with infinite weight, so that the labeled data are clamped, and a regularizer based on the graph Laplacian [38]. The minimization of the cost function results in an harmoninc function. In [14], the LLGC method and the Gaussian Random Field Model (GRFM) are further compared to each other and to the Low Density Separation (LDS) method of Chapelle and Zien [10]. T. Huang and V. Kecman notice that both algorithms are manifold-like methods, and have the similar property of searching the class boundary in the low density region (and in this respect they have similarity with the Gradient Transductive SVMs [10] too). LLGC has been recently extended to clustering and ranking problems. Relying on the fact that LLGC has demonstrated impressive performance on relatively complex manifold structures, the authors in [8] propose a new clustering algorithm which builds upon the LLGC method. They claim that LLGC naturally leads to an optimization framework that picks clusters on manifold by minimizing the mean distance between points inside the clsuters while maximizing the mean distance between points in different clusters. Moreover, they show that this framework is able to: (i) simultaneously optimize all learning parameters, (ii) pick the optimal number of clusters,(iii) allow easy detection of both global outliers and outliers within clusters, and can also be used to add previously unseen points to clusters without re-learning the original cluster model. Similarly, in [30], A. Vinueza and G.Z. Grudic show that LLGC performs at least as well as the best known outlier detection algorithm, and can predict class outliers not only for training points but also for points introduced after training. Zhou et al. in [36] propose a simple universal ranking algorithm derived from LLGC, for data lying in an Euclidean space and show that this algorithm is superior to local methods which rank data simply by pairwise Euclidean distances or inner products. Aso note that for large scale real world problems they prefer to use the iterative version of the algorithm instead of the closed form based on matrix inversion. Empirically, usually a small number of iterations seem sufficient to yield high quality ranking results.

In the following we propose extension on LLGC to cope with the computational complexity, to broaden its range of applicability, and to improve its predictive accuracy. As reported in [37], the complexity of many graph-based methods is close to $O(n^3)$. Speed-up improvements have been proposed, for example in [20,11,40,33,13], but their effectiveness has not yet been shown for large real-world problems. Section 3 will give the definition of the original LLGC algorithm and detail our extensions. In Section 4 we will support our claims with experiments on textual data. Finally Section 5 summarizes and provides directions for future work.

3 Original LLGC and Extensions

A standard semi-supervised learning algorithm is the so-called LLGC algorithm [34], which tries to balance two potentially conflicting goals: locally, similar

examples should have similar class labels, and globally, the predicted labels should agree well with the given training labels. The way LLGC achieves that can intuitively be seen as the steady state of a random walk on the weighted graph given by the pairwise similarities of all instances, both labeled and unlabeled ones. At each step each example passes on its current probability distribution to all other instances, were distributions are weighted by the respective similarities.

In detail, the LLGC works as follows:

1. Set up an affinity matrix A, where $A_{ij} = e^{-\frac{||x_i - x_j||^2}{\sigma^2}}$ for $i \neq j$, and $A_{ii} = 0$.
2. Symmetrically normalize A yielding S, i.e. $S = D^{-0.5}AD^{-0.5}$ where D is a diagonal matrix with $D(i,i)$ being the sum of the i-th row of A, which is also the sum of the i-th column of A, as A is a symmetrical matrix.
3. Setup matrix Y as a $n * k$ matrix, where n is the number of examples and k is the number of class values. Set $Y_{ik} = 1$, if the class value of example i is k. All other entries are zero, i.e. unlabeled examples are represented by all-zero rows.
4. Initialise $F(0) = Y$, i.e. start with the given labels.
5. Repeat $F(t+1) = \alpha * S * F(t) + (1 - \alpha) * Y$ until F converges. α is a parameter to be specified by the user in the range $[0, 1]$. High values of α focus the process on the propagation of the sums of the neighbourhood, i.e. the local consistency, where low values put more emphasis onto the constant injection of the originally given labels, and thereby focus more on global consistency.

The seminal LLGC paper [34] proves that this iteration converges to:

$$F^* = (1 - \alpha) * (I - \alpha * S)^{-1} * Y$$

The normalized rows of F^* can be interpreted as class probability distributions for every example. The necessary conditions for convergence are that $0 \leq \alpha \leq 1$ holds, and that all eigenvalues of S are inside $[-1, 1]$.

Before introducing the extensions designed in order to achieve the goals mentioned in the previous section, let us notice the following: LLGC's notion of similarity is based on RBF kernels, which are general and work well for a range of applications. But they are not always the best approach for computing similarity. For text classification problems usually the so-called cosine similarity measure is the method of choice, likewise other domains have there preferred different similarity measures. Generally, it is possible to replace the RBF kernel in the computation of the affinity matrix with any arbitrary kernel function, as long as one can show that the eigenvalues of S will still be within $[-1, 1]$, thus guaranteeing convergence of the algorithm. One way of achieving this is to use "normalized" kernels. Any kernel k can be normalized like so ([2]):

$$k_{norm}(x, y) = \frac{k(x, y)}{\sqrt{k(x, x) * k(y, y)}}$$

As the experiments reported below concern text classification problems, for which the so-called cosine similarity measure is the method of choice (likewise

other domains have there preferred different similarity measures), we will employ this similarity measure (which is already normalized) instead of RBF kernels.

3.1 Reduce Computational Complexity by Sparsifying A

The main source of complexity in the LLGC algorithm is the affinity matrix. It needs $O(n^2)$ memory and the matrix inversion necessary for computing the closed form needs, depending on the algorithm used, roughly $0(n^{2.7})$ time, where n is the number of examples. If there are only a few thousand examples in total (both labeled and unlabeled), this is feasible. But we also want to work with 10^5 examples and even more. In such a setting even only storing the affinity matrix in main memory becomes impossible, let alone computing the matrix inversion.

Our approach to sparsification is based on the insight that most values in the original affinity matrix are very close to zero anyways. Consequently we enforce sparsity by only allowing the k nearest neighbours of each example to supply a non-zero affinity value. Typical well-performing values for k range from a few dozen to a hundred. There is one caveat here: kNN is not a symmetrical relationship, but the affinity matrix has to be symmetrical. It is easy to repair this shortcoming in a post-processing step after the sparse affinity matrix has been generated: simply add all "missing" entries. In the worst case this will at most double the number of non-zero entries in A. Therefore the memory complexity of LLGC is reduced from $O(n^2)$ to a mere $O(k * n)$, which for small enough values of k allows to deal with even millions of examples. Additionally, when using the iterative version of the algorithm to compute F^*, the computational complexity is reduced to $O(k * n * n_{iterations})$, which is a significant improvement in speed over the original formulation, especially as the number of iterations needed to achieve (de facto) convergence is usually rather low. E.g. even after only ten iterations usually most labels do not change any more.

Computing the sparse affinity matrix is still $O(n^2)$ timewise, but for cases where $n \geq 5000$ we use a hierarchical clustering-based approximation, which is $O(n * log(n))$. Alternatively, there is currently a lot of research going on trying to speed-up nearest-neighbour queries based on smart data-structures, e.g. kD-trees, or cover trees[4].

3.2 Allow Pre-labeling of the Unlabeled Data

LLGC starts with all-zero class-distributions for the unlabeled data. We allow pre-labeling by using class-distributions for unlabeled data that have been computed in some way using the training data:

$$Y_{ij} = prob_j(classifier_{labeledData}(x_i))$$

where $classifier_{labeledData}$ is some classifier that has been trained on just the labeled subset of data given. For text mining experiments as described below this is usually a linear support vector machine. There are at least two arguments for allowing this pre-labeling (or priming) of the class probability distributions

inside LLGC. A pragmatic argument is that simply in all experiments we have performed we uniformly achieve better final results when using priming. We suspect that this might not be true in extreme cases when the number of labeled examples is very small, and therefore any classifier trained on such a small set of examples will necessarily be rather unreliable. There is also a second more fundamental argument in favour of priming. Due to the sparsification of the affinity matrix, which in its non-sparse version describes a fully-connected, though weighted graph, this graph might be split into several isolated subgraphs. Some of these subgraphs may not contain any labeled points anymore. Therefore the propagation algorithm would have no information left to propagate, and thus simply return all-zero distributions for any example in such a neighbourhood. Priming resolves this issue in a principled way.

One potential problem with priming is the fact that the predicted labels might be less reliable than the explicitly given labels. In a different and much simpler algorithm[12] for semi-supervised learning this problem was solved by using different weights for labeled and unlabeled examples. When the weights reflected the ratio of labeled to unlabeled examples, then usually predictive accuracy was satisfactory. In a similar spirit we introduce a second parameter β, which scales down the initial predictions for unlabeled data in the primed LLGC algorithm:

$$Y_{ij} = \beta * prob_j(classifier_{labeledData}(x_i))$$

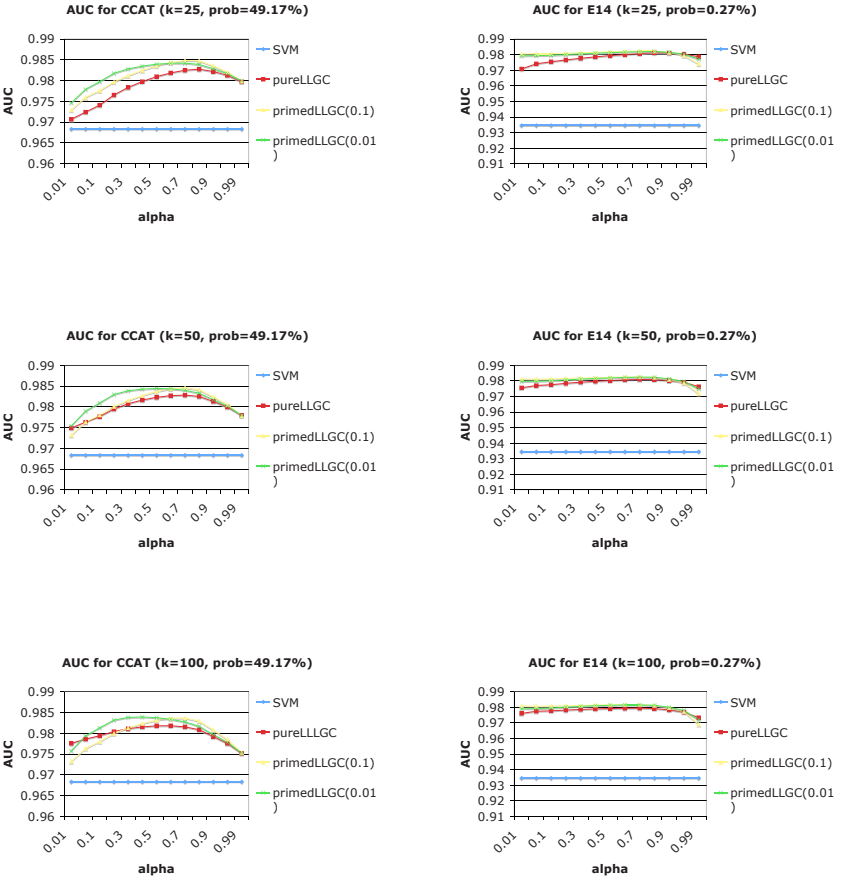
if x_i is an unlabeled example. In the experiments reported below we usually find that values for β as chosen by cross-validation are reasonably close to the value that the ratio-heuristic would suggest.

4 Experiments

In this section we evaluate the extended LLGC algorithm on text classification problems by comparing it to a standard linear support vector machine. As we cannot compare to the original LLGC algorithm for computational reasons (see previous section for details), we at least include both a "pure" version which uses only sparsification and the cosine-similarity, and the "primed" version, which uses the labels as predicted by the linear support vector machine to initialise the class distributions for the unlabeled data. As explained in the previous section, we down-weigh these pre-labels by setting $\beta = 0.1$ and also by $\beta = 0.01$, to see how sensitive the algorithm is with respect to β. We also investigate differently sized neighbourhoods of sizes 25, 50, and 100.

The dataset we use for this comparison is the recently released large and cleaned-up Reuters corpus called RCV12[19]. We use a predefined set of 23149 labeled examples as proper training data, and another 199328 examples as the unlabeled or test data. Therefore we have training labels for slightly more than 10% of the data. RCV12 defines hundreds of overlapping categories or labels for the data. We have run experiments on the 80 largest categories, treating each category separately as a binary prediction problem. To evaluate we have chosen AUC (area under the ROC curve) which recently has become very popular

Table 1. AUC values for categories CCAT and E14, SVM, pure LLGC, and primed LLGC, for a range of α values. CCAT is the largest category, E14 a rather small one.



especially for text classification [5], as it is independent of a specific threshold. Secondly, as some typical text classification tasks can also be cast as ranking tasks (e.g. the separation of spam email from proper email messages), AUC seems especially appropriate for such tasks, as it provides a measure for how much better the ranking computed by some algorithm is over a random ranking.

As there is not enough space to present the results for all these 80 categories here, we have selected only two (CCAT and E14), where CCAT is the largest one, and E14 is considerably smaller. Table 1 depicts AUC for the various algorithms over a range of values for α. From top to bottom we have graphs for neighbourhoods of size 25, 50, and 100.

The trends that can be seen in these graphs hold for all the other categories not shown here as well. Usually all LLGC variants outperform the support vector machine which was only trained on the labeled examples. The difference becomes more pronounced for the smaller categories, i.e. were the binary learning

Table 2. AUC values for the ECML Challenge submission, pure LLGC, and a linear support vector machine, averaged over three mailboxes; ranks are hypothetical except for the first row

Algorithm	AUC	Rank
primedLLGC(k=100,alpha=0.8,beta=1.0)	0.9491	1/21
support vector machine	0.9056	7/21
pure LLGC(alpha=0.99)	0.6533	19/21

problem is more skewed. Pure LLGC itself is also usually outperformed by the primed version, except sometimes at extreme values of α (0.01 or 0.99). For larger categories the differences between pure and primed LLGC are also more pronounced, and also the influence of α is larger, with best results to be found around the middle of the range. Also, with respect to β, usually best results for $\beta = 0.1$ are found in the upper half of the α range, whereas for the smaller $\beta = 0.01$ best results are usually found at lower values for α. Globally, $\beta = 0.1$ seems to be the slightly better value, which confirms the heuristic presented in the last section, as the ratio between labeled and unlabeled data in this domain is about 0.1.

4.1 Spam Detection

Another very specific text classification problem is the detection of spam email. Recently a competition was held to determine successful learning algorithms for this problem[5]. One of the problems comprised a labeled mailbox with 7500 messages gathered from publically available corpora and spam sources, whereas for prediction three different mailboxes of size 4000 were supplied. Each mailbox had an equal amount of spam and non-spam, but that was not known to the participants in the competition. The three unlabeled prediction mailboxes were very coherent for their non-spam messages, as they were messages of single Enron users. Again, that was not known to the participants. A solution based on a lazy feature selection technique in conjunction with the fast LLGC method described here was able to tie for first place at this competition [24]. In Table 2 we report the respective AUCs for the submitted solution, as well as for a support vector machine, and a pure LLGC approach. This time the support vector machine outperforms the pure LLGC solution, but again the primed LLGC version is the overall winner.

5 Conclusions

In this paper we have extended the well-known LLGC algorithm in three directions: we have extended the range of admissible similarity functions, we have improved the computational complexity by sparsification and we have improved predictive accuracy by priming. An preliminary experimental evaluation using a large text corpus has shown promising results, as has the application to

spam detection. Future work will include a more complete sensitivity analysis of the algorithm, as well as application to non-textual data utilizing a variety of different kernels.

References

1. R.K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. Technical Report RC23462, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA, 2004.
2. M.-F. Balcan and A. Blum. On a theory of learning with similarity functions. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 73–80, New York, NY, USA, 2006. ACM Press.
3. M.-F. Balcan, A. Blum, P.P. Choi, J. Lafferty, B. Pantano, M.R. Rwebangira, and X. Zhu. Person identification in webcam images: an application of semi-supervised learning. In *Proc. of the 22nd International Conference on Machine Learning (ICML 05), Workshop on Learning with Partially Classified Training Data*, pages 1–9, Bonn, Germany, August 2005.
4. A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, New York, NY, USA, 2006. ACM Press.
5. S. Bickel, editor. *Proceedings of the ECML/PKDD 2006 Discovery Challenge Workshop*. Humboldt University Berlin, 2006.
6. A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In C.E. Brodley and A. Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*. Morgan Kaufmann, 2001.
7. A. Blum and T.M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, Madison, Wisconsin, USA, July 1998.
8. M. Breitenbach and G.Z. Grudic. Clustering with local and global consistency. Technical Report CU-CS-973-04, University of Colorado, Department of Computer Science, 2004.
9. O. Chapelle, J. Weston B., and Schölkopf. Cluster kernels for semi-supervised learning. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 585–592. MIT Press, 2002.
10. O. Chapelle and A. Zien. Semi-supervised learning by low density separation. In *Proc. of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 57–64, Barbados, January 2005.
11. O. Delalleau, Y. Bengio, and N.L. Roux. Efficient non-parametric function induction in semi-supervised learning. In *Proceedings of the 10th International Workshop on Artificial Intelligence and statistics (AISTAT 2005)*, 2005.
12. K. Driessens, P. Reutemann, B. Pfahringer, and C. Leschi. Using weighted nearest neighbor to benefit from unlabeled data. In *Proceedings of the Asia-Pacific Conference on Knowledge Discovery in Databases (PAKDD2006)*, 2006.
13. J. Garcke and M. Griebel. Semi-supervised learning with sparse grids. In *Proceedings of the Workshop on Learning with Partially Classified Training Data (ICML2005)*, Bonn, Germany, 2005.

14. T.M. Huang and V. Kecman. Performance comparisons of semi-supervised learning algorithms. In *Proc. of the 22nd International Conference on Machine Learning (ICML 05), Workshop on Learning with Partially Classified Training Data*, pages 45–49, Bonn, Germany, August 2005.
15. T. Joachims. Transductive inference for text classification using support vector machines. In I. Bratko and S. Dzeroski, editors, *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pages 200–209. Morgan Kaufmann, 1999.
16. T. Joachims. Transductive learning via spectral graph partitioning. In T. Fawcett and N. Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 290–297. AAAI Press, 2003.
17. R. Jones. *Learning to extract entities from labeled and unlabeled text.* PhD thesis, Carnegie Mellon University, School of Computer Science, Pittsburgh, Pennsylvania, USA, 2005.
18. R.I. Kondor and J.D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In C. Sammut and A.G. Hoffmann, editors, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML)*, 2002.
19. D. Lewis, Y. Yang, T. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
20. M. Mahdavani, N. de Freitas, B. Fraser, and F. Hamze. Fast computation methods for visually guided robots. In *Proceedings of the The 2005 International Conference on Robotics and Automation (ICRA)*, 2005.
21. A.Y. Ng, M.T. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.
22. K. Nigam, A. McCallum, S. Thrun, and T.M. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2/3), 2000.
23. C.S. Oliveira, F.G. Cozman, and I. Cohen. Splitting the unsupervised and supervised components of semi-supervised learning. In *Proc. of the 22nd International Conference on Machine Learning (ICML 05), Workshop on Learning with Partially Classified Training Data*, pages 67–73, Bonn, Germany, August 2005.
24. B. Pfahringer. A semi-supervised spam mail detector. In Steffen Bickel, editor, *Proceedings of the ECML/PKDD 2006 Discovery Challenge Workshop*, pages 48–53. Humboldt University Berlin, 2006.
25. C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *7th IEEE Workshop on Applications of Computer Vision*, pages 29–36. IEEE Computer Society, 2005.
26. M. Seeger. Learning from labeled and unlabeled data. Technical report, University of Edinburgh, Institute for Adaptive and Neural Computation, 2001.
27. A.J. Smola and R. Kondor. Kernels and regularization on graphs. In B. Schölkopf and M.K. Warmuth, editors, *Computational Learning Theory and Kernel Machines*, Lecture Notes in Computer Science 2777, pages 144–158, 2003.
28. M. Szummer and T. Jaakkola. Partially labeled classification with markov random walks. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 945–952. MIT Press, 2001.
29. V.N. Vapnik. *Statistical learning theory.* J. Wilsley, New York, USA, 1998.
30. A. Vinueza and G.Z. Grudic. Unsupervised outlier detection and semi-supervised learning. Technical Report CU-CS-976-04, University of Colorado, Department of Computer Science, 2004.

31. J. Weston, C. Leslie, E. Le, D. Zhou, A. Elisseff, and W.S. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.
32. D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 189–196, 1995.
33. K. Yu, S. Yu, and V. Tresp. Blockwise supervised inference on large graphs. In *Proc. of the 22nd International Conference on Machine Learning, Workshop on Learning with Partially Classified Training Data*, Bonn, Germany, 2005.
34. D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Yu Thrun, K.S. Lawrence, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.
35. D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proc. of the 22nd International Conference on Machine Learning (ICML 05)*, pages 1041–1048, Bonn, Germany, August 2005.
36. D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In S. Yu Thrun, K.S. Lawrence, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.
37. X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.
38. X. Zhu. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, School of Computer Science, Pittsburgh, Pennsylvania, USA, 2005.
39. X. Zhu, Z. Ghahramani, and J.D. Lafferty. Semi-supervised searning using gaussian fields and harmonic functions. In T. Fawcett and N. Mishra, editors, *Machine Learning, Proceedings of the Twentieth International Conference (ICML)*, 2003.
40. X. Zhu and J. Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of the 22nd International Conference on Machine Leaening (ICML2005)*, 2005.

A Machine Learning Approach to Detecting Instantaneous Cognitive States from fMRI Data

Rafael Ramirez and Montserrat Puiggros

Music Technology Group
Universitat Pompeu Fabra
Ocata 1, 08003 Barcelona, Spain
{rafael,mpuiggros}@iua.upf.es

Abstract. The study of human brain functions has dramatically increased in recent years greatly due to the advent of Functional Magnetic Resonance Imaging. In this paper we apply and compare different machine learning techniques to the problem of classifying the instantaneous cognitive state of a person based on her functional Magnetic Resonance Imaging data. In particular, we present successful case studies of induced classifiers which accurately discriminate between cognitive states produced by listening to different auditory stimuli. The problem investigated in this paper provides a very interesting case study of training classifiers with extremely high dimensional, sparse and noisy data. We present and discuss the results obtained in the case studies.

Keywords: Machine learning, feature extraction, fMRI data.

1 Introduction

The study of human brain functions has dramatically increased in recent years greatly due to the advent of Functional Magnetic Resonance Imaging (fMRI). While fMRI has been used extensively to test hypothesis regarding the location of activation for different brain functions, the problem of automatically classifying cognitive states has been little explored. The study of this problem is important because it can provide a tool for detecting and tracking cognitive processes (i.e. sequences of cognitive states) in order to diagnose difficulties in performing a complex task.

In this paper we describe an approach to detecting the instantaneous cognitive state of a person based on her Functional Magnetic Resonance Imaging data. We present a machine learning approach to the problem of discriminating instantaneous cognitive states produced by different auditory stimuli. We present the results of two case studies in which we have trained classifiers in order to discriminate whether a person is (1) listening to melodic tonal stimuli or listening to nonsense speech, (2) listening to an auditory stimulus or mentally rehearsing the stimulus, (3) listening to melody, speech or rehearsing, (4) listening to a pure tone or a band-passed noise burst, and (5) listening to a low-frequency

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 248–259, 2007.

tone or a high-frequency tone. The problem investigated in this paper is also interesting from the machine learning point of view since it provides an interesting case study of training classifiers with extremely high dimensional (10,000-15,000 features), sparse (32-84 examples) and noisy data.

We apply and compare different machine learning techniques to the task of predicting a subject cognitive state given her observed fMRI data. We associate a class with each of the cognitive states of interest and given a subject's fMRI data observed at time t, the classifier predicts one of the classes. We train the classifiers by providing examples consisting of fMRI observations (restricted to selected brain areas) along with the known cognitive state of the subject. We select the brain areas by applying feature selection methods tuned to the data characteristics.

The rest of the paper is organized as follows: Section 2 sets out the background for this research. In Section 3, we describe our approach to the problem of detecting the instantaneous cognitive state of a person based on her Functional Magnetic Resonance Imaging data. Section 4 presents two case studies. In Section 5 we discuss the results of the case studies, and finally Section 6 presents some conclusions and indicates some areas of future research.

2 Background

Functional Magnetic Resonance Imaging is a brain imaging technique that allows the observation of brain activity in human subjects based on the increase in blood flow to the local vasculature that accompanies neural activity in the brain. It produces time-series data that represents brain activity in a collection of 2D slices of the brain. The collection of the 2D slices form a 3D image of the brain containing in the order of 12000 voxels, i.e. cubes of tissue about 3 millimeters on each side. Images are usually taken every 1-5 seconds. Despite the limitations in temporal resolution, fMRI is arguably the best technique for observing human brain activity that is currently available. Figure 1 shows a fMRI image showing the instantaneous activity of a section of the brain and the activity over time of one of its voxels (white voxels are those with highest activity while dark voxels are those with lowest activity).

Functional Magnetic Resonance Imaging has been widely applied to the task of identifying the regions in the brain which are activated when a human performs a particular cognitive function (e.g. visually recognizing objects). Most of the reported research summarizes average fMRI responses when a human is presented with a particular stimulus repeatedly. Regions in the brain activated by a particular task are identified by comparing fMRI activity during the period where the stimulus is presented with the activity detected under a control condition. Other research describes the effects of varying stimuli on activity, or correlations among activity in different brain regions. In all these cases, the results are statistics of effects averaged over multiple trials and multiple subjects.

Haxby et al [4] detect different patterns of fMRI activity generated when a human views a photograph of different objects (e.g. faces, houses). Although

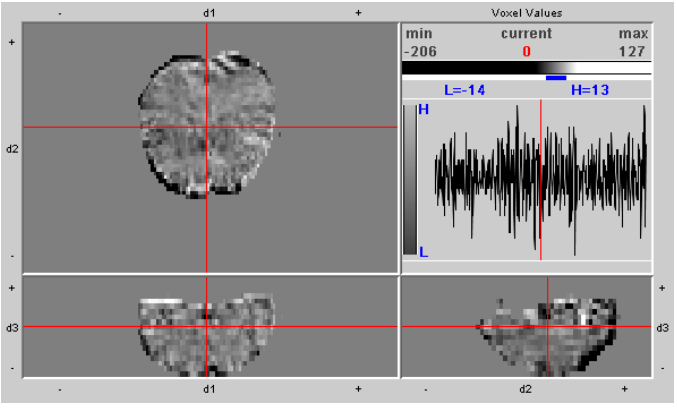


Fig. 1. fMRI image showing the instantaneous activity of a section of the brain and the activity over time of one of its voxels: white voxels are those with highest activity while dark voxels are those with lowest activity

this information was not specifically used for classifying subsequent fMRI data, Haxby et al reported that they could automatically identify the data samples related to the same object category. Wagner et al [9] reported that they have been able to predict whether a verbal experience is to be remembered later based on the amount of activity in particular brain regions during the experience. Closer to the work reported in this paper is the work by Mitchell et al [6,7] who have applied machine learning methods to the same problem investigated here. In particular, they have trained classifiers to distinguish whether a subject is looking at a picture or a sentence, reading an ambiguous or non-ambiguous sentence, and the type of word (e.g. a word describing food, people, etc.) to which a subject is exposed. Cox et al [2] applied support vector machine to fMRI data in order to classify patterns of fMRI activation produced by presenting photographs of various categories of objects.

3 Classifying Cognitive States

In this section we present our approach to training and evaluating classifiers for the task of detecting the instantaneous cognitive state of a person. Given a person's observed instantaneous fMRI data at time t, we train a classifier in order to predict the cognitive state that gave rise to the observed data. The training data is a set of examples of fMRI observations along with the known cognitive state.

3.1 Learning Algorithms

In this paper we explore different machine learning techniques to induce a classifier of the following form.

$$Classifier(fMRIdata(t)) \rightarrow CognState$$

where $fMRIdata(t)$ is an instantaneous fMRI image at time t and $CognState$ is a set of cognitive states to be discriminated. For each subject in the fMRI data sets we trained a separate classifier. We explored a number of classifier induction methods, including:

- *Decision Trees.* A decision tree classifier recursively constructs a tree by selecting at each node the most relevant attribute. This process gradually splits up the training set into subsets until all instances at a node have the same classification. The selection of the most relevant attribute at each node is based on the *information gain* associated with each node of the tree (and corresponding set of instances). We have explored the decision tree building algorithm C4.5 [8] without pruning and with postpruning (using *subtree raising*).
- *Support Vector Machines (SVM).* SVM [3] take great advantage of using a non linear attribute mapping that allows them to be able to predict non-linear models (thought they remain linear in a higher dimension space). Thus, they provide a flexible prediction, but with a higher computational cost necessary to perform all the computations in the higher dimensional space. SVM have been applied successfully in applications involving high dimensional data. Their classification accuracy largely depends on the choice of the kernel evaluation function and the parameters which control the amount to which deviations are tolerated (denoted by epsilon). In this paper we have explored SVM with linear and polynomial kernels (2nd, 3rd and 4th order) and we have set epsilon to 0.05.
- *Artificial Neural Networks (ANN).* ANN learning methods provide a robust approach to approximating a target function. In this paper we apply a gradient descent back propagation algorithm [1] to tune the neural network parameters to best fit the fMRI training set. The back propagation algorithm learns the weights for a multi layer network, given a network with a fixed set of units and interconnections. We set the momentum applied to the weights during updating to 0.2 and the learning rate (the amount the weights are updated) to 0.3. We use a fully-connected multi layer neural network with one hidden layer (one input neuron for each attribute and one output neuron for each class).
- *Lazy Methods.* Lazy Methods are based on the notion of lazy learning which subsumes a family of algorithms that store the complete set of given (classified) examples of an underlying example language and delay all further calculations until requests for classifying yet unseen instances are received. In this paper we have explored the k-Nearest Neighbor (k-NN) algorithm (with $k \in \{1, 2, 3, 4, 7\}$) which in the past has been used successfully in other applications involving high dimensional data, and is capable of handling noisy data well if the training set has an acceptable size. However, k-NN does not behave well in the presence of irrelevant attributes.
- *Ensemble Methods.* One obvious approach to making more reliable decisions is to combine the output of several different models. In this paper we explore

the use of methods for combining models (called *ensemble* methods) generated by machine learning. In particular, we have explored *voting, stacking, bagging* and *boosting*. In many cases they have proved to increase predictive performance over a single model. For voting and stacking we considered decision trees, SVM, ANN, and 1-NN (for stacking the decision trees algorithm was used as 'meta-learner'. For bagging and boosting we applied decision trees.

3.2 Feature Selection

As the classification task reported in this paper clearly involves a high dimensional training data, it is necessary to apply feature selection methods before training the classifiers. In this paper, we consider two feature selection strategies:

- select features according to how well they discriminate the classes of interest.
- select features according to how well they discriminate each class of interest from *fixation*, i.e. the periods during which the subjects are typically not performing any task but instead are staring at a fixation point.

The feature selection strategies are motivated by the fact that fMRI binary classification problems naturally give rise to three types of data: data corresponding to the two target classes, C_1 and C_2, and data corresponding to the fixation condition. Data corresponding to C_1 and C_2 is composed of signal plus noise, while data corresponding to the fixation condition contains only noise, i.e. it contains no relevant signal. Thus, two natural feature selection methods are *voxel discriminability*, i.e. how well the feature discriminates C_1 and C_2, and *voxel activity*, i.e. how well the feature distinguishes C_1 or C_2 from the fixation class [6]. While the former selection method is a straightforward method for selecting voxels which discriminate the two classes, the later focuses on choosing voxels with large signal-to-noise ratios, although it ignores whether the feature actually discriminates the two classes. Within the fMRI community it is common to use voxel activity to select a subset of relevant voxels. In more detail, the voxel discriminability and voxel activity feature selection methods are as follows:

- *voxel discriminability.* For each voxel and target class, a t-test is performed comparing the fMRI activity of the voxel in examples belonging to the two stimuli of interest. In the case of three-class classification tasks, instead of the t-test, an f-test is performed comparing the fMRI of the voxel in examples belonging to the different stimuli of interest. n voxels are then selected by choosing the ones with larger t-values.
- *voxel activity.* For each voxel and target class, a t-test is performed comparing the fMRI activity of the voxel in examples belonging to a particular stimulus to its activity in examples belonging to fixation periods. For each class, n voxels are then selected by choosing the ones with larger t-values. Note that these voxels may discriminate only one target class from fixation.

3.3 Classifier Evaluation

We evaluated each induced classifier by performing the standard 10-fold cross validation in which 10% of the training set is held out in turn as test data while the remaining 90% is used as training data. When performing the 10-fold cross validation, we leave out the same number of examples per class. In the data sets, the number of examples is the same for each class considered, thus by leaving out the same number of examples per class we maintain a balanced training set. In order to avoid optimistic estimates of the classifier performance, we explicitly remove from the training set all images occurring within 6 seconds of the hold out test image. This is motivated by the fact that the fMRI response time is blurred out over several seconds.

4 Case Studies

In this section we describe two case studies on training classifiers using fMRI data. We summarize the results we have obtained and postpone the discussion of the results to the next section.

4.1 Melody, Speech and Rehearsal Study

In this fMRI study [5] a total of 6 right handed subjects with a mean age of 27 (3 men and 3 women) participated. Twenty-one short unfamiliar piano melodies, each of 3 seconds duration, were recorded using a MIDI synthesizer. Melodies consisted of 5 to 17 notes in the C key (i.e. white keys on the piano). Note durations ranged from 106 to 1067 msec and patterns were chosen to be rhythmical according to Western music. An equal number of nonsense sentences (nouns and verbs were replaced by pseudo words), also approximately 3 seconds long, were recorded and digitalized. Each trial in this experiment consisted of an initial stimulus presentation (music or sentence; 3 seconds duration), followed by a 15 second rehearsal period, a representation of the stimulus (signaling the subject to stop rehearsing), and ended with 15 seconds of rest. The next trial was then initiated by a new stimulus item. Thus, each full trial extended over a period of 36 seconds. fMRI images were acquired every 2 seconds. The voxel dimensions were set to 6 x 4 x 4 mm.

We used this data to investigate the feasibility of training successful classifiers to detect whether a subject is listening to melodic tonal stimuli, listening to nonsense speech, mentally rehearsing a melody or simply resting (control condition). In particular, we trained classifiers for the tasks of distinguishing among the cognitive states for (1) listening to a melody versus listening to nonsense speech, (2) listening to an auditory stimulus (a melody or nonsense speech) versus mentally rehearsing the stimulus, and (3) listening to a melody versus listening to nonsense speech versus mentally rehearsing the auditory stimulus (i.e. a three class classifier). This is, given the general classifier of the form $Cl(fMRIdata(t)) \rightarrow CS$, we are interested in the set CS of cognitive states to be discriminated to be

$\{melody, speech\}$, $\{audition, rehearsal\}$, and $\{melody, speech, rehearsal\}$, respectively for (1), (2) and (3) as above.

Initially, we filter the fMRI data by eliminating voxels outside the brain. This is done by discarding the voxels below an activation threshold. The average number of voxels per subject after filtering was approximately 12,000 (although this varied significantly from subject to subject). Once the fMRI data is sifted, we proceed to select voxels based on both the voxel discriminability and voxel activity feature selection methods described in the previous section. We average the activity values of contiguous voxels representing brain regions. We restricted the amount of features for training the classifiers to 10-22.

There were a total of 84 examples available for each subject for the two-class classification tasks (i.e. 42 examples for each class), and 126 examples for the three-class classification task (i.e. 42 examples for each class). For the two-class classification tasks, the expected classification accuracy of the default classifier (one which chooses the most common class) is 50% (measured in correctly classified instances percentage), while for the three-class classification task, the expected accuracy is 33%. For the melody-versus-speech, audition-versus-rehearsal, and melody-versus-speech-versus-rehearsal classifiers the average accuracies obtained for the most successful trained classifier using the most successful feature selection strategy were 97.19%, 84.83%, and 69.44%, respectively. For these classifiers the best subject's accuracies were 100%, 98.57%, and 81.67%, respectively. The results are statistically significant which indicates that it is indeed feasible to train successful classifiers to distinguish these cognitive states. The correctly classified instances percentage for each subject and each learning method is presented in Tables 1-4.

Table 1. Classifiers accuracies for listening to a melody versus listening to nonsense speech (using voxel discriminability feature selection)

Subject	1	2	3	4	5	6
DTrees	76.50	96.33	86.33	77.50	79.00	100.00
SVM	95.50	98.00	91.00	88.50	95.50	100.00
ANN	88.00	98.00	91.33	70.00	98.00	100.00
k-NN	90.50	96.33	93.00	91.00	97.50	100.00
Bagging	76.50	96.33	88.00	88.50	90.50	100.00
Boosting	84.00	96.33	95.00	90.50	93.00	100.00
Voting	91.00	96.33	94.67	82.00	97.50	100.00
Stacking	95.50	96.33	86.00	86.00	100.00	100.00

4.2 Pure Tones and Band-Passed Noise

In this fMRI study [10] twelve subjects with normal hearing listened passively to one of six different stimulus sets. These sets consisted of either pure tones (PTs) with a frequency of 0.5, 2 or 8 kHz, or band-passed noise (BPN) bursts with the same logarithmically spaced center frequencies and a bandwidth of one octave (i.e. from 0.35-0.7, 1.4-2.8, and 5.6-11.2 kHz, respectively). All stimuli were 500

Table 2. Classifiers accuracies for listening to auditory stimulus (melody or nonsense speech) versus mentally rehearsing the stimulus (using voxel discriminability feature selection)

Subject	1	2	3	4	5	6
DTree	82.50	67.33	79.82	70.50	76.50	81.00
SVM	97.50	78.33	81.07	88.00	79.00	83.50
ANN	95.00	74.67	80.00	84.00	64.50	70.00
k-NN	95.00	77.33	79.64	79.50	65.50	72.50
Bagging	80.00	74.67	79.46	74.50	77.00	72.00
Boosting	87.50	67.00	79.82	74.50	81.00	79.00
Voting	92.50	76.33	79.82	77.00	74.50	72.00
Stacking	85.00	72.67	74.46	78.50	66.50	81.50

Table 3. Classifiers accuracies for listening to a melody versus listening to nonsense speech versus mentally rehearsing the auditory stimulus (using voxel discriminability feature selection)

Subject	1	2	3	4	5	6
DTrees	42.81	72.50	73.33	53.81	59.33	66.67
SVM	50.00	68.00	73.33	57.38	42.33	73.33
ANN	45.00	77.00	78.33	60.24	54.67	69.76
kNN	43.57	77.00	80.00	54.05	55.33	57.14
Bagging	41.86	74.50	78.33	57.14	54.00	58.57
Boosting	41.38	70.50	71.67	46.90	57.00	61.67
Voting	44.76	77.00	81.67	58.81	45.67	57.14
Stacking	40.67	61.00	80.00	44.29	63.00	77.86

msec in duration including 50 msec rise/fall times to minimize on/offset artifacts. Stimuli were presented at a rate of 1Hz during the "stimulus-on" intervals of the functional scans. The subjects underwent 12 functional runs consisting of four 32 sec cycles divided into two 16-sec "'stimulus-on" and "stimulus-off" epochs. During six runs each PT and BPN bursts were the on-stimuli. The voxel size was 3.75 x 3.75 x 4.4 mm^3.

We used this data to train classifiers to detect whether a subject is listening to a high or low frequency tone, and whether the subject is listening to a pure tone or a band-passed noise burst. In particular, we trained classifiers for the tasks of distinguishing among the cognitive states for (1) listening to a high frequency PT versus listening to a low frequency PT, (2) listening to a PT versus listening to a BPN burst (both in a middle frequency). Given the general classifier of the form $Cl(fMRIdata(t)) \rightarrow CS$, we are interested in the set CS of cognitive states to be discriminated to be $\{PTHigh, PTLow\}$, and $\{PTMiddle, BPNMiddle\}$.

We select voxels in the same manner as in the previous case study. This is, we initially filter the fMRI data by eliminating voxels outside the brain by discarding the voxels below an activation threshold. The average number of voxels per subject after filtering was approximately 14,000 (it varied significantly from subject to subject). Once the fMRI data is sifted, we proceed to select voxels

Table 4. Classifiers accuracies for listening to a melody versus listening to nonsense speech versus mentally rehearsing the auditory stimulus (using voxel activity feature selection)

Subject	1	2	3	4	5	6
DTrees	45.95	64.31	82.14	50.48	77.00	72.38
SVM	66.19	67.08	82.14	57.38	69.67	77.62
ANN	48.81	61.81	79.11	50.48	63.00	72.38
k-NN	64.29	69.03	81.96	57.14	72.33	73.10
Bagging	51.19	71.67	73.04	52.38	82.67	70.95
Boosting	46.67	69.17	80.71	49.05	75.00	67.86
Voting	55.24	68.89	84.29	50.48	69.33	72.38
Stacking	54.76	59.86	76.96	55.71	74.33	65.24

Table 5. Listening to a high frequency PT versus listening to a low frequency PT (using voxel discriminability feature selection)

Subject	1	2	3	4	5	6
DTrees	100.0	90.00	96.67	97.50	100.0	80.83
SVM	100.0	100.0	100.0	100.0	100.0	100.0
ANN	100.0	100.0	100.0	100.0	100.0	100.0
k-NN	100.0	100.0	100.0	100.0	100.0	100.0
Bagging	96.67	96.67	96.67	97.50	96.67	90.00
Boosting	100.0	90.00	96.67	97.50	100.0	80.83
Voting	100.0	100.0	100.0	100.0	100.0	100.0
Stacking	100.0	96.67	100.0	100.0	100.0	100.0

based on both the voxel discriminability and voxel activity feature selection methods described before. We average the activity values of contiguous voxels representing brain regions. We restricted the amount of features for training the classifiers to 12-25.

There were a total of 32 training examples available for each subject (i.e. 16 examples for each class). The expected correctly classified instances percentage of the default classifier (selecting the most common class) is 50%. For the both PT-High versus PT-Low, and the PT versus BPN classifiers we obtained average accuracies of 100% for the SVM, k-NN and voting. These results are clearly statistically significant and indicate that it is feasible to train successful classifiers to distinguish these cognitive states. The correctly classified instances percentage for each subject and each learning method is presented in Table 5 and Table 6. Similar results were obtained using voxel activity feature selection (we omit the presentation of the corresponding tables due to space limitations).

5 Discussion

The difference between the results obtained and the accuracy of a baseline classifier, i.e. a classifier guessing at random (50% and 33% in the case of the two-class and three-class classification task, respectively) indicates that the fMRI

Table 6. Listening to a middle frequency PT versus listening to a middle frequency BPN burst (using voxel discriminability feature selection)

Data Set	(1)	(2)	(3)	(4)	(5)	(6)
DTrees	94.17	83.37	96.67	74.17	96.67	91.67
SVM	100.0	100.0	100.0	100.0	100.0	100.0
ANN	100.0	100.0	100.0	97.50	100.0	100.0
k-NN	100.0	100.0	100.0	100.0	100.0	100.0
Bagging	97.50	93.33	96.67	80.83	100.0	91.67
Boosting	94.17	86.67	96.67	87.50	96.67	91.67
Voting	100.0	100.0	100.0	100.0	100.0	100.0
Stacking	100.0	100.0	100.0	100.0	100.0	100.0

data contains sufficient information to distinguish these cognitive states, and machine learning methods are capable of learning the fMRI patterns that distinguish these states. It is worth noting that every learning algorithm investigated (decision trees, SVM, ANN, k-NN and the reported ensemble methods) produced significantly better than random classification accuracies for every study. This supports our statement about the feasibility of training classifiers for the case studies reported. However, note that this does not necessary imply that it is feasible to train classifiers for arbitrary tasks.

The results also indicate that certain tasks seem to be more difficult to discriminate than others. For example, the average accuracy of the melody-versus-speech classifiers is consistently higher than that of the audition-versus-rehearsal classifiers. This may seem to indicate, as previously suggested, that there is more commonality in the underlying brain processes of audition and rehearsal, than in the underlying processes of listening to different types of auditory stimuli. Currently, the general question of exactly which cognitive states can be reliably discriminated remains an open question.

The accuracy of the classifiers for different subjects varies significantly, even within the same study and using the same learning method. Subjects producing high accuracies with one learning method tend to produce high accuracies with the other learning methods. These uneven accuracies among subjects may be due to the data being corrupted (e.g. by head motion during scanning). In any case, it has been reported that there exists considerable variation in fMRI responses among different subjects.

We have selected the number of features n, i.e. regions raging from 1 to 12 voxels, empirically. We incrementally considered different values for n and stop when average classifier accuracy stops improving. The number of features used ranged from 10 to 15 and from 18-22 for the two-class classification and three-class classification tasks, respectively.

It is worth mentioning that in all the experiments performed we provided no information about relevant brain regions involved in the tasks performed by the subjects. This contrasts with other approaches (e.g. [6,2]) where the input to the classifiers is the set of voxels in the regions of interests (ROIs) selected for each particular study. Here, we have treated equally *all* voxels in the fMRI studies

regardless of which brain region they belong to. Incorporation information about the ROI for each fMRI study would very likely improve the accuracies of the classifiers. We decided not to provide any ROIs information in order to eliminate any feature selection bias.

We expected the k-NN classifier to underperform the other classifiers given the high dimensional and sparse training sets (k-NN is known to be very sensitive to irrelevant features). However, the results show no clear trend in this respect. This led us to think that our feature selection process indeed eliminated irrelevant features.

Mitchell et al [6] reported that voxel activity feature selection widely outperformed voxel discriminability feature selection in several fMRI studies. This result may be explained by the high dimensionality, noisy and sparseness characteristics of the fMRI data. Typically, in fMRI data only a few voxels contain a signal related to the stimulus under study and given the noisy characteristics of the data it is expected to select irrelevant voxels which appear good discriminators. Thus, choosing voxels with high signal-to-noise ratios (as it is the case with voxel activity feature selection) would eliminate a considerable number of irrelevant voxels. However, in the two case studies reported in this paper there is no clear accuracy difference between the classifiers trained with voxels selected by voxel activity feature selection and those trained using voxel discriminability feature selection. This may be due to the fact that, in contrast with the results reported by Mitchell et al, we have trained our classifiers with a substantially smaller number of voxels (approximately 40 compared to 800). While voxel discriminability feature selection may select irrelevant voxels, voxel activity feature selection may choose high signal to noise voxels that cannot discriminate the target classes. It may be the case that by choosing a smaller number of features the irrelevant voxels selected by voxel discriminability feature selection are removed, which in turn minimizes the difference of the results for the two feature selection methods. Also, the fact that classification was done using only a small set of features, classification could, in principle, be used to extract information about a subjects cognitive state on a near real-time basis.

6 Conclusion

In this paper we have explored and compared different machine learning techniques for the problem of classifying the instantaneous cognitive state of a person based on her functional Magnetic Resonance Imaging data. The problem provides a very interesting instance of training classifiers with extremely high dimensional, sparse and noisy data. We presented successful case studies of induced classifiers which accurately discriminate between cognitive states involving different types of auditory stimuli. Our results seem to indicate that fMRI data contains sufficient information to distinguish these cognitive states, and that machine learning techniques are capable of learning the fMRI patterns that distinguish these states. Furthermore, we proved that it is possible to train successful classifiers using only a small number of features (i.e. 12-15 voxels)

extracted from the studied fMRI data, and with no prior anatomical knowledge. This contrasts previous approches in which a large number of voxels is considered (e.g. 800 voxels) and which consider regions of interest in the brain in order to simplify the problem We considered two feature selection strategies: voxel discriminability feature selection and voxel activity feature selection. Contrary to previous results, we found no clear accuracy difference between the classifiers trained with voxels selected by voxel activity feature selection and those trained using voxel discriminability feature selection, in the two case studies described. This result deserves further investigation. As future work, we are particularly interested in exploring rule-based machine learning techniques (e.g. Inductive logic programming) in order to explain the predictions of the classifiers and to incorporate domain knowledge into the learning process.

Acknowledgments. This work was supported by the Spanish Ministry of Education and Science under Grant TIN2006-14932-C02-01 (ProSeMus Project). We would like to sincerely thank the fMRI Data Center for providing the fMRI data.

References

1. Chauvin, Y. et al. (1995). Backpropagation: Theory, Architectures and Applications. Lawrence Erlbaum Assoc.
2. Cox, D. D., Savoy, R. L. (2003). Functional magnetic resonance imaging (fMRI) "brain reading": Detecting and classifying distributed patterns of fMRI activity in human visual cortex. NeuroImage, 19, 261-270.
3. Cristianini N., Shawe-Taylor J. (2000). An Introduction to Support Vector Machines, Cambridge University Press
4. Haxby, J. et al. (2001) Distributed and Overlapping Representations of Faces and Objects in Ventral Temporal Cortex. Science 2001, 293:2425-2430.
5. Hickok, G., Buchsbaum, B., Humphries, C., Muftuler, T. (2003). Auditory-Motor Interaction Revealed by fMRI: Speech, Music and Working Memory in Area Spt. Journal of Cognitive Neuroscience, Vol. 15, Issue 5. July 2003
6. Mitchell, T.M., Hutchinson, R., Niculescu, R.S., Pereira, F., Wang, X., Just, M. and Newman S. (2004). Learning to Decode Cognitive States from Brain Images, Machine Learning, Vol. 57, Issue 1-2, pp. 145-175.
7. Mitchell, T., Hutchinson, R., Just, M., Niculescu, R.N., Pereira, F., Wang, X. (2003). Classifying Instantaneous Cognitive States from fMRI Data, American Medical Informatics Association Symposium, October 2003.
8. Quinlan, J.R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106.
9. Wang, X., et al. (2003) Training fMRI Classifiers to Detect Cognitive States across Multiple Human Subjects. Neural Information Processing Systems.
10. Wessinger, C.M., VanMeter, J., Tian, B., Van Lare, J., Pekar, J., Rauschecker, J.P. (2001). Hierarchical organization of the human auditory cortex revealed by functional magnetic resonance imaging. J Cogn Neurosci. 2001 Jan 1;13(1):1-7.

Discovering Correlated Items in Data Streams

Xingzhi Sun, Ming Chang, Xue Li, and Maria E. Orlowska

School of Information Technology and Electrical Engineering
The University of Queensland, Brisbane, Australia
{sun, mingc, xueli, maria}@itee.uq.edu.au

Abstract. Recently, the problem of finding frequent items in a data stream has been well studied. However, for some applications, such as HTTP log analysis, there is a need to analyze the correlations amongst frequent items in data streams. In this paper, we investigate the problem of finding correlated items based on the concept of unexpectedness. That is, two items x and y are correlated if both items are frequent and their actual number of co-occurrences in the data stream is significantly different from the expected value, which can be computed by the frequencies of x and y. Based on the *Space-Saving* algorithm [1], we propose a new one-pass algorithm, namely *Stream-Correlation*, to discover correlated item pairs. The key part of our algorithm is to efficiently estimate the frequency of co-occurrences of items with small memory space. The possible error can be tightly bounded by controlling the memory space. Experiment results show the effectiveness and the efficiency of the algorithm.

1 Introduction

A data stream [2,3] is a sequence of items that arrive at a rapid rate. Nowadays, many applications require to process high volume of data streams, such as telecom call records, network traffic measurements, web click-streams, and time-stamped data from sensor networks.

Finding frequent items in data streams is regarded as one of the important research problems in streaming data management. While the task is simply to find the items with the frequency above a specified threshold, it is very useful in many applications. For example, in network traffic monitoring, it is of great importance to track IP addresses that generate the considerable amount of traffic in the network. The challenge of this research is that the total number of items could be so large (considering the number of valid IP addresses) that it is impossible to keep exact information for each item. Many approaches [4,5,6,7,8,9,10,1] have been proposed to use a fixed small amount of memory for dynamically maintaining the information of items, such that the frequent items can still be identified but only with bounded error on their frequencies.

While all the above mentioned studies focus on finding frequent single items, there is also a need to analyze the dependency among those frequent items in a data stream. A motivating application of such analysis is to detect "Undetectable Hit Inflation (UHI)", which is recently described in [11]. UHI is a

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 260–271, 2007.

type of click fraud in Internet advertising. Briefly, in the click-through payment program ("pay-per-click"), for claiming more revenue, two dishonest web sites collaborate together to inflate the clicks to the advertising web site. Due to the space limit, please refer to [12,11] for details of UHI.

In [11], the above-mentioned fraud activity can be detected by mining the strong dependency between two dishonest web sites in a stream of HTTP requests, which is available from the Internet Service Provider (ISP). The task is to find any two web sites x and y such that x is frequently requested in the data stream and a large proportion of x requests are followed by y requests in a specified time interval T. Such a problem is modelled as: finding any association $(x \xrightarrow{T} y)$ between two items x and y from the data stream by one scan.

In [11], the support and confidence model (with slight difference) is adopted for discovering the dependency of two web sites (items). However, it is known that in some cases, the support and confidence may be misleading as interestingness measures. For example, let us consider two independent web sites W_1 and W_2, both of which are frequently requested. Suppose that the requests of the web site W_2 are so frequent that even without the dependency with W_1, it can still be expected that within some time interval T, a request of W_1 is likely followed by a request of W_2. In this case, based on the interestingness measure defined in [11], W_1 and W_2 will be regarded as dishonest web sites by mistake. In general, the frequencies of different items can vary significantly, failing to consider the intrinsic frequencies of these items may provide the misleading result in the dependency analysis.

To rectify the identified shortcoming, we evaluate the dependency of items in terms of "unexpectedness", i.e., a pattern/rule is interesting if it is unexpected to prior knowledge. Particularly, we introduce a more general problem, discovering correlated items in data streams, as stated below: *Given a data stream and a time interval T, two items x and y are correlated if 1) both items are frequent, and 2) their actual number of co-occurrences (related to the interval T) in the data stream is significantly greater than the expected value, which is derived based on the assumption that occurrences of x and the occurrences of y are independent.*

The first condition regulates that the potentially interesting items should be of statistical significance. The second condition specifies the correlation based on the unexpectedness. Because we compute the expected number of co-occurrences of two items based on the assumption that these two items are independent, the level of "unexpectedness" can reflect their correlation. Our task is to discover all correlated item pairs with possible bounded errors by scanning the data stream only once. The discovered item pair (x, y) can be interpreted as: x and y are likely to occur together. Note that in this problem, we do not consider the order between item x and y. However, the discussions can be extend to the ordered case readily.

Since finding frequent items has been well studied, our research focuses on 1) computing the expected number of co-occurrences for two given items x and y, and more importantly, 2) finding the actual number of co-occurrences with a bounded error. In this paper, we define the co-occurrence of x and y based

on the concept of *minimal occurrence (MO)* [13]. Given the interval T, any co-occurrence of two items x and y in a stream refers to a MO of the un-ordered item pair (x, y) with constraint T.

When computing the expected number of co-occurrences of two items x and y, we assume that the occurrences of x and the occurrences of y are independent. In addition, we assume that *for any type of item, its occurrences in data stream follow the Poisson process*. This assumption is realistic in various applications. For example, it is well known that the number of requests on an IP address follows the Poisson distribution. So, in HTTP log analysis, the occurrences of a given item (i.e., the requests on an IP address) can be well modelled by a Poisson process. Based on this assumption and the definition of MO, we derive the formula for computing the expected number of co-occurrences of two items x and y, which is a function of the frequencies of x and y, and the parameter T.

Considering the task of finding the actual number of co-occurrences, obviously, it is not possible to maintain a counter for every item pairs. We propose a data structure, *Correlation-Summary*, to dynamically organize a bounded number of counters for item pair. On top of the *Space-Saving* algorithm [1], we develop a one-pass algorithm, *Stream-Correlation,* to approximately count the number of co-occurrences for all potentially frequent item pairs. The proposed algorithm can efficiently find the results with bounded error by using limited memory space.

The rest of this paper is organized as follows. Section 2 gives the related work. In Section 3, we formulate the problem of discovering correlated item pairs. In Section 4, our algorithm is presented followed by the theoretical analysis. Section 5 shows the experiment results. Finally, we conclude the paper in Section 6.

2 Related Work

The problem of finding frequent items in data streams can be described as: Given a data stream S of length N, a frequency parameter $\alpha \in (0, 1)$, an error parameter $\epsilon \ll \alpha$, and a probabilistic parameter ρ, at any time, with a small bounded memory, find the items with their estimate frequency such that 1) all items with true frequency greater than $(\alpha - \epsilon)N$ are output, and 2) the estimated frequency is higher than the true frequency by at most ϵN with high probability ρ. Substantial work [4,5,6,7,8,9,10,1] has been done to handle the problem of finding frequent items and its variations, e.g., top-k frequent items. The proposed techniques can be classified as counter-based approaches [10,6,9,1] and sketch-based (or hash-based) approaches [8,7,5,4]. Please refer to [1] for details.

In our problem, we apply the *Space-Saving*, a counter-based algorithm proposed in [1], to find the frequent items. In the algorithm, a fixed number of counters are dynamically allocated to items. When an item is observed, if there exists a counter for this item, its counter is increased; otherwise, counters will be reallocated based on certain techniques. The *Space-Saving* algorithm only uses $\lceil \frac{1}{\epsilon} \rceil$ counters to guarantee the error bound ϵN (i.e., $\rho = 1$). We will give the algorithm description in later section.

The most relevant research to our work is [11], which discusses the associations between two items in the data stream. A rule $(x \xrightarrow{T} y)$ is interesting if x is frequent (w.r.t. support) and a large proportion (w.r.t. confidence) of x occurrences are followed by y occurrences in the specified time interval T. However, in our work, we deal with the problem of correlation analysis. Instead of using a support and confidence model, we define the interestingness measures based on the unexpectedness. Due to this key difference, we need to compute the expected number of co-occurrences based on the probability theory. Also, when counting the number of co-occurrences of items, we propose different data structure and algorithm procedures.

In [14], correlation is discussed in the transaction database. Given two items x and y, let $P(x), P(y)$, and $P(x, y)$ be the support of $\{x\}, \{y\}$, and $\{x, y\}$ in the transaction database respectively. The occurrences of x and the occurrences of y are defined as independent if $P(x, y) = P(x)P(y)$. Otherwise, the occurrences of x and y are dependent and their correlation can be measured by $\frac{P(x,y)}{P(x)P(y)}$. In a data stream, it is hard to define "transactions". So, we need to use the unexpectedness as our interestingness measure.

3 Problem Statement

Let us consider a finite set E of *items*. An *event* is a pair (a, t), where $a \in E$ and t is the *timestamp* of the event. A *stream* S of *length* N is a sequence of events $(a_1, t_1), \ldots, (a_I, t_I), \ldots, (a_N, t_N)$, which are totally ordered by their timestamps. Given a data stream S, its *duration* $Dur(S)$ is the time span of S, namely, $Dur(S) = t_N - t_1$. Let a *window* w be $[t_s, t_e]$, where t_s and t_e are the start time and the end time of w respectively. The *window size* of w is $(t_e - t_s)$.

Given a stream S, the *frequency* of an item x, denoted as $F(x)$, is the number of occurrences of x in S. For two items x and y, we define the *co-occurrence* of the item pair (x, y) based on the concept of *minimal occurrence* (MO). Note that we do not consider the order between x and y, i.e., $(x, y) = (y, x)$.

In [13], Mannila *et al.* have proposed the concept of *minimal occurrence* (MO) to represent an episode that occurs in a sequence. According to their definition, a window w is a *minimal occurrence* of an item pair (x, y) *iff* 1) the window contains (x, y), and 2) there does not exist a subwindow $w' \subset w$ (i.e., $t_s \leq t'_s$, $t_e \geq t'_e$, and $Size(w) \neq Size(w')$) such that w' contains (x, y). To our problem, we add the condition 3): $Size(w) \leq T$, where T is a predefined window size.

Here we give an example of *minimal occurrence* of an item pair in a stream. A stream S is visualized in Figure 1. Suppose that T is 3. For a non-ordered item pair (b, c), the MOs of (b, c) in S are $[1, 2], [8, 10]$, and $[10, 12]$.

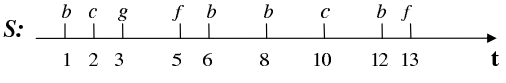


Fig. 1. An example of MOs

Definition 1 (co-occurrence frequency). *Given a stream S and a window size T, we define the co-occurrence frequency of an item pair (x, y) as the number of minimal occurrences of (x, y) with constraint T, and we denote it as $F(x, y)$.*

Definition 2 (expected co-occurrence frequency). *Given a stream S, a window size T, and two items x and y with the frequency $F(x)$ and $F(y)$ respectively, let us assume that 1) x and y occur independently in S and 2) both occurrences of x and occurrences of y follow the Poisson process. The expected co-occurrence frequency of x and y, $E(x, y)$, is defined as the expected number of MOs of (x, y) with constraint T, which is computed under the above two assumptions.*

Theorem 1. *Let the frequencies of two items x and y in a stream S be $F(x)$ and $F(y)$ respectively. Given the window size T, the expected co-occurrence frequency $E(x, y)$ can be computed as: $E(x, y) = \frac{2F(x)F(y)}{F(x)+F(y)}(1 - e^{-\frac{(F(x)+F(y))T}{Dur(S)}})$.*

Proof. When the occurrences of an item x follow the Poisson process, the formula $P_x(k, t) = \frac{(\lambda_x t)^k}{k!}e^{-\lambda_x t}$ gives the probability that x occurs exactly k times in a given time interval t. In addition, the density function for the interval t between two successive x occurrences is $f_x(t) = \lambda_x e^{-\lambda_x t}$. In the above formulas, λ_x is the expected number of x occurrences per time unit. To our problem, we have $\lambda_x = \frac{F(x)}{Dur(S)}$, where $Dur(S)$ is the duration of S.

Let us first discuss a single occurrence of x at t_0 in the stream. Now we consider the following four situations. A_1 : the next x occurs after $t_0 + T$; A_2 : y occurs in the interval $(t_0, t_0 + T)$ at least once; A_3 : the next x occurs at $t_0 + t$ where $0 < t \leq T$; A_4 : y occurs in the interval $(t_0, t_0 + t)$ at least once.

Because we assume that the occurrences of x and the occurrences of y are independent, according to the definition of MO, the probability that the event (x, t_0) can contribute to an MO of (x, y) with the later occurrence of y is $E_\Delta = P(A_1)P(A_2) + \int_0^T P(A_3)P(A_4)dt$.

Since the occurrences of both x and y follow the Poisson process, we know that $P(A_1) = P_x(0, T) = e^{-\lambda_x T}$, $P(A_2) = 1 - P_y(0, T) = 1 - e^{-\lambda_y T}$, $P(A_3) = f_x(t) = \lambda_x e^{-\lambda_x t}$, and $P(A_4) = 1 - P_y(0, t) = 1 - e^{-\lambda_y t}$. Therefore, we can compute that $E_\Delta = \frac{F(y)}{F(x)+F(y)}(1 - e^{-\frac{(F(x)+F(y))T}{Dur(S)}})$.

Considering the occurrences of y before t_0 as well, the expected number of MOs of (x, y) contributed by a single occurrence of x is $2E_\Delta$. Also, because there are $F(x)$ occurrences of x, the expected co-occurrence frequency of (x, y) is $E(x, y) = 2F(x)E_\Delta$[1] $= \frac{2F(x)F(y)}{F(x)+F(y)}(1 - e^{-\frac{(F(x)+F(y))T}{Dur(s)}})$.

Problem Definition: Given a stream S of length N, a window size T, two threshold $0 < \alpha < 1$ and $\beta > 1$, the problem of discovering correlated items in the stream is to find any un-ordered item pair (x, y) that satisfies the following conditions: (i) x is a frequent item, $F(x) > \lceil \alpha N \rceil$, (ii) y is a frequent item, $F(y) > \lceil \alpha N \rceil$, (iii) x and y are *positively correlated*, $\frac{F(x,y)}{E(x,y)} > \beta$.

[1] Strictly, the occurence of (x, y) in the fist and last T interval on the stream should be discussed differently. However, because $T \ll Dur(S)$, such differences are neglected.

4 Algorithm

To discover correlated item pairs in a data stream, we propose the *Stream-Correlation* algorithm. According to the problem definition in Section 3, by scanning the stream once, we need to complete the following three tasks: 1) approximately find all frequent items from the stream, and meanwhile 2) for any two frequent items x and y, count the co-occurrence frequency for (x, y) with possible bounded error, and finally 3) approximately output the correlated item pairs. Section 4.1, 4.2, 4.3 will discuss these three tasks respectively.

4.1 Finding Frequent Items

We apply the *Space-Saving* algorithm, proposed in [1], to fulfil this task. For the completeness of the representation, we briefly introduce the algorithm. In the algorithm, a data structure, called *Stream-Summary*, is used to dynamically maintain the fixed number (m) of counters for items. Each counter consists of three fields: the corresponding item e_i, the estimated frequency of the item $Count(e_i)$, and the maximum possible over-estimation of the frequency, $\epsilon(e_i)$. For brevity, we can regard the *Stream-Summary* as a list of counters for m items $e_1, \ldots, e_m$, which are always decendingly ordered by their estimated frequencies.

Algorithm 1 is the pseudocode for the *Space-Saving* algorithm. When an event $(a_I,\ t_I)$ in the stream S arrives, if the item a_I is monitored in the *Stream-Summary*, the counter of a_I is incremented. Otherwise, a_I takes the place of e_m, which is the item that currently has the least estimated frequency *min*; also, we set $Count(a_I)$ as $min + 1$ and the over-estimation $\epsilon(a_I)$ as *min*.

In [1], it is proved that regardless of the data distribution and user-supplied frequency threshold, to find all frequent items with the maximal possible error $\epsilon \in (0, 1)$, the *Space-Saving* algorithm only requires to maintain $\lceil \frac{1}{\epsilon} \rceil$ number of counters, i.e., $m = \lceil \frac{1}{\epsilon} \rceil$. For any item e with the *true frequency* $F(e) > \epsilon N$ is guaranteed to be in the *Stream-Summary*. Also, for any item e_i in the *Stream-Summary*, we always have $Count(e_i) - \epsilon N \leq F(e_i) \leq Count(e_i)$.

4.2 Finding Co-occurrence Frequency for Frequent Item Pairs

In this section, based on the *Spacing-Saving* algorithm, we propose our key algorithm, *Stream-Correlation*, to find the co-occurrence frequency for item pairs.

Algorithm 1. The Space-Saving Algorithm (Counters m, Stream S)

```
for each event (a_I, t_I) in S do
    if a_I is monitored then
        Increment the counter of a_I
    else
        Let e_m be the item with least estimated frequency, min
        Replace e_m with a_I
        Assign Count(a_I) with min + 1
        Assign ε(a_I) with min
    end if
end for
```

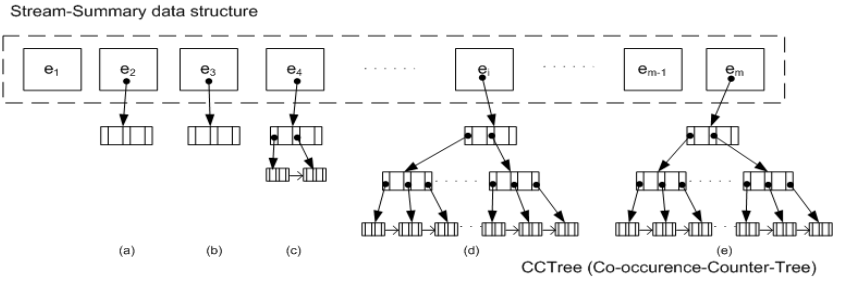


Fig. 2. *Correlation-Summary* data structure

Considering that the volume of stream is very high and the number of items is large, obviously, it is impossible to maintain a counter for each item pair. Naturally, the following **principle** is applied: *We count the co-occurrence of x and y only when both x and y are potentially frequent. In other words, a co-occurrence counter $Count(x, y)$ is allocated to item pair (x, y) iff both x and y are currently stored in the Stream-Summary.*

Once the principle is set, it is clear that if the number of items in the *Stream-Summary* is m, we need to maintain at most $\frac{m(m-1)}{2}$ co-occurrence counters. The challenge now becomes how to organize these co-occurrence counters such that they can be incrementally maintained in an efficient way. In the remainder of this section, we first design a data structure, *Correlation-Summary*, to effectively organize the co-occurrence counters. Based on the *Correlation-Summary*, we develop our algorithm *Stream-Correlation* to efficiently count the co-occurrence frequency for item pairs. Finally, we discuss the complexity of the algorithm.

***Correlation-Summary* data structure.** Figure 2 shows the data structure, which consists of two levels. The top level is the *Stream-Summary* data structure which has been introduced in Section 4.1. At any moment, the items are decendingly ordered by their estimated count, denoted as $e_1, e_2, \ldots, e_m$. At the second level, for each item e_i in the *Stream-Summary*, we maintain a group of co-occurrence counters which are associated with e_i, denoted as G_{e_i}. Considering an item pair (e_i, e_j), intuitively, the co-occurrence counter $Count(e_i, e_j)$ could be put either in group G_{e_i} or in group G_{e_j}. However, in our data structure, we enforce the **constraint** that *a co-occurrence counter $Count(e_i, e_j)$ always associates with the item that has lower estimated frequency. That is, $Count(e_i, e_j) \in G_{e_i}$ only when $Count(e_j) \geq Count(e_i)$*. From this constraint, we know that for any item e_i $(1 \leq i \leq m)$ in the *Stream-Summary*, there are at most[2] $i-1$ co-occurrence counters in G_{e_i}, i.e., $Count(e_i, e_1), \ldots, Count(e_i, e_{i-1})$. As a result, when the least frequent item e_m in the *Stream-Summary* is replaced, all the co-occurrence counters that contain the item e_m can be dumped by simply removing G_{e_m}. This guarantees that co-occurrence counters are only maintained for items which are currently in the *Stream-Summary*. To facilitate the update

[2] Some co-occurrence counters can be missing if the corresponding item pairs have not been counted yet.

on co-occurrence counters, for any item e_i, we adopt the B^+ tree structure to organize the co-occurrence counters in G_{e_i}. We call this B^+ tree a CCTree (co-occurrence counter tree) of e_i, denoted as CT_{e_i}. An item e_i and its CCTree CT_{e_i} are linked with a pointer pointing to the root of the tree. Note that without loss of generality, each item can be mapped into an integer. So, a leaf node in the CCTree of e_i is in the form of $< e_j, Count(e_i, e_j) >, 1 \leq j \leq i-1$, where e_j is the key of the node and $Count(e_i, e_j)$ gives the co-occurrence frequency for item pair (e_i, e_j). With this data structure, given two items e_i and e_j, their co-occurrence counter can be quickly located in the *Correlation-Summary*.

Incremental update algorithm. Now we present the complete algorithm *Stream-Correlation*, as shown in Algorithm 2. Essentially, this algorithm is to incrementally maintain the structure *Correlation-Summary* when a new event occurs. For any event (a_I, t_I) in the data stream, we update *Correlation-Summary* in two steps. First, the *Stream-Summary* is updated. Note that according to the **constraint**, this update may lead to the change on the structure of CCTrees as well. Second, we check the events occurred in $[t_I - T, t_I)$ and increment co-occurrence counters if the minimal occurrences (MOs) are detected.

In the first step, we follow the *Space-Saving* algorithm to update the counter of a_I. Under the following two circumstances, the CCTree(s) needs to be updated as well. First, when the least frequent item e_m is replaced by a_I, the CCTree of a_m is dumped and the CCTree of a_I is initialize as empty. Second, due to the increase of $Count(a_I)$, a_I may swap position with other item(s), say e_i. To enforce the **constraint**, the CCTree of a_I, CT_{a_I}, and the CCTree of e_i, CT_{e_i}, should be updated as follows: remove the node with the key e_i from CT_{a_I}, change the key of this node from e_i to a_I, and insert the node into CT_{e_i}.

In the second step, for any event (a_I, t_I), we update the co-occurrence counters as follows. At the moment t_I, we always buffer the events occurring in the window $[t_I - T, t_I)$, denoted as $(a_{I-k}, t_{I-k}), \ldots, (a_J, t_J), \ldots, (a_{I-1}, t_{I-1})$, where k is the number of events in the current window. According to the *Space-Saving* algorithm, the new arriving item a_I is guaranteed to be included in the *Stream-Summary*, so we check previously occurred event (a_J, t_J) (where J from $I-1$ to $I-k$) to see whether item pair (a_J, a_I) needs to be counted. According to the **principle**, we only consider the event (a_J, t_J) where a_J is currently in the *Stream-Summary*. If $a_J \neq a_I$, $Count(a_J, a_I)$ is increased by one because events (a_J, t_J) and (a_I, t_I) form an MO of (a_J, a_I). Otherwise (in the condition $a_J = a_I$), we need to 1) remove (a_J, t_J) from window $[t_I - T, t_I)$, and 2) stop further checking the rest events occurred in the window $[t_I - T, t_J)$. Let us first explain the operation 2). For any event $(a_{J'}, t_{J'})$ occurring earlier than (a_J, t_J) where $t_I - T \leq t_{J'} < t_J$, according to the definition of MO, events $(a_{J'}, t_{J'})$ and (a_I, t_I) cannot form an MO for item pair $(a_{J'}, a_I)$ because of the existence of event (a_J, t_J). Considering the operation 1), due to similar reason, for any event $(a_{I'}, t_{I'})$ arriving later than (a_I, t_I) (i.e, $t_{I'} > t_I$), event (a_J, t_J) and $(a_{I'}, t_{I'})$ cannot be matched as an MO for item pair $(a_J, a_{I'})$. So, (a_J, t_J) will not be considered any more and should be removed.

Algorithm 2. The Stream-Correlation Algorithm (Counters m, Stream S)

```
for each event (a_I, t_I) in S do
    /*Step1: find frequent items in stream */
    Update Stream-Summary according to Algorithm 1;
    Adjust the CCTree(s) in Correlation-Summary if necessary;
    /*Step2: update co-occurrence counter*/
    Adjust the events in a recent T interval, denoted as
    S_T = (a_{I-k}, t_{I-k}), ..., (a_J, t_J), ..., (a_I, t_I);
    for each event (a_J, t_J) where J from I − 1 downto I − k do
        if (a_I == a_J) then
            Remove (a_J, t_J) from S_T;
            break;
        else if a_J is in the Stream-Summary then
            Update co-occurrence counter Count(a_I, a_J) in Correlation-Summary;
        end if
    end for
end for
```

When a co-occurrence counter $Count(a_J, a_I)$ requires an update, to locate the $Count(a_J, a_I)$ in the *Correlation-Summary*, we first find the item (from a_J and a_I) with the lower rank in the *Stream-Summary*. Suppose that such item is a_I, i.e., $Count(a_I) \leq Count(a_J)$. We search the node with key a_J from the CCTree of a_I. If such a node exits, $Count(a_J, a_I)$ is incremented. Otherwise, it means that the co-occurrence of (a_J, a_I) is counted at the first time, then a new node, with key a_J and $Count(a_J, a_I) = 1$, is initialized and inserted into CT_{a_I}.

Complexity. Now we discuss the space and time complexity of the *Stream-Correlation* algorithm.

According to the structure of *Correlation-Summary*, it is straightforward that given the size of *Stream-Summary* m, the *Stream-Correlation* algorithm requires at most $\frac{m(m+1)}{2}$ counters.

Let us consider the time complexity in terms of processing time per event. As mentioned in previous part, for each event, the algorithm updates the *Correlation-Summary* in two steps. For the first step, it has been proved in [1] that the update on the *Stream-Summary* takes $O(1)$ amortized cost. Now we consider the cost related to updating the structure of CCTrees. When the last item e_m is replaced by a_I, updating CCTrees requires $O(1)$ time. For the situation that two items need to swap their positions in the *Stream-Summary*, the cost for this operation is $O(\log m)$, which is the complexity for removing / inserting a node from a B^+ -tree of size m. So, updating the counter for a single item in *Correlation-Summary* requires $O(c_0 \log m)$[3] amortized cost per event. Considering the second step, updating a co-occurrence counter $Count(e_i, e_j)$ in the co-occurrence counter tree requires at most $O(\log m)$ cost. For each event (a_I, t_I) in the data stream, the *Stream-Correlation* algorithm needs to update at most k co-occurrence counters associate with a_I, where k is the number of events in the interval $[t_I - T, t_I)$ and $k < T$. So, the time complexity for second step is at most $O(T \log m)$. By combining the above analysis, we know that the *Streaming-Correlation* algorithm has at most $O((T + c_0) \log m)$ processing time per event in the data stream.

[3] An item can swap its position with more than one item in *Stream-Summary*. We consider that the average number of swaps per event is bounded by a constant c_0.

4.3 Error Bound Analysis and Computing Correlated Item Pairs

Recall that given the size of *Stream-Summary* m, the error bound ϵ for the frequency of a single item is $\frac{1}{m}$. That is, for any item e_i in the *Stream-Summary*, we have $F(e_i) \in [Count(e_i) - \epsilon N, Count(e_i)]$, where $F(e_i)$ is the true frequency of item e_i. Now, we discuss the error bound for the co-occurrence frequency.

Theorem 2. *For any two items e_i and e_j in the Stream-Summary, let $F(e_i, e_j)$ be the true value of co-occurrence frequency for item e_i and e_j in a stream of length N. Given the error bound ϵ for the single item, the Stream-Correlation algorithm guarantees that the condition $Count(e_i, e_j) \leq F(e_i, e_j) \leq Count(e_i, e_j) + 2\epsilon N$ always holds. This is true regardless of the item distribution in the stream.*

Proof. Because both e_i and e_j are currently recorded in the *Stream-Summary* (with their error $\epsilon(e_i)$ and $\epsilon(e_j)$ respectively), there must exist a moment t such that 1) since t, both e_i and e_j are in the *Stream-Summary*, and 2) $\nexists\, t' < t$ and t' satisfies the condition 1). According to the **principle**, $Count(e_i, e_j)$ is the true number of co-occurrences for e_i and e_j after t. So, it is obvious that $Count(e_i, e_j) \leq F(e_i, e_j)$. Suppose that e_i is the item which is not recorded in the *Stream-Summary* until the moment t. From the *Space-Saving* algorithm, we know that e_i can be maximally overestimated $\epsilon(e_i)$ times before t. Note that according the definition of MO, one occurrence of e_i can lead to two occurrences of (e_i, e_j) at maximum. So, the number of co-occurrences of (e_i, e_j) before t cannot be greater than $2\epsilon(e_i)$. Therefore, in general, we have $F(e_i, e_j) \leq Count(e_i, e_j) + 2Max\{\epsilon(e_i), \epsilon(e_j)\} \leq Count(e_i, e_j) + 2\epsilon N$.

Theorem 3. *Given the error bound τ $(0 < \tau < 1)$ for the co-occurrence frequency, the Stream-Correlation algorithm requires $\frac{\lceil \frac{2}{\tau} \rceil (\lceil \frac{2}{\tau} \rceil + 1)}{2}$ counters.*

Proof. First, according to Theorem 2, the error bound τ equals to 2ϵ. Second, we know that $m = \lceil \frac{1}{\epsilon} \rceil$. Third, the *Stream-Correlation* algorithm requires at most $\frac{m(m+1)}{2}$ counters. The theorem is proved based on the above three points.

Finally, we discuss the procedures of approximately generating correlated item pairs from the *Correlation-Summary*. For any two items x and y, we have $F(x) \in [Count(x) - \epsilon(x), Count(x)]$, $F(y) \in [Count(y) - \epsilon(y), Count(y)]$ and $F(x, y) \in [Count(x, y), Count(x, y) + 2Max\{\epsilon(x), \epsilon(y)\}]$. The approximate correlation discovery can take two approaches, i.e., false positive oriented and false negative oriented. The former may include some item pairs which are not correlated in terms of the given threshold, while the latter may miss some correlated item pairs. Let $E^+(x, y)$ ($E^-(x, y)$ respectively) be the approximate expected co-occurrence frequency computed by the upper (lower respectively) bounds of $F(x)$ and $F(y)$. The condition for false positive approach is $\frac{Count(x,y)+2Max\{\epsilon(x),\epsilon(y)\}}{E^-(x,y)} > \beta$, while the condition for false negative one is $\frac{Count(x,y)}{E^+(x,y)} > \beta$. In the procedures, for any item e_i that is frequent, we need to traverse its co-occurrence counter tree CT_{e_i}. For any node corresponding to e_j, if the condition (either false positive or false negative) is satisfied, the item pair (e_i, e_j) is output.

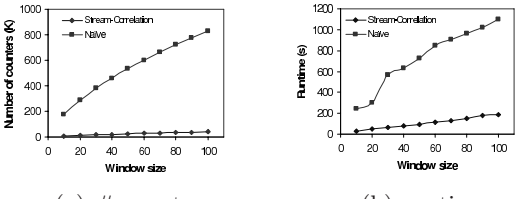
5 Experiment Results

In this section, we show the effectiveness and the efficiency of the *Stream-Correlation* algorithm by comparing it with the *naive approach*, which maintains a counter for every item and item pair. Although the naive approach guarantees the accurate result for the problem of correlation analysis, it is not practical for the dataset with a large number of item types because the required size of memory will be too large. We implement the algorithms by Java and all experiments are conducted on a PC with 3 GHz CPU and 1 gigabytes memory, running Microsoft Windows XP. The dataset used in the experiments is a sequence of URLs fetched from the web proxy servers of an university. Due to privacy issue, there is no timestamp information for each HTTP request, and the URLs are truncated. The number of URLs in the dataset is 10^6, which means $Dur(s) = N = 10^6$.

For finding the positive correlated item pairs from the dataset, we always set the thresholds as $\alpha = 0.005$ and $\beta = 1.5$. The number of single counters in the *Stream-Correlation* is 500, which indicates that the maximal possible error for the item frequency and co-occurrence frequency are $\frac{1}{500}$ and $\frac{1}{250}$ respectively.

We test the *Stream-Correlation* algorithm and the naive algorithm with different window size T (from 10 to 100). Let us fist evaluate the effectiveness of the *Stream-Correlation* by comparing its output with the accurate result (the output of the naive approach). In all the conducted experiments, the *Stream-Correlation* algorithm achieves both recall and precision equal to 1, which means that there is no accuracy loss in terms of the item pairs discovered in the data.

The efficiency of the algorithm is measured by 1) the number of counters used in the algorithm and 2) the runtime of the algorithm. Figure 3(a) shows that the number of counters maintained by the *Stream-Correlation* is far less than that of the naive approach. This is because in the real dataset, the number of items (different URLs) are very large. As a result, it is very memory-consuming to maintain a counter for every item and item pair. We do have some experiments in which the naive approach can not work due to running out of memory. In Figure 3(b), we can see that the *Stream-Correlation* algorithm is much faster than the naive approach in terms of the runtime because it maintains and processes significantly smaller number of counters.



(a) # counters (b) runtime

Fig. 3. Efficiency comparison between Stream-Correlation and Naive approach

6 Conclusion

In this paper, we have investigated the problem of finding correlated item pairs in a data stream. We define that two items x and y are correlated if both items are frequent, and their actual number of co-occurrences in the data stream is significantly different from the expected value. By modelling the occurrences of each type of item as a Poisson process, we give the expected number of minimal occurrences of (x, y), which is computed based on the frequencies of x and y. A one-pass algorithm has been proposed with the focus on estimating the actual number of co-occurrences of item pairs. The algorithm can efficiently discover the correlated item pairs with a bounded error by using limited memory space. The experiment results on the real data show that compared with the naive approach, our algorithm can significantly reduce the runtime and the memory usage, but without the loss of accuracy on the discovered item pairs.

References

1. Metwally, A., Agrawal, D., Abbadi, A.E.: Efficient computation of frequent and top-k elements in data streams. In: ICDT. (2005) 398–412
2. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: PODS. (2002) 1–16
3. Golab, L., Ozsu, M.T.: Issues in data stream management. SIGMOD Rec. **32**(2) (2003) 5–14
4. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. In: ICALP. (2002) 693–703
5. Cormode, G., Muthukrishnan, S.: What's hot and what's not: tracking most frequent items dynamically. In: PODS. (2003) 296–306
6. Demaine, E., Lopez-Ortiz, A., Munro, J.: Frequency estimation of internet packet streams with limited space. In: Algorithms - ESA 2002. 10th Annual European Symposium. Proceedings, Rome, Italy, Springer-Verlag (2002) 348–60
7. Estan, C., Varghese, G.: New directions in traffic measurement and accounting: Focusing on the elephants, ignoring the mice. ACM Trans. Comput. Syst. **21**(3) (2003) 270–313
8. Jin, C., Qian, W., Sha, C., Yu, J.X., Zhou, A.: Dynamically maintaining frequent items over a data stream. In: CIKM. (2003) 287–294
9. Karp, R.M., Shenker, S., Papadimitriou, C.H.: A simple algorithm for finding frequent elements in streams and bags. ACM Trans. Database Syst. **28** (2003) 51–55
10. Singh Manku, G., Motwani, R.: Approximate frequency counts over data streams. In: VLDB. (2002) 346–57
11. Metwally, A., Agrawal, D., Abbadi, A.E.: Using association rules for fraud detection in web advertising networks. In: VLDB. (2005)
12. Anupam, V., Mayer, A.J., Nissim, K., Pinkas, B., Reiter, M.K.: On the security of pay-per-click and other web advertising schemes. Computer Networks **31** (1999) 1091–1100
13. Mannila, H., Toivonen, H.: Discovering generalized episodes using minimal occurrences. In: KDD. (1996) 146–151
14. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: generalizing association rules to correlations. In: SIGMOD. (1997) 265–276

Incremental Clustering in Geography and Optimization Spaces

Chih-Hua Tai[1], Bi-Ru Dai[2], and Ming-Syan Chen[1]

[1] National Taiwan University
hana@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw
[2] National Taiwan University of Science and Technology
brdai@csie.ntust.edu.tw

Abstract. Spatial clustering has been identified as an important technique in data mining owing to its various applications. In the conventional spatial clustering methods, data points are clustered mainly according to their geographic attributes. In real applications, however, the obtained data points consist of not only geographic attributes but also non-geographic ones. In general, geographic attributes indicate the data locations and non-geographic attributes show the characteristics of data points. It is thus infeasible, by using conventional spatial clustering methods, to partition the geographic space such that similar data points are grouped together. In this paper, we propose an effective and efficient algorithm, named incremental clustering toward the Bound INformation of Geography and Optimization spaces, abbreviated as *BINGO*, to solve the problem. The proposed *BINGO* algorithm combines the information in both geographic and non-geographic attributes by constructing a summary structure and possesses incremental clustering capability by appropriately adjusting this structure. Furthermore, most parameters in algorithm *BINGO* are determined automatically so that it is easy to be applied to applications without resorting to extra knowledge. Experiments on synthetic are performed to validate the effectiveness and the efficiency of algorithm *BINGO*.

1 Introduction

Due to the widespread use of satellite surveillance system, geographic information system (GIS), cellular phones, and sensor networks, vast spatial data with geographic attributes are obtained and collected every day. In light of the useful information from these data, spatial data clustering, which is an important technique in data mining, has received a significant amount of research attention for years [2,5,7]. The main goal of spatial data clustering is to group data points according to the properties in their geographic attributes. In most cases, for example, each cluster is required to be connective to itself in the geographic space.

Conventional spatial clustering techniques[3,6,7,11,12,13,16], in general, can be divided into partition-based methods, hierarchical methods, density-based

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 272–283, 2007.

methods, and grid-based methods. The partition-based clustering algorithms such as CLARANS [12] start with an initial partition of the data set, and then iteratively optimize an objective function by moving the data points among k clusters until the optimal partition is reached. Hierarchical methods use either a top-down splitting manner or a bottom-up merging manner to generate the clusters. CURE [6] is an example of such methods. Density-based clustering algorithms such as DBSCAN [3] create clusters according to the density information of geographic regions. Grid-based methods such as WaveCluster [13] quantize the data space into finite grids and then perform clustering on these grids. According to the geographic attributes, most conventional spatial clustering methods group data points into connective and non-overlapped clusters. In addition, several novel techniques such as the works in [4,14,15] discuss clustering spatial data in presence of physical constraints.

The real applications, however, have called for the need of new clustering methods to deal with spatial data points with non-geographic attributes. Note that data points obtained in many real applications such as weather observation station and sensor network consist of both geographic and non-geographic attributes at the same time. In general, geographic attributes indicate the data locations, whereas non-geographic attributes show the characteristics of data points. To explore interesting information from this kind of data, both the connective and non-overlapped properties in the geographic space, which are common requirements in spatial data clustering, and the data similarity in their non-geographic attributes should be considered in the generation of clusters. However, in most conventional spatial clustering methods, only the geographic attributes are taken into consideration. It is thus infeasible, by using conventional spatial clustering methods, to partition the geographic space such that the data points in the same subregions are similar in their non-geographic attributes. More specifically, when we consider the data points consisting of geographic and non-geographic attributes, named as dual data points in this paper, the clustering problem of partitioning the geographic space such that the dissimilarity between data points in the same subregions is minimized cannot be directly dealt with by any of conventional spatial clustering algorithms. We further explain this problem by the examples below.

Example 1.1: Consider the Data set 1 in Figure 1(a), where the black triangles and white diamonds are used to represent the clustering result of K-means applied to the non-geographic attributes. Given such a data set, conventional spatial clustering algorithms concern only the geographic attributes and thus generate clusters as shown in Figure 1(b). In Figure 1(b), the black points form a cluster, the white points form another cluster, and the gray ones are regarded as noises. As a result, the dissimilar data points are grouped together because of their positions in the geographic space. □

Example 1.2: Consider the same data set in Figure 1(a). An alternative method to combine the information from both geographic and non-geographic attributes is using clustering algorithms such as K-means with an extended objective

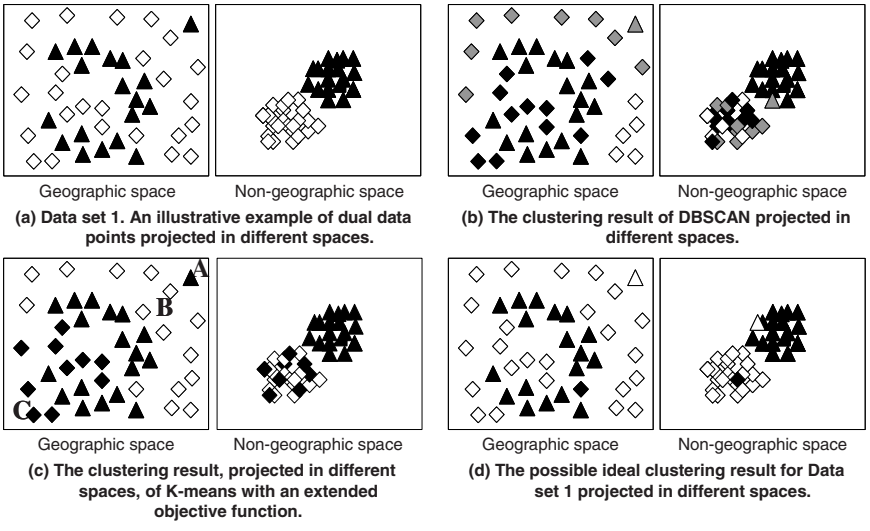


Fig. 1. Illustrative examples of dual data points

function which takes a weighted average between geographic and non-geographic attributes. Consequently, clusters may be generated as shown in Figure 1(c), in which data points are assigned to either a black cluster or a white cluster. Note that the black cluster does not form a connective non-overlapped geographic region due to the low similarity between the black cluster and the white diamond points such as B (or due to the high similarity between the black cluster and the black triangle points such as A.) On the other hand, some similar data points such as B and C may be considered dissimilar because of their locations in the geographic space being too far away from one another. This kind of methods suffer from measuring the concept of connection and non-overlap of a cluster in distances. □

In comparison with the clustering results shown in Figures 1(b) and 1(c), the result in Figure 1(d) is considered more preferable and useful because each cluster is a connective and non-overlapped region in the geographic space while the dissimilarity between data points in the cluster is smaller (the result projected in the non-geographic space in Figure 1(d) is purer then that in Figure 1(c).) Such result not only conforms to the common requirements of spatial data clustering but also takes account of minimizing the dissimilarity between data in clusters.

One of the main challenges for clustering dual data points is to deal with both the geographic and non-geographic attributes at the same time such that data points in the same cluster form a connective but non-overlapped region in the geographic space and their dissimilarity in the non-geographic space is minimized. As usual, the dissimilarity between data points can be estimated from their distances in the non-geographic space. However, as shown in Example 1.2, it is inappropriate to substitute the connective meaning of a cluster with distances in the problem of clustering dual data points. Note that the distance is

an absolute measurement. In contrast, whether a cluster is connective or not will depend on the projection of other clusters in the geographic space. This concept makes the problem of clustering dual data points more challenging.

In [10], C.-R. Lin *et al.* proposed the first solution, algorithm ICC, for the problem of clustering dual data points. Based on the techniques of SVM [1,8] and complete-link [9] algorithms, their method conducts an interlaced process of performing clustering in the non-geographic space and classification in the geographic space to adjust the geographic scope of each cluster while minimizing the dissimilarity between points in the clusters. However, algorithm ICC suffers from inefficiency due to the usage of the complete-link algorithm. In addition, their results heavily depend on the parameters of SVM. Note that selecting optimal parameters of SVM is usually a difficult and time consuming task. It is thus hard to apply algorithm ICC to real applications due to the efficiency concerns, especially in situations where data points change as time goes by.

To remedy these problems, we propose in this paper the $BINGO$ (incremental clustering toward the Bound INformation of Geography and Optimization spaces) algorithm for solving the problem of clustering dual data points. Explicitly, we do not treat geographic attributes as distances, but instead, devise a summary structure, named $NeiGraph$, to integrate non-geographic attributes into geographic regions. Based on $NeiGraph$, we design $BINGO-OFF$ and $BINGO-ON$ incremental clustering methods for having high quality and high execution-efficiency results respectively. Furthermore, most parameters in our algorithms are determined automatically so as to facilitate users to use our methods. As shown in the experimental studies, our algorithm executes much more efficiently than algorithm ICC while being able to generate clusters to similar costs.

The rest of this paper is organized as follows. The problem definition is given in Section 2. Algorithm $BINGO$ and a series of illustrative examples are presented in Section 3. In Section 4, we conduct the performance studies. Finally, this paper concludes in Section 5.

2 Problem Description

In this paper, the data we dealt with consist of two types of attributes, geographic attributes and non-geographic attributes. Since the non-geographic attributes are used to infer the dissimilarity which we aim to minimize between data points in the same clusters, the non-geographic attributes are also called optimization attributes.

Given a database D_t containing dual data points at time t, we will generate clusters C_t that comply with the two principles below.

Principle 1. Projected to the space formed by geographic attributes, each cluster locates within a connective region while not overlapping with other clusters.

Principle 2. Clustering cost according to a cost function should be minimized.

In the following, precise definitions of our problem are given.

Definition 1. *(Dual Data Point) A dual data point is an object d with attributes* $\{a_1^G, a_2^G, ..., a_g^G, a_1^O, a_2^O, ..., a_o^O, \} \in \mathbb{R}^{g+o}$*, where attributes* $\{a_1^G, a_2^G, ..., a_g^G\}$ *form the geographic space G and attributes* $\{a_1^O, a_2^O, ..., a_o^O, \}$ *form the optimization space O.*

Definition 2. *(Event) An event* e_t *at time t is an adjustment of data points, including inserting new data points into and deleting existing data points from database* D_{t-1}*.*

Definition 3. *(Cost) Given clusters* $C_t = \{c_1, c_2, ..., c_k\}$ *of a database* D_t*, which contains* N_t *dual data points* $\{d_1, d_2, ..., d_{N_t}\}$*, the clustering cost is defined as*

$$Cost(C_t) = \frac{\sum_{c_k \in C_t} \sum_{d_i \in c_k} \sum_{d_i.a_j^O \in O} \left(d_i.a_j^O - c_k.a_j^O\right)^2}{N_t},$$

where $c_k.a_j^O$ *is the center of the attribute* a_j^O *of cluster* c_k*.*

Note that the clustering cost is defined only upon the optimization attributes.

Definition 4. *(Incremental Dual Data Clustering Problem) Whenever an event* e_t *occurs at time t, the clustering problem is to cluster the dual data points in database* D_t *into k groups such that each group locates a connective non-overlapped region in the geographic space G while minimizing the clustering cost,* $Cost(C_t)$*.*

3 Algorithm BINGO

Algorithm *BINGO* is a new approach for solving incremental dual data clustering problem. To comply with the principles of this problem, we devise a summary structure, name *NeiGraph*, and design the *BINGO* algorithm based on this structure. *NeiGraph* is a graph in which each node represents a connective non-overlapped region and the edge between two nodes implies that these nodes can be merged to be a new node. Algorithm *BINGO* consists of three major parts: the binding information (Bind-Info) procedure, the generating clusters (Generate-Clusters) procedure, and the tuning borders (Tune-Borders) procedure.

First, in the Bind-Info procedure, we partition the geographic space G into various sizes of grids and construct *NeiGraph* by taking each grid as a node and adding an edge between nodes if the corresponding grids are next to each other. *NeiGraph* thus can effectively combine the data information in the optimization space O into geographic regions by summarizing the optimization attributes of data points in a grid into the corresponding node. After that, the Generate-Clusters procedure selects representative nodes in *NeiGraph* as seeds, and then expands seeds along the edges to complete connective non-overlapped clusters. Whenever an event e^t occurs, the Tune-Borders procedure is activated to tune the borders among clusters and update the clustering results. We next show how *NeiGraph* is constructed and used to help generating clusters that comply with the principles of dual data clustering problem.

3.1 Binding Information

To overcome the challenge of generating connective non-overlapped clusters while minimizing the clustering cost, we first propose the concept of T-region to draw out a connective geographic region in which data points are considered to be similar to one another.

Definition 5. *(T-region) Given a real number T, a T-region is a geographic region in which any two data points are within the distance of T in the optimization space O.*

With the definition of T-region, we partition the geographic space G into various sizes of grids (T-regions) in the top-down manner. Specifically, starting from the largest grid which contains all data points in D_0, we iteratively examine every grid and divide the grid into 2^g equal-size sub-grids if it is not a T-region until all the grids are T-regions. At the same time, $NeiGraph$ is constructed by regarding each grid as a node and adding an edge between two nodes if the corresponding grids are next to each other. Data information in the optimization space O can also be bound in $NeiGraph$ by summarizing the optimization attributes of data points in a grid into the corresponding node. Therefore, $NeiGraph$ can effectively combine the data information in the geographic and optimization spaces when it is completely constructed, and provides a broader view of data points in the database. Moreover, we can even merge nodes to clarify $NeiGraph$ if there is an edge between the nodes and the merger would not change the property of being a T-region. Note that a node in $NeiGraph$ could be regarded as a micro-cluster of data points, and every micro-cluster forms a connective non-overlapped region in the geographic space G.

A problem left is how to decide the value of T such that data points in the same T-region can be considered to be similar to one another. A mechanism for automatically choosing a proper value of T is designed based on the lemma and the theorem below. For interest of space, proofs of theorem are shown in Appendix B.

Lemma 1. *Given a complete graph (a graph that there is an edge between any two nodes) consisting of n nodes, partitioning these n nodes into k groups will divide all edges into two categories, which are intra-edges (within groups) and inter-edges (between groups). The upper bound of the number of inter-edges appears when k groups have equal number of nodes.*

Theorem 1. *Given k clusters with totally n data points, there are at least $\frac{n^2-nk}{2k}$ intra-edges within clusters.*

Proof. According to Lemma 1, the maximum number of inter-edges between k clusters is obtained when each of the k groups is with the size of $\frac{n}{k}$. In other words, when grouping n points into k clusters, there are at most $\frac{n}{k} \times \frac{n}{k} \times C_2^k = (\frac{n}{k})^2 \frac{k(k-1)}{2}$ inter-edges between clusters. There are thus at least $C_2^n - (\frac{n}{k})^2 \frac{k(k-1)}{2} = \frac{n^2-n}{2} - \frac{n^2k^2-n^2k}{2k^2} = \frac{n^2k-nk^2}{2k^2} = \frac{n^2-nk}{2k}$ intra-edges within clusters no matter how the n points are grouped into k clusters. Consequently,

the minimum number of intra-edges within k clusters containing totally n data points is $\frac{n^2-nk}{2k}$. ■

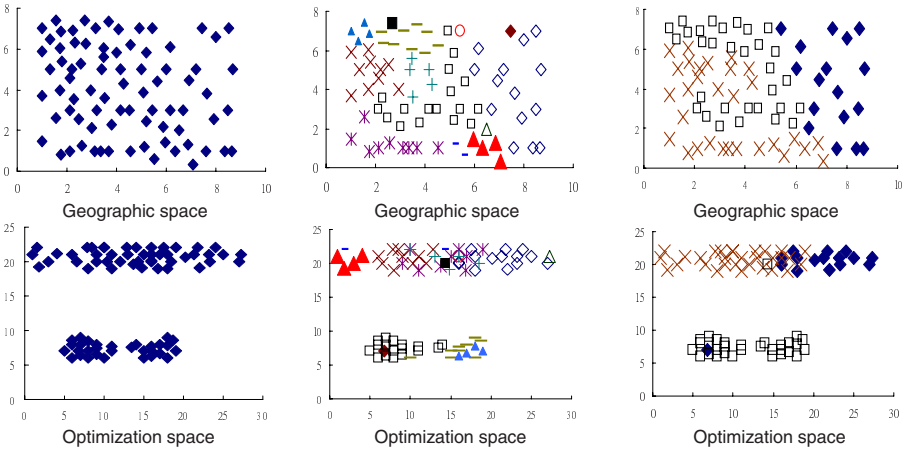
According to Theorem 1, for a data set with N_0 points, there are at least $\frac{N_0^2-N_0k}{2k}$ intra-edges within clusters. Therefore, we refer to the $\frac{N_0^2-N_0k}{2k}$-th smallest one among all the pairwise distances for capturing the concept of data points with high similarity in this paper. The value of T is then set as double of the reference distance. In other words, the distance between any data point in a T-region and the center of the corresponding micro-cluster is smaller than T so that the center is regarded having high similarity with the point and can be used to represent the point. Note that, although the value of T could be any smaller number, it is not necessary to generate results with lower clustering cost but incur higher execution time. Moreover, to further enhance the execution efficiency of our approach, the value of T could be approximated by a sampling mechanism since we do not need the precise value of T and the number of data points N_0 is usually quite large. The approximation of T from a sample data set is not only efficient but also satisfactory. In Example 3.1, we show the effectiveness of $NeiGraph$ constructed in this procedure.

Example 3.1: Given the Data set 2 containing 78 dual data points in Figure 2(a), we execute the Bind-Info procedure with $k = 3$. Figure 2(b) shows the projection of the result in different spaces, where data points presented in the same color and shape belong to the same T-region. There are a total of 13 T-regions produced. Note that each T-region is a node in $NeiGraph$. As Figure 2(b) shows, all the nodes represent connective and non-overlapped geographic regions while the data points in the same node are similar to one another in the optimization space O. □

3.2 Generation of Clusters

In the previous procedure, we combined the data information in the geographic and optimization spaces in $NeiGraph$. To farther conform to the principles of dual data clustering problem, the Generate-Clusters procedure follows the bound information to generate clusters in two major steps. The first step selects k dissimilar representative nodes in $NeiGraph$ as seeds of k clusters. The second step expands seeds along the edges between nodes to complete clusters.

Intrinsically, data points located nearby are possible to be dissimilar from one another. Because of the definition of T-region, it is possible to generate small nodes (the nodes with few data points) surrounded by large nodes in $NeiGraph$. The node with only one black square point in Figure 2(b) is an example of such case. To avoid choosing these small nodes as seeds and thus generating small clusters, only the nodes with size larger than average size are deemed representative and preferred to be seeds. In addition, to minimize the clustering cost, data points in different clusters are expected to be dissimilar to one another. We thus expect the seeds to be as dissimilar to one another as possible. In consideration of the execution complexity, however, we do not find the most dissimilar set of k seeds. Instead, beginning with choosing two most



(a) Data set 2 projected in different spaces.

(b) The output of Bind-Info procedure for Data set 2 projected in different spaces.

(c) The output of Generate-Clusters procedure for Data set 2 projected in different spaces.

Fig. 2. Illustrative examples of algorithm *BINGO*

dissimilar nodes as seeds, our method iteratively selects a node which has the lowest similarity to all the chosen seeds as a new seed until there are k seeds picked. Although this greedy method does not select the best set of seeds, our methods still possess the capability of gathering similar data points as shown in the experimental studies in Section 4.

After k seeds are chosen, each seed initially forms a cluster by itself. Then, k clusters are completed through the iterative merger process which stops when all the nodes in $NeiGraph$ are clustered. At each iteration, we identify the most similar pair of a cluster and its un-clustered neighboring node in $NeiGraph$, and then expand the cluster with the neighboring node. Note that each node in $NeiGraph$ represents a connective non-overlapped geographic region containing data points with high similarity. Our results thus conform to the principles of dual data clustering problem since the expanding process is greedily done along the edges in $NeiGraph$. Example 3.2 below gives a draft of the clustering results of algorithm $BINGO$.

Example 3.2: Following the result of Example 3.1, we execute the Generate-Clusters procedure to generate the clustering result for Data set 2. First, three nodes which contain data points marked as '◊', '□', and '×' are picked as seeds. Then, each seed forms a cluster and is expanded along the edges to generate the clustering result as shown in Figure 2(c). Note that each cluster is connective and non-overlapped in the geographic space G. On the other hand, clusters in the optimization space O still tend to gather similar data points under the connective and non-overlapped constraints of clusters in the geographic space G. □

3.3 Tuning Cluster Borders

Note that data points are allowed to change as time goes by. In order to provide the incremental clustering capability of our approach, we propose an incremental tuning mechanism to refine clusters when data points are updated. Generally, data points in the database D_{t-1} can be updated through the deletion and insertion operations at time t. Although we handle only the deletion and insertion operations in this paper, we are able to deal with the modifications of data points since a modification can be treated as a deletion followed by an insertion. We now present the details of data adjustment process.

(I) Deletion:

When a data point d_r is removed from the database D_{t-1}, delete d_r from node r_i and cluster c_j which d_r is assigned to. Then, add r_i into the changed set S_c.

(II) Insertion:

When a data point d_a is added to the database D_{t-1}, locate d_a into the corresponding node r_i in $NeiGraph$ according to its position in the geographic space G, and assign d_a to the cluster c_j which contains r_i. Then, add r_i into the changed set S_c.

After the data adjustment occurs, the Tune-Borders procedure improves the clustering results to reduce the cost in the optimization space O. In this process, we design a greedy strategy to revise the clustering results based on $NeiGraph$. The concept of this strategy is to expand each cluster if possible by iteratively drawing in its neighboring nodes in the geographic space G from other clusters such that a lower clustering cost can be obtained. The non-overlapped and connective constraints of clusters in the geographic space G can also be complied with by checking the connective region of each cluster. Although this process seems to be expensive, only the nodes influenced directly by the data adjustment and the nodes that become new neighboring nodes of some clusters during the expanding iterations are possible to change their clusters.

The revision of clustering results depends on the information summarized in $NeiGraph$. The nodes in $NeiGraph$, however, may not be T-regions anymore because of the data adjustment. Therefore, it is important to decide a suitable time for reforming the harmful nodes (the nodes which are not T-regions) in $NeiGraph$. Nevertheless, this decision incurs the trade-off between the execution efficiency and the clustering quality. A better way for balance is to check the nodes only when we use the information carried by the nodes. Hence, if a node in $NeiGraph$ is not a T-region, it is split by the same method in Section 3.1 only at the iteration that there is a cluster trying to pull it in. That is, only the nodes carrying the information used to tune the borders of clusters will be examined and, if necessary, split. For those changed but not examined nodes, we will record them in the changed set S_c until they are examined.

In addition to the Tune-Borders procedure, an alternative way to achieve the incremental clustering capability of our approach is to reform every harmful nodes in $NeiGraph$ whenever data adjustment occurs, and then activate the

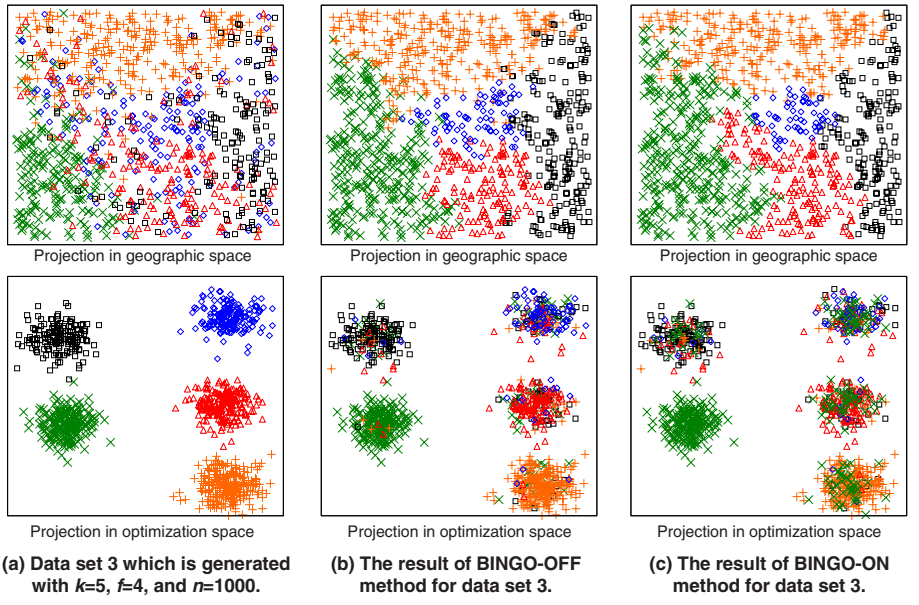


Fig. 3. Visualization of Dataset 3 and the clustering results of algorithms

Generate-Clusters procedure in Section 3.2 to produce clustering results. Compared to the Tune-Borders procedure, this method can generate clusters with lower cost. However, it also incurs higher time complexity due to the modification of whole $NeiGraph$ in the worst case. Therefore, we regard this method as the $BINGO-OFF$ method for having high quality results, and regard the Tune-Borders procedure as the $BINGO-ON$ method for having high execution-efficiency results. In Section 4, we conduct a series of experiments to compare the performances of these methods.

4 Performance Studies

In this section, we conduct a series of experiments to assess the performance of algorithm $BINGO$ on a computer with a CPU clock rate of 3 GHz and 480 MB of main memory. The simulation model used to generate synthetic data is the same as that used in [10]. In Section 4.1, we show the clustering effectiveness of algorithm $BINGO$ with visualization of outputs. Results on scaleup experiments are presented in Section 4.2.

4.1 Experiment I: Effectiveness

In this experiment, we apply algorithm $BINGO$ to a complex data set in Figure 3(a), and demonstrate that our methods can achieve fine clustering quality with

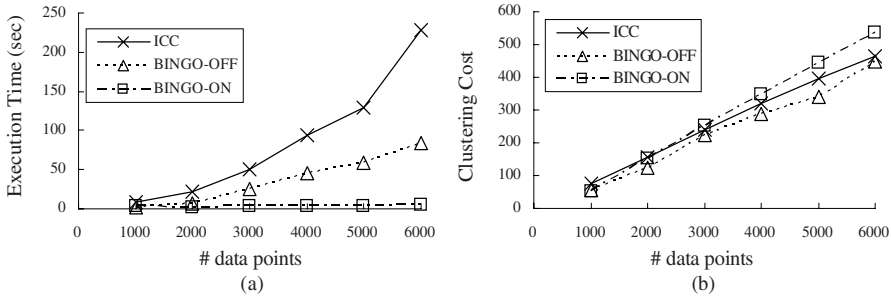


Fig. 4. (a) Scaleup performance of ICC, BINGO-OFF, and BINGO-ON, and (b) the corresponding clustering costs

the visualization of the results in both the geographic space G and the optimization space O. The result of our $BINGO-OFF$ method is shown in Figure 3(b), and the result of our $BINGO-ON$ method is shown in Figure 3(c). Note that the generated clusters in Data set 3 are heavily overlapped in the geographic space G. Both $BINGO-OFF$ and $BINGO-ON$ still produce connective and non-overlapped clusters in the geographic space G while gathering up similar data points in the optimization space O with best efforts. To further evaluate the clustering effectiveness on clustering costs, we compare our methods with algorithm ICC in the next experiment.

4.2 Experiment II: Scaleup Performance

We compare in this experiment the execution efficiency and the clustering cost of algorithm ICC and our methods. The execution time of these methods against different size of data sets is shown in Figure 4(a), and the corresponding clustering costs are shown in Figure 4(b). It can be seen that both of our methods execute more efficiently than algorithm ICC, especially the $BINGO-ON$ method. In addition, our $BINGO-OFF$ method can generate clusters with similar cost to that of ICC. The $BINGO-ON$ method, on the other hand, incurs higher cost due to the greedy policy in the Tune-Borders procedure. Therefore, when we concern more about the cost than the execution time, $BINGO-OFF$ is preferred. On the contrary, when the data size is quite large and the timing resource is limited, the $BINGO-ON$ method would be a better choice.

5 Conclusions

We proposed in this paper a new effective and efficient algorithm $BINGO$ for incremental clustering dual data points. Algorithm $BINGO$ integrated information in the geographic and optimization spaces by constructing a summary structure $NeiGraph$. Based on $NeiGraph$, the $BINGO-OFF$ and $BINGO-ON$ methods are designed to possess the incremental clustering capability and generate effective clustering results in which each cluster forms a connective and

non-overlapped region while gathering similar data points. Furthermore, most parameters in algorithm $BINGO$ are determined automatically so that it is easy to be applied to applications without resorting to extra knowledge. Experimental simulations have been performed to validate the effectiveness and the efficiency of algorithm $BINGO$.

Acknowledgements

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE.

References

1. A. Ben-Hur, D. Horn, H. Siegelmann, and V. Vapnik. Support vector clustering. *J. Machine Learning Research*, 2, 2001.
2. M.-S. Chen, J. Han, and P. S. Yu. Data mining: An overview from database perspective. *IEEE TKDE*, 8(6), 1996.
3. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of KDD*, 1996.
4. V. Estivill-Castro and I. Lee. Autoclust+: Automatic clustering of point-data sets in the presence of obstacles. In *Proc. of TSDM*, 2000.
5. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurasamy. *Advances in Knowledge Discovery and Data Mining*. Cambridge, Mass: MIT Press, 1996.
6. S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. In *Proc. of SIGMOD*, 1998.
7. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
8. D. Hush and C. Scovel. Polynomial-time decomposition algorithms for support vector machines. In *Machine Learning*, 2003.
9. B. King. Step-wise clustering procedures. *J. Am. Statistical Assoc.*, 69, 1967.
10. C.-R. Lin, K.-H. Liu, and M.-S. Chen. Dual clustering: Integrating data clustering over optimization and constraint domains. *IEEE TKDE*, 17(5), 2005.
11. A. Nanopoulos, Y. Theodoridis, and Y. Manolopoulos. C2p: Clustering based on closest pairs. In *Proc. of VLDB*, 2001.
12. R. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *Proc. of VLDB*, 1994.
13. G. Sheikholeslami, S. Chatterjee, and A. Zhang. Wavecluster: A wavelet based clustering approach for spatial data in very large database. *VLDBJ*, 8(3/4), 1999.
14. A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Spatial clustering in the presence of obstacles. In *Proc. of ICDE*, 2001.
15. O. R. Zaiane, A. Foss, C. H. Lee, and W. Wang. On data clustering analysis: Scalability, constraints, and validation. In *Proc. of PAKDD*, 2002.
16. J. Zhang, W. Hsu, and M. L. Lee. Clustering in dynamic spatial database. *Journal of Intelligent Information System*, 24(1), 2005.

Estimation of Class Membership Probabilities in the Document Classification

Kazuko Takahashi[1], Hiroya Takamura[2], and Manabu Okumura[2]

[1] Keiai University, Faculty of International Studies, 1-9 Sanno, Sakura, Japan
takak@u-keiai.ac.jp
[2] Tokyo Institute of Technology, Precision and Intelligence Laboratory
4259 Nagatsuta-cho Midori-ku, Yokohama, Japan
{takamura, oku}@pi.titech.ac.jp

Abstract. We propose a method for estimating class membership probabilities of a predicted class, using classification scores not only for the predicted class but also for other classes in a document classification. Class membership probabilities are important in many applications in document classification, in which multiclass classification is often applied. In the proposed method, we first make an accuracy table by counting the number of correctly classified training samples in each range or cell of classification scores. We then apply smoothing methods such as a moving average method with coverage to the accuracy table. In order to determine the class membership probability of an unknown sample, we first calculate the classification scores of the sample, then find the range or cell that corresponds to the scores and output the values associated in the range or cell in the accuracy table. Through experiments on two different datasets with both Support Vector Machines and Naive Bayes classifiers, we empirically show that the use of multiple classification scores is effective in the estimation of class membership probabilities, and that the proposed smoothing methods for the accuracy table work quite well. We also show that the estimated class membership probabilities by the proposed method are useful in the detection of the misclassified samples.

1 Introduction

When a classifier predicts a class for an evaluation sample in a document classification, estimating the probability with which the sample belongs to the predicted class (class membership probability) is useful in many applications. As an example for human decision making, we describe the need of class membership probabilities in "an automatic occupation coding system" in social surveys. The occupation coding is a task for various statistical analyses in sociology, in which researchers assign occupational codes to occupation data collected as responses to open-ended questions in social surveys. To help the human annotators (coders), we have developed an automatic occupation coding system with machine learning [13], as well as a system called the NANACO system [14], which displays outputs from the automatic system as candidates of occupational codes.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 284–295, 2007.

The NANACO system is currently being applied in important social surveys in Japan such as the JGSS [1], in which the coders have asked us to supply a measure of confidence for the first-ranked candidate of their confidece decisions. In fact, class membership probabilities are widely noticed in different applications [5,3].

Although the class membership probabilities can be estimated easily using classification scores provided from a classifier (hereafter referred to as *scores*) [2], estimates should be calibrated because they are often quite different from true values. Representative proposed methods include Platt's method [10] and various methods by Zadrozny et al. [15,16,17]. Platt [10] used Support Vector Machines (SVMs) and directly estimated class membership probabilities using a sigmoid function to transform scores into the probabilities. Zadrozny et al. proposed a "binning" method for Naive Bayes [15], and Isotonic regression for SVMs and Naive Bayes [17]. In the notable binning method (*Zadrozny's binning method*), the authors used the first-ranked scores of training samples, rearranged training samples according to their scores and made bins with equal samples per bin. In the Isotonic regression method, Zadrozny et al. proposed a method for a multiclass classifier by dividing a multiclass classifier into binary classifiers.

In the document classification, a multiclass classification is often applied. When a multiple classifier outputs a score for each class, a predicted class is determined not by the absolute value of the score but by the relative position among the scores. Therefore, the class membership probabilities are likely to depend not just on the first-ranked score but also on other scores. We propose a new method for estimating class membership probabilities of the predicted class, using scores not only for the predicted class but also for other classes. In the proposed method, we first make an accuracy table by counting the number of correctly classified training samples in each range or cell (hereafter referred to as *cell*) of scores. We then apply smoothing methods such as a moving average method to the accuracy table to yield reliable probabilities (accuracies). In order to determine the class membership probability of an unknown sample, we first calculate the scores of the sample, then find the cell that corresponds to the scores, and output the values associated in the cell in the accuracy table.

2 Related Work

2.1 Platt's Method

Platt [10] proposed a method using SVMs and directly estimating class membership probabilities using a sigmoid function $P(f) = 1/\{1 + exp(Af + B)\}$ with the score f because f is substituted into a monotonically increasing sigmoid function. The parameters A and B were estimated with the maximum

[1] Japanese General Social Surveys (`http://jgss.daishodai.ac.jp/`).

[2] In Naive Bayes or decision trees, scores can be the probabilities. Even in Support Vector Machines, scores can be transformed into probabilities by $(f - min)/(max - min)$ [8] or $(f + max)/2 * max$ [17], where f, max, and min represents scores of a sample, the minimum score, and the maximum score respectively.

likelihood method beforehand. To avoid overfitting the train set, Platt used an out-of-sample model. Using five types of datasets from Reuters [4] and other sources, a good experimental result was obtained for probability values. Bennett [2], however, using the Reuters dataset showed that the sigmoid method could not fit the Naive Bayes scores. Bennett proposed other sigmoid families in approximating the posterior distribution given the Naive Bayes log odds approximation. Zadrozny et al. [17] also showed that Platt's method could not be applied for some datasets and proposed a different approach.

2.2 Zadrozny's Binning Method

Zadrozny et al. [15] proposed the discrete non-parametric binning method, which indirectly estimates class membership probabilities by referring to the "bins" made for the Naive Bayes classifier beforehand. The method is described as follows. First, samples are rearranged in order of the values of their scores, and intervals are created to ensure that the number of samples falling into each area (bin) equals a fixed value. For each bin Zadrozny et al. computes lower and upper boundary scores. Next, the accuracy of samples in each bin is calculated. Finally, an evaluation of the new sample is done using the score to find a matching bin, and the accuracy of the bin is then assigned to the sample. Using KDD'98 datasets, a good experimental result was obtained in terms of some evaluation criteria such as Mean Squared Error (MSE) or average log-loss (bin=10). The method has a problem, however, in answering how the best number of bins can be determined.

2.3 Isotonic Regression and Expansion for a Multiclass Classifier

Based on a monotonic relationship between classifier scores and accuracies, Zadrozny et al. [17] next proposed a method via the PAV (Pair-Adjacent Violators) algorithm, which has been widely researched for problems of Isotonic regression. As a result of experiments for SVMs and Naive Bayes, PAV performed slightly better than the sigmoid method, while it always worked better than the binning method. For a multiclass classifier, they applied PAV as follows. First they transformed a multiclass classifier into binary classifiers by a one-against-all code matrix, all-pairs, and a one-against-all normalization. Next they calibrated the predictions from each binary classifier and combined them to obtain target estimates. The performance of PAV for a multiclass classifier using 20 Newsgroups dataset [3] for Naive Bayes was much better in terms of MSE, although the error rate was not good.

2.4 Comparison of the Methods

Niculescu-Mizil et al. [8] compared 10 classifiers for calibration using Platt's method with Zadrozny's method via Isotonic regression using 8 datasets from

[3] `http://people.csail.mit.edu/jrennie/20Newsgroups/`. Zadrozny et al. used the original dataset (19,997 documents).

UCI and other sources, and showed that Platt's method was most effective when the data was small, while Isotonic regression was more powerful when there was sufficient data to prevent overfitting [4]. Jones et al. [5] compared Isotonic regression with the sigmoid method, and showed that the sigmoid method outperformed Isotonic regression by root-mean squared error and log-loss. The reason for the outperformance was that Isotonic regression tended to overfit.

3 Proposed Method

We propose generating an accuracy table using multiple scores and by applying a smoothing method to the accuracy table as follows.

STEP 1 Create cells for an accuracy table.
STEP 2 Smooth accuracies.
STEP 3 Estimate class membership probability for an evaluation sample.

Before describing the details of STEP 1-3, we explain the reason for using multiple scores. We assume that a multiple classifier outputs a score for each class. A predicted class is determined not by the absolute value of the score, but by the relative position among the scores because the predicted class for an evaluation sample is a class with the largest value in multiple scores. For example, even if the first-ranked score is low, as long as the second-ranked score is very low (the difference between the two classes is large), the classification output will likely be reliable. In contrast, when the score for the first-ranked class is high, if the score of the second-ranked class is equally high (a negligible difference in score between the two classes), then the classification output is unreliable. Therefore, the class membership probabilities are likely to depend not just on the first-ranked score, but also on other scores. For effective calibration, it may be better to use not only the first-ranked score, but also other-ranked score.

STEP 1. To generate an accuracy table, we need a pair of scores and classification status (incorrect/correct) for each sample. To obtain these pairs, we divide the whole of the training dataset into two datasets: a) a training dataset to make "an accuracy table" and b) a test dataset for the table. We employed cross-validation. For example, in a 5-fold cross-validation, we divide the whole training data into five groups, and use four-fifths of the data to build a classifier with the remaining one-fifth of the data used to output scores from the classifier; we repeat the process four more times (a total of five times) by changing the data used for training and outputting the score to make an accuracy table.

We create cells for an accuracy table as follows. First, the score is used as an axis, divided into even intervals. For example, the size of an interval may be 0.1 on SVMs. In the case of using multiple scores, this step takes place for each score. When we use the first-ranked scores and the second-ranked scores, we split a rectangle up into several intervals. Second, we decide, on the basis of the score,

[4] Niculescu-Mizil et al. also showed that both Platt's method and Isotonic regression improved the probabilities predicted by any boosting model.

to which cell each training sample belongs. Finally, we check the classification status (correct/incorrect) of the training samples in each cell and calculate the accuracy of that cell, that is, its ratio of correctly classified samples. In this method, an accuracy table can be made for any number of scores (dimensions) used because the training samples do not need be sorted according to their scores for create cells. However, this proposed method has a similar problem as Zadrozny's binning method in that we can discover the best size of cell intervals only by experiments. Furthermore, because the number of samples for a cell may be different, the reliabilities of accuracies in cells are probably variant. To solve the problem, we use coverage for each cell as weight.

STEP 2. The original accuracy table generated above does not yield reliable probabilities (accuracies), when there are no or very few samples for some cells. Therefore, we propose smoothing on the original accuracy table. There are simple smoothing methods such as Laplace's law (Laplace) and Lidstone's law (Lidstone) [6]. In this paper, we denote, for an observed cell $c(f)$ in which f is the classification score, the number of training data samples that appear in the cell by $N(c(f))$ and denote the number of correctly classified samples within all the samples in the cell by $N_p(c(f))$. The smoothed accuracy $P_{Lap}(f)$ is formulated as $P_{Lap}(f) = (N_p(c(f)) + 1)/(N(c(f)) + 2)$, The smoothed accuracy $P_{Lid}(f)$ is formulated as $P_{Lid}(f) = (N_p(c(f)) + \delta)/(N(c(f)) + 2\delta)$, where δ specifies the added pseudo-counts. The value of δ for Lidstone was determined for each accuracy table, using cross-validation within the training data.

In both Laplace and Lidstone, accuracies are smoothed using solely the samples in the cell in question. However, further examination of the entire accuracy table shows that nearby cells fairly often have similar accuracies. Therefore, using the accuracies of cells near the target cell should be effective. We apply some smoothing methods such as a moving average method (MA) and a median method (Median) [1], which use values near the smoothing value target. The MA and Median are computed according to the following formula:

$$P_{MA}(f) = \frac{\frac{N_p(c(f))}{N(c(f))} + \sum_{s \in Nb(c(f))} \frac{N_p(s)}{N(s)}}{n}, \tag{1}$$

$$P_{Median}(f) = median_{s \in Nb(c(f))} \left(\frac{N_p(c(f))}{N(c(f))}, \{ \frac{N_p(s)}{N(s)} \} \right), \tag{2}$$

where $Nb(c(f))$ is the set of cells that are adjacent to cell $c(f)$ whose accuracy can be defined (i.e., there is at least one sample), and n gives $|Nb(c(f))| + 1$. Furthermore, we propose an extended MA, the moving average with coverage method (MA_cov), in which cells with many samples are more weighted in accuracy computation because the accuracy of those cells are more reliable.

$$P_{MA\_cov}(f) = \frac{\frac{N_p(c(f))}{N(c(f))} C(c(f)) + \sum_{s \in Nb(c(f))} \frac{N_p(s)}{N(s)} C(s)}{C(c(f)) + \sum_{s \in Nb(c(f))} C(s)}, \tag{3}$$

where $C(c(f))$ is the number of the samples in the cell $c(f)$ divided by the number of all the samples. In this paper, we simply use the cells directly neighboring the target cell as surrounding cells. For example, in the case of using the first-ranked score and the second-ranked score, we use nine cells; the up cell, down cell, left cell, right cell, the diagonal cells and the target cell.

STEP 3. We first calculate the classification scores of the sample, then find the range or cell that corresponds to the scores, and output the values associated in the range or cell in the accuracy table.

4 Experiments

4.1 Experimental Settings

Classifier. We used SVMs, and also used the Naive Bayes classifier for experiments to show the generality of the proposed method. The reason why we selected SVMs is that SVMs are widely applied to many applications in document classification [4]. Since SVMs are a binary classifier, we used the one-versus-rest method [7] to extend SVMs to a multiple classifier [5]. Following Takahashi et al. [13], we set the SVMs kernel function to be a linear kernel.

DataSet. We used two different datasets: the JGSS dataset, which is Japanese survey data, and UseNet news articles (20 Newsgroups), which were also used in Zadrozny et al.'s experiments [17] [6]. First, we used the JGSS dataset (23,838 samples) taken from respondents who had occupations [13]. Each instance of the respondents' occupation data consists of four answers: "job task" (open-ended), and "industry" (open-ended), both of which consisted of much shorter texts than usual documents, and have approximately five words in each, and "employment status" (close-ended), and "job title" (close-ended). We used these features for learning. The number of categories was nearly 200 and by past occupation coding, each instance was encoded into a single integer value called an occupational code. We used JGSS-2000, JGSS-2001, JGSS-2002 (20,066 samples in total) for training, and JGSS-2003 (3,772 samples) for testing. The reason why we did not use cross-validation is that we would like to imitate the actual coding process; we can use the data of the past surveys, but not of future surveys. To generate an accuracy table, we used a 5-fold cross-validation within the training data; we split 20,066 samples into five subsets with equal size, used four of them for temporary training and the rest for outputting the pairs of the scores and the status, and repeated five times with different combinations of four subsets. We used all the pairs to make an accuracy table. The second dataset, (the 20 Newsgroups dataset), consists of 18,828 articles after duplicate articles are removed. The number of categories is 20, corresponding to different UseNet discussion groups [9]. We employed a 5-fold cross-validation.

[5] `http://chasen.org/~taku/software/TinySVM/`

[6] The accuracies were 74.5% (JGSS dataset) and 87.3% (20 Newsgroups dataset).

Table 1. Relationships of cell intervals and the number of cells in SVMs

cell intervals	0.05	0.1	0.2	0.3	0.5
the number of cells (the first-ranked score used)	60	30	16	12	7

Table 2. Negative Log-Likelihood in the best case in each method for creating cells. A boldface number indicates the best log-likelihood of the two methods.

classifier	SVM	SVM	Naive Bayes
dataset	JGSS dataset	20 Newsgroups dataset	20 Newsgroups dataset
equal intervals	**2369.3**(# cells=30)	**1472.3**(# cells=30)	1679.8(# cells=16)
equal samples	2678.3(# cells=12)	1572.9(# cells=7)	**1671.0**(# cells=12)

Cell Intervals. We experimentally determined the best cell intervals. For these experiments, we created some accuracy tables with different cell intervals: 0.05, 0.1, 0.2, 0.3, and 0.5 etc. For example, Table 1 shows the relationships of cell intervals and the number of cells in the case of the first-ranked scores used.

Evaluation Metrics. We used the log-likelihood of test data to evaluate each method in Experiment 1. Larger values of log-likelihood are considered to be better. For simplicity, we use the negative log-likelihood [7]. As an evaluation method in Experiment 2, we used a reliability diagram, a ROC (receiver operating characteristic) curve, reliability for each coverage, accuracy for each threshold, and the ability to detect misclassified samples.

4.2 Experiment 1: Comparison of the Methods

The Proposed Method for Creating Cells. Before Experiment 1, we conducted simple experiments to confirm the effectiveness of the proposed method for making cells with equal cell intervals by comparing the proposed method with the method with equal samples for each cell. We used the values without smoothing and used only the first-ranked scores. Table 2 shows the results in the best cases by changing the number of cells from 7 to 60. The tendencies in other cases with the different number of cells were much the same as in Table 2. Thus, we confirmed the effectiveness of the proposed method for creating cells [8].

Evaluations by Log-Likelihood. Tables 3 and 4 show the negative log-likelihood of the JGSS dataset and the 20 Newsgroups dataset by the proposed methods as well as other methods for different numbers of used scores and

[7] The negative log-likelihood L is given by $L = -\sum_i log(p(x_i))$ where $p(x_i)$ is a predicted class membership probability of an evaluation sample.

[8] Zadrozny's number of bins ($bin = 10$) [15] was similar to that of Table 2.

Table 3. Negative Log-Likelihood (the JGSS dataset) 3,772 samples. MA_cov represents the Moving Average with coverage method. A boldface number indicates the best log-likelihood of all cases. In the case of rank1&rank2&rank3, the values were much better than rank1 but slightly worse than rank1&rank2 in each cell interval for each method except for the Sigmoid method (2232.9).

cell intervals	used scores	no smoothing	Laplace's Law	Lidstone's Law	Moving Average	Median	MA_cov	Sigmoid
0.1	rank1	2369.3	2368.9	2368.9	2367.5	2372.6	2364.7	2367.6
	rank1&rank2	-	2356.8	2355.8	2245.8	-	**2232.7**	2246.9
0.2	rank1	2371.3	2371.0	2370.3	2369.3	2370.0	2369.3	2367.6
	rank1&rank2	-	2252.7	2254.7	2240.6	2241.8	2235.0	2246.9
0.5	rank1	2381.9	2381.8	2381.6	2395.9	2396.4	2409.9	2367.6
	rank1&rank2	2265.8	2265.6	2265.7	2327.5	2298.8	2320.6	2246.9

different intervals on SVMs, respectively [9]. The Lidstone column shows the result when the predicted optimal value of δ is used. The dash (-) indicates that we cannot compute log-likelihood for those cases because the argument of the log function in some cells was 0. We discuss the results in Tables 3 and 4. First, for the SVMs, the best case for each dataset was the method using both the first-ranked score and the second-ranked score, in which we applied the moving average with coverage method (cell intervals=0.1). Second, for each method we obtained better log-likelihood scores when we used multiple scores than when we used a single score. In particular, using both the first-ranked score and the second-ranked score was the best for both datasets. The reason is that in multi-class classification, the probability of the first-ranked class depends not only upon the first-ranked scores, but also upon the second-ranked scores as mentioned in Section 3. Third, in the case of using multiple scores with smaller cell intervals (e.g. 0.1), smoothing methods such as MA or MA_cov, which use the accuracies of cells near the target cell, were more effective than the other methods.

To show the generality of the above conclusions, we conducted experiments of the same kind as in the above-mentioned experiments, except for Lidstone, using the Naive Bayes classifier for the 20 Newsgroups dataset. In Table 5, the dash (-) indicates the same meaning as in Tables 3 and 4. We obtained the same results as shown in Tables 3 and 4. First, the best of all cases was the method using both the first-ranked score and the second-ranked score, in which we smoothed by the Moving Average method with larger number of cells (e.g. 30). Second, for each method we obtained a better log-likelihood when we used multiple scores than when we used a single score. Third, in the case of using multiple scores with a larger number of cells (e.g. 30), smoothing methods such as MA or MA_cov were effective.

[9] The negative log-likelihood by the simple transforming formula mentioned in Section 1 was 5493.2 (Niculescu-Mizil et al.), 3142.7 (Zadrozny et al.) in the JGSS dataset, (-) (Niculescu-Mizil et al.) and 3463.6 (Zadrozny et al.) in the 20 Newsgroups dataset.

Table 4. Negative Log-Likelihood (the 20 Newsgroups dataset) 3,765 samples for each fold. MA_cov represents the Moving Average with coverage method. A boldface number indicates the best log-likelihood score of all cases. In the case of rank1&rank2&rank3, the values were much better than rank1 but slightly worse than rank1&rank2 in each cell interval for each method except for the Sigmoid method (1377.5).

cell intervals	used scores	no smoothing	Laplace's Law	Lidstone's Law	Moving Average	Median	MA_ cov	Sigmoid
0.1	rank1	1472.3	1472.4	1472.2	1468.1	1469.6	1467.4	1482.3
	rank1&rank2	-	1390.2	1388.3	1362.3	-	**1360.3**	1386.6
0.2	rank1	1472.5	1472.7	1472.5	1474.4	1473.3	1482.7	1482.3
	rank1&rank2	-	1365.4	1366.9	1374.9	-	1377.7	1386.6
0.5	rank1	1487.4	1487.5	1487.4	1503.9	1497.0	1537.9	1482.3
	rank1&rank2	1388.1	1387.7	1387.8	1447.2	1408.7	1479.4	1386.6

Table 5. Negative Log-Likelihood (the 20 Newsgroups dataset) 3,765 samples for each fold. MA_cov represents the Moving Average with coverage method. A boldface number indicates the best log-likelihood of all cases.

# cells	used scores	no smoothing	Laplace's Law	Moving Average	Median	MA_ cov
30	rank1	-	1680.6	1670.1	1668.4	1675.0
	rank1&rank2	-	1439.7	**1409.8**	-	1415.3
16	rank1	1680.2	1679.8	1679.6	1675.8	1696.2
	rank1&rank2	-	1428.1	1515.5	-	1536.2
7	rank1	1697.2	1697.2	1712.0	1713.5	1732.8
	rank1&rank2	-	1474.8	1626.3	1644.8	1664.1

Finally, we obtained results of the sigmoid method using multiple scores as shown in the right column in Tables 3 and 4. For expansion of the sigmoid function, we used the formula: $P_{Log}(f_1, \cdots, f_r) = 1/(1 + \exp(\sum_{i=1}^{r} A_i f_i + B))$, where f_i represents the ith-ranked classification score. The parameters A_i $(\forall i)$ and B are estimated with the maximum likelihood method. In the sigmoid method, we also showed that the values of log-likelihood were better when we used multiple scores than when we used a single score. The sigmoid method showed an average performance in the methods in the two tables in each dataset.

4.3 Experiment 2: Evaluation of the Proposed Method

Reliability Diagram and ROC curve. We used the reliability diagram and the ROC curve to evaluate the proposed method. To plot a reliability diagram, we used an average of the estimates of the samples in each interval (e.g. $[0, 0.1]$) as a predicted value (x) and the average of actual values corresponding to the estimates as a true value (y). Figure 1 shows three reliability diagrams in the JGSS dataset. In the proposed method, all points were near the diagonal line. In the reliability diagram, the farthest a point is from the diagonal line, the

worse the performance. As a whole, the proposed method was better than both a method without smoothing and the sigmoid method. This tendency was the same as in the 20 Newsgroups dataset. Figure 1 also shows three ROC curves in the 20 Newsgroups dataset. On a ROC curve, the nearer a ROC curve is to upper left line, the better a method is. The proposed method was the best of the three methods, and this tendency was the same as in the JGSS dataset.

We also investigated the predicted values by the proposed method and the actual values by increasing the coverage every 10% from 10% to 100%. Although there is a limitation such that both the predicted value and the actual value are not the values of the sample itself, the predicted values are nearly the same as the actual values in descending order and ascending order in both datasets.

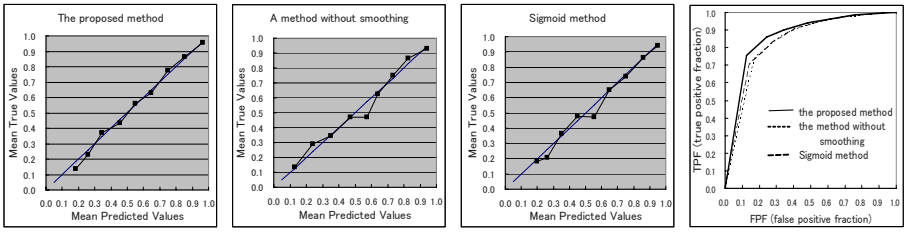


Fig. 1. The Reliability Diagrams in the JGSS dataset and the ROC curves in the 20 Newsgroups dataset (right) on SVMs

Accuracy for each Threshold. If we could select samples accurately enough either to process or not to process using our own threshold, the work done by humans would be lighter. We investigated accuracies of the proposed method by increasing the threshold of estimates every 0.1 from 0 to 0.9. The proposed method always outperformed both a method without smoothing and sigmoid methods in both datasets. For example, when the threshold was set to 0.9, accuracies were approximately 96% in the JGSS dataset and 96% in the 20 Newsgroups dataset. As for coverage, the proposed method scored then second and first in the JGSS dataset and the 20 Newsgroups dataset, respectively.

Ability to Detect Misclassified Samples. We ordered all the test instances by ascending order of the estimated class membership probability and counted the number of error samples in the set of samples with low probability. We compared our method with *the raw score method* [12], in which the distance from the separation hyperplane is directly used instead of the probability. We evaluated these methods by the ratio of the detected errors. Figure 2 shows the number of error samples detected by the proposed method and those by the raw score method in both datasets. The proposed method always surpassed the raw score method in each dataset. In the 20 Newsgroups dataset especially, the proposed method performed better when coverage was lower, which is desirable for us, since, in practice, we would like to find many errors by manually checking only a small amount of data. The reason for the difference of the two methods

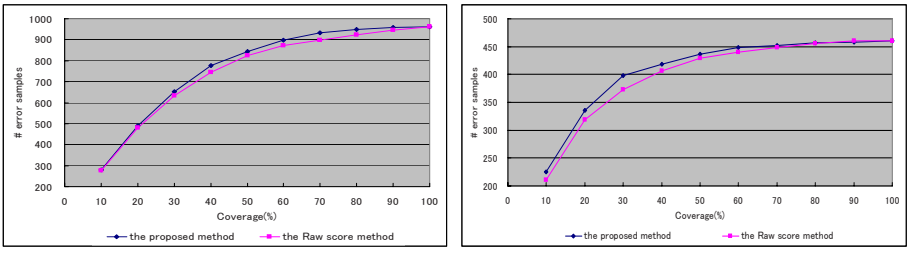


Fig. 2. Ability to Detect Misclassified Samples on SVM (the JGSS dataset (left) and the 20 Newsgroups dataset (right))

is not clear in the JGSS dataset. A possible explanation would be that the JGSS dataset has many very short samples, among which there are only a few active features. Those short samples do not have enough information for precise probability estimation.

5 Conclusion

In this paper, to estimate class membership probabilities, we proposed using multiple scores outputted by classifiers, and generating an accuracy table with smoothing methods such as the moving average method or the moving average with coverage method. Through the experiments on two different datasets with both SVMs and Naive Bayes classifiers, we empirically showed that the use of multiple classification scores was effective in the estimation of class membership probabilities, and that proposed smoothing methods for the accuracy table worked quite well. Further research will be necessary to discover effective cell intervals. The use of information criteria such as AIC (Akaike Information Criteria) [11] is the next research area we intend to pursue.

Acknowledgements. This research was partially supported by MEXT Grant-in-Aid for Scientific Research (c)6530341.

References

1. T. Agui and M. Nakajima. *Graphical Information Processing.* Morikita Press, Tokyo, 1991.
2. P. N. Bennett. Assessing the Calibration of Naive Bayes's Posterior Estimates. Technical Report CMU-CS-00-155, pages 1-8. School of Computer Science, Carnegie Mellon University, 2000.
3. Y.S. Chan and H.T.Ng. Estimating Class Priors in Domain Adaptaion for Word Sense Disambiguation. In *Proceedings of 21st International Conference on Computaional Linguistic and 44th Annual Meeting of the ACL*, pages 89-96, 2006.

4. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of 10th European Conference on Machine Learning*, pages 137-142, 1998.
5. R. Jones, B. Rey, O. Madani, and W. Griner. Generating Query Substitutions. In *Proceedings of International World Wide Web Conference*, pages 387-396, 2006.
6. K. Kita. *Laguage and Computing Volume 4: Probabilistic Language Model*. University of Tokyo Press, Tokyo, 1999.
7. U. Kressel. Pairwise classification and support vector machines. In B. Schölkopf et al. (Eds.) *Advances in Kernel Methods Support Vector Learning*, pages 255-268. The MIT Press, 1999.
8. A. Niculescu-Mizil and R.Caruana. Predicting Good Probabilities With Supervised Learning. In *Proceedings of 22nd International Conference on Machine Learning*, pages 625-632, 2005.
9. K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103-134, 2000.
10. J. C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In A. J. Smola et al. (Eds.) *Advances in Large Margin Classifiers*, pages 1-11. MIT Press, 1999.
11. Y. Sakamoto, M. Ishiguro, and G. Kitagawa. *Akaike Information Criterion Statistics*. Kyoritsu Press, Tokyo, 1983.
12. G. Schohn and D. Cohn. Less is More: Active Learning with Support Vector Machines. In *Proceedings of 17th International Conference on Machine Learning*, pages 839-846, 2000.
13. K. Takahashi, H. Takamura, and M. Okumura. Automatic Occupation Coding with Combination of Machine Learning and Hand-Crafted Rules. In *Proceedings of 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 269-279, 2005.
14. K. Takahashi, A. Suyama, N. Murayama, H. Takamura, and M. Okumura. Applying Occupation Coding Supporting System for Coders (NANACO) in JGSS-2003. *Japanese Value and Behavioral Pattern Seen in JGSS in 2003*, pages 225-242. the IRS at Osaka University of Commerce, 2005.
15. B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive Bayeian classifiers. In *Proceedings of 7th International Conference on Knowledge Discovery and Data Mining*, pages 609-616, 2001.
16. B. Zadrozny and C. Elkan. Learning and Making Decisions When Costs and Probabilities are Both Unknown. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 204-213, 2001.
17. B. Zadrozny and C. Elkan. Transforming Classifier Scores into Accurate Multiclass Probability Estimates. In *Proceedings of 8th International Conference on Knowledge Discovery and Data Mining*, pages 694-699, 2002.

A Hybrid Multi-group Privacy-Preserving Approach for Building Decision Trees*

Zhouxuan Teng and Wenliang Du

Department of Electrical Engineering and Computer Science
Syracuse University, Syracuse, NY 13244, USA
zhteng@syr.edu, wedu@ecs.syr.edu

Abstract. In this paper, we study the privacy-preserving decision tree building problem on vertically partitioned data. We made two contributions. First, we propose a novel *hybrid* approach, which takes advantage of the strength of the two existing approaches, randomization and the secure multi-party computation (SMC), to balance the accuracy and efficiency constraints. Compared to these two existing approaches, our proposed approach can achieve much better accuracy than randomization approach and much reduced computation cost than SMC approach.

We also propose a multi-group scheme that makes it flexible for data miners to control the balance between data mining accuracy and privacy. We partition attributes into groups, and develop a scheme to conduct group-based randomization to achieve better data mining accuracy. We have implemented and evaluated the proposed schemes for the ID3 decision tree algorithm.

Keywords: Privacy, SMC, Randomization.

1 Introduction

In today's information age, both the volume and complexity of data available for decision-making, trend analysis and other uses continue to increase. To "mine" these vast datasets for useful purposes has spurred the development of a variety of data mining techniques. Of considerable interest is abstracting information from a dataset composed of information which may be located at different sites, or owned by different people or agencies, i.e., distributed databases. However, data owners must be willing to share all their data. Issues of privacy and confidentiality can arise which prohibit data owners from contributing to a data warehouse. To address these critical privacy and confidentiality issues, privacy-preserving data mining (PPDM) techniques have emerged.

In this paper, we study a specific PPDM problem: building decision trees on vertically partitioned data sets. In this PPDM problem, the original data set

* This work was supported by Grant ISS-0219560, ISS-0312366 and CNS-0430252 from the United States National Science Foundation.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 296–307, 2007.

D is vertically divided into two parts, with one part D_a known by Alice, and the other part D_b known by Bob. The problem is to find out how Alice and Bob conduct data mining on the vertically joint data set $D = D_a \cup D_b$, without compromising their private information.

A number of solutions have been proposed in the literature to solve various privacy-preserving data mining problems. They can be classified into two general categories: the *secure multi-party computation* (SMC) and the *randomization* approaches. In the SMC approach, Alice and Bob run a cryptographic protocol to conduct the joint computation. SMC can conduct the required computation while ensuring that the private inputs from either party are protected from each other. Previous results using the SMC approach include [3,6,8]. In the randomization approach, one of the parties (e.g. Alice) adds some noise to her data to disguise the original data D_a, and then she sends the disguised data set $\widehat{D_a}$ to Bob; Several schemes have been proposed for conducting data mining based on the partially disguised joint data formed by $\widehat{D_a}$ and D_b, including [2,1,5,4][1].

The contribution of this paper is two-fold: First, we have developed a hybrid scheme that can harness the strength of both SMC and randomization schemes to achieve a better accuracy and efficiency. Second, we have developed a general multi-group scheme, which provides a flexible mechanism for data miner to adjust the balance between privacy and accuracy.

Our proposed hybrid approach and multi-group approach are general and can be applied to various data mining computations, including decision tree building and association rule mining. In this paper, they are applied to the ID3 decision tree algorithm[2].

2 Problem Definition and Background

In this paper we focus on a specific decision tree building problem for vertically partitioned data. The problem is illustrated in Figure 1(a).

Definition 1. *(Two-party decision tree building over vertically partitioned data) Two parties, Bob and Alice, each have values of different attributes of a data set. They want to build a decision tree based on the joint database. However neither of them wants to disclose the accurate values of the attribute he/she is holding to other party, i.e., nobody can actually have the "joint" database.*

2.1 ID3 Algorithm

In a decision tree, each non-leaf node contains a splitting point, and the main task for building a decision tree is to identify the test attribute for each splitting point. The ID3 algorithm uses the information gain to select the test attribute.

[1] Some of these studies are not targeted at the vertically partitioned data, they can nevertheless be trivially extended to deal with this kind of data partition scenario.

[2] Our scheme can also be applied to other decision tree algorithms.

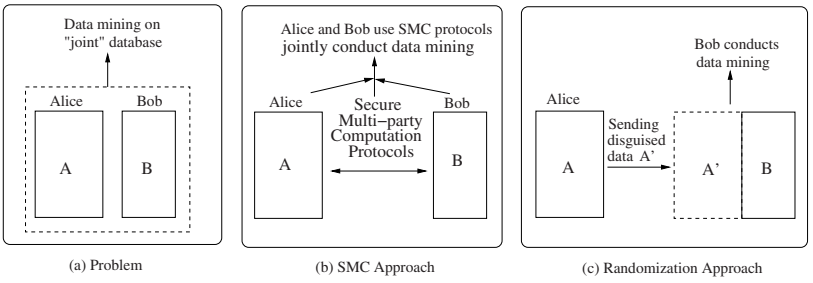


Fig. 1. Problem and different approaches

Information gain can be computed using *entropy*. In the following, we assume there are m classes in the whole training data set. We know

$$Entropy(S) = -\sum_{j=1}^{m} Q_j(S) \log Q_j(S), \tag{1}$$

where $Q_j(S)$ is the relative frequency of class j in S. We can compute the information gain for any candidate attribute A being used to partition S:

$$Gain(S, A) = Entropy(S) - \sum_{v \in A} (\frac{|S_v|}{|S|} Entropy(S_v)), \tag{2}$$

where v represents any possible values of attribute A; S_v is the subset of S for which attribute A has value v; $|S|$ is the number of elements in S.

In decision tree building, assume that the set S is associated with a node V in the tree. All the records in S has the same values for certain attributes (each corresponds to a node from the root to V). We use an logical `AND` expression $E(S)$ to encode those attributes, namely all the records in S satisfy the expression $E(S)$. Let D represent the entire data set. We use $N(E)$ to represent the number of records in the data set D that satisfies the expression E. Then,

$$\begin{aligned} |S| &= N(E(S)) \\ |S_v| &= N(E(S_v)) \\ &= N(E(S) \wedge (A = v)) \\ Q_j(S) &= \frac{N(E(S) \wedge (Class = j))}{N(E(S))}. \end{aligned}$$

From the above equations, we know that as long as we can compute $N(E)$ for any logical `AND` expression E, we can get all the elements that allow us to compute entropies and information gains. We show how to compute $N(E)$ using the SMC approach or the randomization approach for vertically-partitioned data.

The SMC Approach. The SMC approach is depicted in Figure 1(b). Let us divide E into two parts, $E = E_a \wedge E_b$, where E_a contains only the attributes

from Alice, while E_b contains only the attributes from Bob. Let V_a be a vector of size n: $V_a(i) = 1$ if the ith record satisfies E_a; $V_a(i) = 0$ otherwise. Because E_a belongs to Alice, Alice can compute V_a from her own share of attributes. Similarly, let V_b be a vector of size n: $V_b(i) = 1$ if the ith data item satisfies E_b; $V_b(i) = 0$ otherwise. Bob can compute V_b from his own share of attributes.

Note that a nonzero entry of $V = V_a \wedge V_b$ (i.e. $V(i) = V_a(i) \wedge V_b(i)$ for $i = 1, \ldots, n$) means the corresponding record satisfies both E_a and E_b, thus satisfying E. To compute $N(E)$, we just need to find out how many entries in V are non-zero. This is equivalent to computing the dot product of V_a and V_b:

$$N(E) = N(E_a \wedge E_b) = V_a \cdot V_b = \sum_{i=1}^{n} V_a(i) * V_b(i).$$

A number of dot-product protocols have already been proposed in the literature [6,3]. With these SMC protocols, Alice and Bob can get (and only get) the result of $N(E)$, neither of them knows anything about the other party's private inputs, except the information that can be derived from $N(E)$.

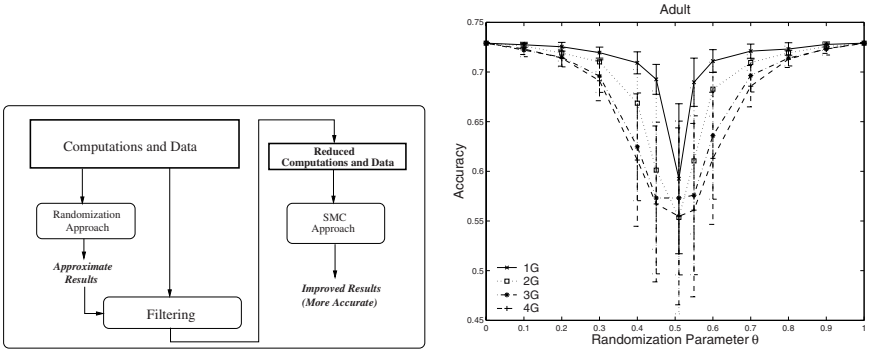
The Randomization Approach. To use the randomization approach to build decision trees, Alice generates a disguised data set $\widehat{D_a}$ from her private data D_a. Alice then sends $\widehat{D_a}$ to Bob. Bob now has the full data set $\widehat{D_a} \cup D_b$, though part of which is disguised. Bob can conduct data mining based on this partially disguised data set. This approach is depicted in Figure 1(c).

There are a number of ways to perform randomization. Our scheme in this paper is based on the randomized response technique [7]. They were proposed in several existing work [5,4] to deal with categorical data in privacy-preserving data mining. Readers can get details from the literature and we do not describe them in detail here due to page limitations.

3 A Hybrid Approach for Privacy-Preserving Data Mining

Many data mining computations involve *searching* among a set of candidates. For example, in building decision trees, at each tree node, we search for the best test attribute from a candidate set based on certain criteria; in association rule mining, we search through a set of candidates to find those whose supports are above certain threshold. Using SMC to conduct these searches is expensive since the search space can be quite large. If we can reduce the search space using some light-weight computations (in terms of both communication and computation costs), we can significantly reduce the total costs.

Randomization scheme is a very good choice for such a light-weight computation because of two reasons: it is much less expensive than SMC, and yet it produces results good enough for filtering purposes. If Z_u is a significant portion of Z, the costs of SMC is substantially reduced compared to the computations that use SMC alone. The entire hybrid approach is depicted in Figure 2(a).



(a) General Hybrid Approach

(b) Adult: Accuracy vs. Window Size

Fig. 2. General Hybrid Approach and Experiment Results for Adult

In the next section, we describe a *Hybrid-ID3* algorithm that uses randomization to get some candidate splitting attributes at each node and then use SMC method to choose the best one from these candidates.

3.1 The *Hybrid-ID3* Algorithm

Let D represent the entire data set. Let D_a represent the part of the data owned by Alice, and let D_b represent the part of the data owned by Bob. Alice disguises D_a using the randomization approach, and generate the disguised data set $\widehat{D_a}$; she sends $\widehat{D_a}$ to Bob. Bob does the same and sends his disguised part $\widehat{D_b}$ to Alice. Alice forms the entire data set $D_1 = D_a \cup \widehat{D_b}$, while Bob forms $D_2 = \widehat{D_a} \cup D_b$.

We describe the *Hybrid-ID3* algorithm which uses the randomization and SMC schemes as building blocks. In this algorithm, we use $N(E)$ to represent the *actual* number of records in D that satisfy the expression E (computed using the SMC approach.) We use AL to represent a set of candidate attributes. Before conducting this algorithm, Alice and Bob have already exchanged the disguised data. Namely Alice has $D_1 = D_a \cup \widehat{D_b}$, and Bob has $D_2 = \widehat{D_a} \cup D_b$.

Hybrid-ID3(E, AL)

1. Create a node V.
2. **If** $N(E \wedge (class = C)) == N(E)$ for any class C, **then** return V as a leaf node labeled with class C. Namely, all the records that satisfy E belong to class C.
3. **If** AL is empty, **then** return V as a leaf-node with the class $C = \text{argmax}_C$ $N(E \wedge (class = C))$. Namely, C is the majority class among the records that satisfy E.
4. Find the splitting attribute using the following procedure:
   (a) For each test attribute $A \in AL$, Alice computes (estimates) A's information gain from D_1, and Bob computes A's information gain from D_2,

both using the randomization approach. Alice and Bob use the average of their results as A's estimated information gain.
   (b) Select ω test attributes that have the ω highest information gains.
   (c) Using SMC to compute the actual information gains for these ω attributes, and select the one TA with the highest information gain.
5. Label node V with TA.
6. **For** each known value a_i of TA
   (a) Grow a branch from node V for the condition $TA = a_i$.
   (b) **If** $N(E \wedge (TA = a_i)) == 0$ **then** attach a leaf labeled with $C = \text{argmax}_C N(E \wedge (class = C))$, i.e., C is the majority class among the records that satisfy E.
   (c) **Else** attach the node returned by **Hybrid-ID3**$(E \wedge (TA = a_i)$**,** $AL - TA$**)**.

Note that the values of $N(E \wedge (class = C))$ at Step 2 and Step 3 can be obtained from Step 4.c of the previous round. Similarly, computations at Step 6.b can also be obtained from Step 4.c of the same round. Therefore, there are no extra SMC computations in Step 2, 3, and 6.b.

3.2 Privacy and Cost Analysis

Because SMC computations do not reveal any more information about the private inputs than what can be derived from the results, the primary source of information disclosure is from the disguised data due to the randomization approach. Several privacy analysis methods for the randomization approach have been proposed in the literature [1,5]. We will not repeat them in this paper.

Regarding the computation and communication costs, we are only interested in the relative costs compared to the SMC-only approach. Since the computation and the communication costs of the randomization part is negligible compared to the SMC part, we use the amount of SMC computations conducted in the hybrid approach as the measure of the cost, and we compare this cost with the amount of SMC computations conducted in the SMC-only approach. This cost ratio between these two approaches is primarily decided by the window size. We will give the simulation results in section 5.

4 The Multi-group Randomization Scheme

For many data mining computations, calculating the accurate relationship among attributes is important. Randomization tends to make this calculation less accurate, especially when each attribute is randomized independently, because of the bias introduced by the randomization schemes. The randomization schemes proposed in the literature mostly randomize attributes independently. We have found out that such randomization schemes lead to undesirable results for privacy-preserving decision tree building. To achieve better accuracy, we propose a general *multi-group* framework, which can be used for randomization schemes.

In this scheme, attributes are divided into g ($1 \leq g \leq t$) groups (where t is the total number of attributes in the data set); randomization is applied on the unit of groups, rather than on the unit of single attribute. For example, if randomization is to add random noise, then we will add the same noise to the attributes within each group[3]. However, these numbers are independent from group to group. The advantage of this multi-group scheme is that by adding the same random noise to hide several attributes together, the relationship of these attributes are better preserved than if independent random numbers are added. However, the disadvantage of this approach is that if adversaries know the information about one attribute, they can find the information about the other attributes in the same group. Thus, there is a balance between privacy and data mining accuracy. By choosing the appropriate value of g, we can achieve a balance that is suitable for a specific application.

To demonstrate the effectiveness of this multi-group framework, we apply it to a specific randomization scheme, the randomized response scheme, which has been used by various researchers to achieve privacy-preserving data mining. We call our scheme the *Multi-group Randomized Response (MRR)* scheme. The existing randomized response schemes are special case of the MRR scheme: the scheme proposed in [4] is a 1-group scheme, while the schemes proposed in [5] are essentially t-group scheme because each attribute forms its own group.

Data Disguise. In the general randomized response technique, before sending a record to another party (or to the server), a user flips a biased coin for each attribute independently, and decides whether to tell a truth or a lie about the attribute based on the coin-flipping result. In MRR scheme, the process is still the same, the only difference is that now the coin-flipping is conducted for each group, and a user either tells a truth for all the attributes in the same group or tells a lie about all of them.

Estimating $N(E)$. Let $P(E)$ represent the portion of the data set that satisfies E. Estimating $N(E)$ is equivalent to estimating $P(E)$.

Assume that the expression E contains attributes from m groups. We rewrite E using the following expression, with e_k being an expression consisting of only attributes from the group k (we call e_k a sub-pattern of E):

$$E = e_1 \wedge e_2 \wedge \cdots \wedge e_m = \bigwedge_{k=1}^{m} e_k$$

We define a variation of E as $E' = f_1 \wedge \cdots \wedge f_m$, where f_i is equal to either e_i or the bitwise-opposite of e_i (i.e. $\overline{e_i}$). For each expression E, there are totally 2^m different variations, including E itself. We denote these variations of E as E_0 to E_ω, where $E_0 = E$ and $\omega = 2^m - 1$.

[3] If the domains of attributes are different, the range of the random numbers can be adjusted to match their domains.

Theorem 1. *Let $P(E_i \rightarrow E_j)$ represent the probability that an expression E_i in the original data becomes an expression E_j in the disguised data after the randomized response process. We have the following formula:*

$$P(E_i \rightarrow E_j) = \theta^u (1-\theta)^{m-u},$$

where u represents the number of the common bits between the binary forms of number i and number j.

Proof. Proof is omitted due to page limitations.

Let $P^*(E)$ represent the expected number of records, in the *disguised* data set, that satisfies the expression E. $P^*(E)$ can be estimated by counting the number of records that satisfy E in the *disguised* data set. Obviously, we have

$$P^*(E) = \sum_{i=0}^{\omega} P(E_i \rightarrow E_j) P(E_i)$$

If we define a matrix A, such that $A(i,j) = P(E_i \rightarrow E_j)$ for $i = 0, \cdots, \omega$ and $j = 0, \cdots, \omega$, we get the following linear system of equations.

$$\begin{pmatrix} P^*(E_0) \\ \vdots \\ P^*(E_\omega) \end{pmatrix} = A \begin{pmatrix} P(E_0) \\ \vdots \\ P(E_\omega) \end{pmatrix}$$

Theorem 2. *The matrix A defined as above is invertible if and only if $\theta \neq 0.5$.*

Proof. Proof by induction and the proof is omitted due to page limitations.

In situations where $P(E)$ is the only thing we need, just like in the ID3 decision tree building algorithm, there is a much more efficient solution with cost $O(m)$ instead of $O(2^m)$. This technique is similar to the one used in [5] and it is omitted here due to page limitations.

5 Evaluation

To evaluate the proposed hybrid scheme, we have selected three databases from the UCI Machine Learning Repository [4]: *Adult*, *Mushroom*, and *Tic-tac-toe* datasets. We randomly divide all attributes of each data set into two parts with the same cardinality: Alice and Bob's share respectively.

In our experiments, we always used 80% of the records as the training data and the other 20% as the testing data. We use the training data to build the decision trees, and then use the testing data to measure how accurate these trees can predict the class labels. The percentage of the correct predictions is the *accuracy* value in our figures. We repeat each experiment for multiple times, and each time the disguised data set is randomly generated from the same original data set. We plot the means and the standard deviation for the accuracy values. The results for *Tic-tac-toe* dataset is omitted due to page limitations.

[4] ftp://ftp.ics.uci.edu/pub/machine-learning-databases

5.1 Accuracy vs. Number of Groups

Figure 2(b) shows the change of accuracy along the number of groups in the *randomization-only* approach for *Adult* dataset. In the figure, "1G", "2G", "3G", and "4G" indicate that the data are disguised using the 1-group, 2-group, 3-group, and 4-group randomization schemes respectively. From the figure, we can see that the accuracy decreases when the number of groups increases. When θ is close to 0.5 (e.g., $\theta = 0.4$), the rate of deterioration is rapid as the number of group increases. It is interesting to see that the results of the 4-group scheme are very close to those of the 3-group scheme. This is because in this specific *Adult* dataset, most of the expressions that are evaluated in building the tree contain attributes from less than 3 groups.

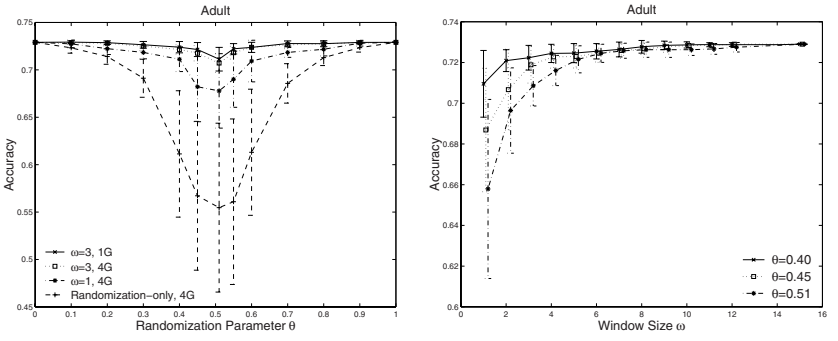
5.2 Accuracy: Hybrid vs. Randomization-Only

Figures 3(a) and 4(a) show the accuracy comparisons between the hybrid approach and the randomization-only approach. The vertical bars in the figures depict the standard deviations. The comparisons are shown for different randomization parameter θ and for different window size ω. In these three figures, "4G" and "1G" indicate that the data are disguised using the 4-group randomization scheme and the 1-group randomized scheme, respectively.

The figures clearly show that the hybrid approach achieves a significant improvement on accuracy compared to the randomization-only approach. When θ is near 0.5, the accuracy of the trees built via the randomization-only approach is just slightly better than the random guess (a random guess can yield 50% of accuracy on average). In contrast, the trees built via the hybrid approach can achieve a much better accuracy.

When the window size is increased to 3, the accuracy difference between the 4-group randomization scheme and the 1-group randomization scheme becomes much small. This means, choosing the 4-group randomization scheme does not degrade the accuracy much when $\omega = 3$, while at the same time, it achieves better privacy than the 1-group randomization scheme.

A surprising result in all these three figures is that when the window size is set to 1, the accuracy can be improved significantly compared to the randomization-only approach. Initially we thought that the hybrid approach with $\omega = 1$ is equivalent to the randomization-only approach. From this result, we realized that they are different, and the difference is at Step 2 and 6.b of the **Hybrid-ID3** algorithm. Step 2 detects whether all the records associated with the current tree node belong to a single class C. If so, we will not further split this node. With the hybrid approach, such a detection is conducted using SMC, which always generates the accurate results. However, using the randomization-only approach, because the result is inaccurate, it is very likely that we will continue splitting the node even when such a splitting is unnecessary. These extra splittings may result in a dramatic different tree structure compared to the tree built upon the original undisguised data, thus cause the significant difference in their accuracy results. Step 6.b has the similar effect.



(a) Hybrid vs. Randomization-Only

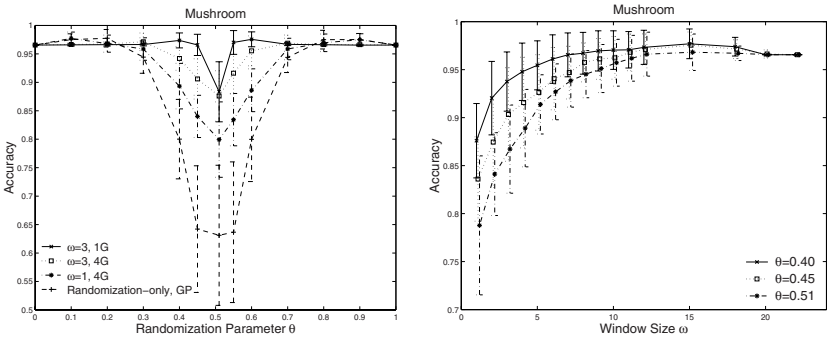
(b) Accuracy vs. Window Size

Fig. 3. Experiment Results for Adult Data Sets

5.3 Accuracy vs. Window Size ω

Figures 3(b) and 4(b) show the relationship between the accuracy and the window size in the hybrid approach where the number of groups g is 4.

The figures show that increasing SMC window size increases the accuracy of the decision tree. The increase is quite rapid when the window size is small; after certain point, the change of the window size does not affect the accuracy much. This means that the actual best test attribute is very likely among the top few candidates. This indicates that choosing a small window size can be the very cost-effective: it achieves a decent degree of accuracy without having to conduct many expensive SMC computations.



(a) Hybrid vs. Randomization-Only

(b) Accuracy vs. Window Size

Fig. 4. Experiment Results for Mushroom Data Sets

5.4 Efficiency Improvement

The motivation of the hybrid approach is to achieve better accuracy than the randomization-only approach, as well as achieve better efficiency than the SMC-only approach. Our previous experiments have shown the accuracy improvement. We now show how well the hybrid approach achieves the efficiency goal. We have summarized the efficiency improvement in Table 1, alone with the degree of accuracy achieved (4-group randomization and $\theta = 0.45$).

In Table 1, A is the accuracy of the hybrid approach minus the accuracy of the randomization-only approach, C is the ratio of the total number of SMC computations in the hybrid approach to that in the SMC-only approach.

The table shows that the efficiency improvement for the Mushroom data set is the most significant. This is because the number of attributes in the Mushroom data set is larger. This trend indicates that the larger the number of attributes, the higher level of efficiency improvement.

Table 1. Performance Improvement

	$\omega = 2$		$\omega = 3$		$\omega = 4$	
	A	C	A	C	A	C
Adult	0.14	19%	0.15	28%	0.16	37%
Mushroom	0.23	10%	0.26	15%	0.27	20%

6 Conclusions and Future Work

We have described a hybrid approach and a multi-group randomization approach for privacy-preserving decision tree buildings over vertically-partitioned data. The hybrid approach combines the strength of the SMC approach and the randomization approach to achieve both high accuracy and efficiency. Our experiments show that the hybrid approach achieves significantly better accuracy compared to the randomization-only approach and it is much more efficient than the SMC-only approach. Our multi-group randomization approach allows data miners to control the trade-off between privacy and data mining accuracy.

For the hybrid approach, we only used a fixed window size throughout the entire decision tree building process. In the future, we will investigate whether a dynamic window size can help further improve the performance, i.e., the window size for different tree nodes might be different, depending on the randomization results. We will also investigate the effectiveness of the hybrid approach on other data mining computations.

References

1. D. Agrawal and C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Santa Barbara, California, USA, May 21-23 2001.

2. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD on Management of Data*, pages 439–450, Dallas, TX USA, May 15 - 18 2000.
3. W. Du and Z. Zhan. Building decision tree classifier on private data. In *Workshop on Privacy, Security, and Data Mining at The 2002 IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, Japan, December 9 2002.
4. W. Du and Z. Zhan. Using randomized response techniques for privacy-preserving data mining. In *Proceedings of The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 505–510, Washington, DC, USA, August 24-27 2003.
5. A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In *Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 2002.
6. J. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, July 23-26 2002.
7. S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, March 1965.
8. Z. Yang, S. Zhong, and R.N. Wright. Anonymity-preserving data collection. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, USA., August 21-24 2005.

A Constrained Clustering Approach to Duplicate Detection Among Relational Data

Chao Wang, Jie Lu, and Guangquan Zhang

Faculty of Information Technology, University of Technology, Sydney
PO Box 123, Broadway, NSW 2007, Australia
{cwang, jielu, zhangg}@it.uts.edu.au

Abstract. This paper proposes an approach to detect duplicates among relational data. Traditional methods for record linkage or duplicate detection work on a set of records which have no explicit relations with each other. These records can be formatted into a single database table for processing. However, there are situations that records from different sources can not be flattened into one table and records within one source have certain (semantic) relations between them. The duplicate detection issue of these relational data records/instances can be dealt with by formatting them into several tables and applying traditional methods to each table. However, as the relations among the original data records are ignored, this approach generates poor or inconsistent results. This paper analyzes the characteristics of relational data and proposes a particular clustering approach to perform duplicate detection. This approach incorporates constraint rules derived from the characteristics of relational data and therefore yields better and more consistent results, which are revealed by our experiments.

1 Introduction

Data mining tasks usually work on large data warehouses where data often comes from multiple sources. The quality of mining results largely depends on the quality of data. One problem that degrades the data quality is the duplicated data records among the sources. Duplicate detection/elimination then is an essential preprocessing for data mining tasks and different methods have been proposed to deal with this problem [1,2,3,4]. The main idea of these methods is to use certain metrics to determine if certain pairs of data records are similar enough to be duplicates. In these methods, each data record is mostly of one same type and exists as an independent instance during the duplicate detecting process.

On the other hand, relational data is common in reality. Databases in complicated applications often have multiple tables to store multi-type records with relations. Semi-structured data over the Web also has the relational characteristic in terms of the referencing via hyperlinks. The requirement of duplicate detection among relational data is then obvious. Traditional methods can still work but without acknowledging the characteristics of relational data, they tend to produce inadequate and even inconsistent results. Recently, several models [5]

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 308–319, 2007.

[6] have been proposed to address this issue. These models are built on probability theories. They capture the relational features between records to collectively de-duplicate them with more accuracy. To make the models work, labeled samples should be supplied for estimating model parameters and this training process often takes a considerable amount of time due to the complexity of the model.

This paper then proposes an efficient approach to detect duplicates among relational data. The characteristics of relational data are analyzed from the perspective of duplicate detection. We define constraint rules that capture these characteristics. Our approach then incorporates these constraint rules into a typical canopy clustering process for duplicate detection. Experiments show that our approach performs well with improved accuracy. Furthermore, as our approach is based on clustering, no labeled samples are essentially required and no extra training process is involved, which sometimes is good for large and raw data sets.

The rest of the paper is organized as follows. Section 2 discusses related work. Situation of relational data and its characteristics are discussed in Section 3. Section 4 defines constraint rules for duplicate detection in relational data. Section 5 presents the constrained clustering approach. Experiments and evaluation results are shown in Section 6. Section 7 concludes the paper and discusses the future works.

2 Related Work

Duplicate detection of data was initially studied in database community as "record linkage" [7]. The problem was formalized with a model in [1] and was further extended in [2]. This model computes over features between pairs of records and generates similarity scores for them. Those pairs with scores above a given threshold are treated as duplicates and transitive closure is performed over them to yield the final result. In [8], clustering-based methods are proposed to identify duplicates in publication references. Their approach performs quick canopy clustering with two thresholds at the first stage and perform more expensive clustering methods within each canopy cluster at the second stage for refined results. The records for de-duplication in these methods are not relational, which means each record is a separate instance with no explicit relation with another.

Supervised learning methods have also been employed to make duplicate detection more adaptive with given data. Cohen et al [3] propose an adaptive clustering method and introduces the notion of pairing function that can be learned to check duplicates. Tejada et al [9] use a mapping-rule learner consisting of a committee of decision tree classifiers and a transformation weight learner to help create mapping between records from different data sources. Bilenko and Mooney [4] use a stochastic model and Support Vector Machine (SVM) [10] to learn string similarity measures from samples so that accuracy can be improved for the given situation. These methods are more adaptive and accurate because of their various learning processes which require an adequate amount of

labeled data. Again, all of these methods work on traditional data records with no relational features.

Recently, the relations between data records have been noticed in duplicate detection research community. Singla and Domingos [5] build a collective model that relies on Conditional Random Fields (CRFs) [11] to capture the relation of records for de-duplication. The relationship is indicated just by common field values of data records. The model proposed by Culotta and McCallum [6], which is based on CRFs as well, deals with multi-type data records with relations other than common field values. These methods improve the accuracy of de-duplication by capturing relational features in their models. They also belong to the learning paradigm, which requires labeled samples for training the model parameters. Due to the complexity of the model, the training and inferencing require considerable time, which poses scalability problem.

3 Situations and Characteristics

3.1 Situations

One situation of duplicated relational data can be found in [6], which gives an example of duplicated records of papers and their venues. In this example, the details of papers (author, title) form a database table and the details of venues (conference/journal name) form another table. Obviously, each paper links to a certain venue, forming a relation between the two records. This example can be further extended so that authors may form a separate table containing details of authors (e.g., name, email address) and papers link to certain records in the author table. This kind of normalization is common in designing databases. But it is not the favorite situation for traditional duplicate detection.

Data on the emerging Semantic Web [12] also has this relational feature. Ontologies are introduced to align data on the Semantic Web. A data record (or instance) then has several property values according to the underlying ontology. Particularly, it may have certain property values that refer to other records. Examples are like that an author record has a "publish" property with values pointing to several publication records. More over, unlike the strict database schema, ontology allows data records to be described in a very flexible way with different angles. For example, a publication record can use a reverse property of "publish", say "writtenBy", to refer to the author records. This flexibility, together with the characteristics of decentralization on the Semantic Web, poses challenges to record deduplication.

3.2 Characteristic of Relational Data

The main characteristic of relational data is certainly the relational feature, i.e., the links between different data records. This often implies that data records may have different types, like the discussed situation where author records link to publication records. Then, multi-type is another characteristic.

In the discussed Semantic Web situation, data instances are not formatted as well as in databases. They are often presented in XML format or described by certain languages (for example, OWL [13]). Therefore, such data instances are semi-structured. In addition, as users can choose different ways to express, the resulting data instances then have different perspectives, not as unified as those in databases.

4 Duplication in Relational Data

Duplication in relational data can happen on every type of related data records. However, due to the characteristics of relational data, there are some certain patterns among them, which allow us to define constraints. We first introduce some basic notations and then define constraint rules.

4.1 Notations

First, for a particular domain of interest, we can obtain a set of types T, and a set of properties P. There are two types of properties in P: data type properties that allow instances to be described with numbers and/or string values; and object properties that link instances to other instances with particular meanings (following the notions in OWL [13]). An instance then can be described with a type and a subset of properties and their corresponding values (numbers, strings, or other instances).

We identify two classes of instances. If an instance d_i has certain object property values that let it link to a set D_i of other instances, then d_i is identified as "*primary instance*". For any instance d_j ($d_j \in D_i$), d_j is identified as "*derived instance*". The two classes are not exclusive. That is, an instance can be both "primary" and "derived" as long as it points to other instances and has other instance pointing to itself. Given an object property link between two instances (denoted by $d_i \rightarrow d_j$), it is easy to determine the classes of the instances.

If two instances d_i and d_j actually refer to one same real world entity, then the two instances are regarded as duplicates (denoted as $d_i = d_j$). Duplicated instances may not be same in terms of their types, property values as they often come from different sources with different qualities and perspectives. But usually they have similar values. Traditional methods thus use certain similarity measures to compute degrees of similarity of two instances. Given a similarity function f, a clustering process can be conducted to group instances with high similarity degrees into same clusters. For an instance d_i grouped into cluster c_k, we denote as $d_i \in c_k$ or simply $c_k(d_i)$.

4.2 Constraint Rules

We define five constraint rules for duplicate detection using clustering approaches. Please note although we call all of them constraints, some actually act more like general rules with little constraint features.

Derived distinction. Given an instance d_p and $D_p = \{d_r | d_p \rightarrow d_r\}$, if $\forall d_i, d_j \in D_p, i \neq j$, then $d_i \neq d_j$.

This rule indicates that all the derived instances from *one* same primary instance should not be duplicates of each other. The reason is quite obvious. Firstly, the application of relating one instance to two or more same other instances is very rare. A paper is always written by different authors if it has more than one author. A conference, in principle, never allows two same papers to be accepted and published. Secondly, the relation between one instance and other several instances often occurs within one data source. Therefore, it is quite easy to maintain so that the derived instances from one same instance are not duplicates. Consider that a person manages his publications to ensure no duplicates occur on his/her web pages. As a result, the instance of this person links to different instances of publications.

Primary similarity. Given two *primary* instances d_a, d_b and one of the resulting clusters c, if $c(d_a, d_b)$, then d_a and d_b have high confidence to be duplicates. We denote $d_a \approx d_b$.

This rule prefers similar primary instances. This rule is based on the observation of the characteristic that primary instances are often described with more detailed and accurate information while derived instances are usually given less attention and hence have less and vaguer details. Therefore, similarity between primary instances are more reliable for duplicate detection.

Derived similarity. Given two *primary* instances d_a, d_b and $d_a \approx d_b$, if we have instances d_x, d_y and cluster c such that $d_a \rightarrow d_x$, $d_b \rightarrow d_y$, $c(d_x, d_y)$, then $d_x \approx d_y$.

This rule treats derived instances that fall in same cluster as duplicates if their corresponding primary instances are treated as duplicates. Strictly speaking, if two primary instances are duplicates, all of their corresponding derived instances should be duplicates as well. However, as noise often exists, it can not be guaranteed that the seeming primary instance duplicates are actual duplicates. To ensure high precision and to prevent false duplicate spreading, we only identify those derived instances that fall in same clusters to be duplicates.

Reinforced similarity. Given instances d_i, d_j, d_m, d_n and clusters c_k, c_l, if we have $d_i \rightarrow d_m$, $d_n \rightarrow d_j$, $c_k(d_i, d_j)$ and $c_l(d_m, d_n)$, then $d_i \approx d_j$ and $d_m \approx d_n$.

This rule addresses the issue of data expressed with different perspectives. Different sources have their own views and describe data from different angles. An entity may be described as a detailed primary instance in one source; But in another source, it could be a simple derived instance. while we may not be confident in the similarity between a primary instance and a derived instance that fall in one same cluster c_k, this similarity will be reinforced if their derived/primary instances also fall into one same cluster c_l. As a result, we treat both pairs as duplicates.

Boosted similarity. Given *derived* instances d_i, d_j, d_m, d_n and clusters c_k, c_l such that $c_k(d_i, d_j)$ and $c_l(d_m, d_n)$, if there exist instances d_x, d_y such that $d_x \not\approx d_y$, $d_x \rightarrow [d_i, d_m]$ and $d_y \rightarrow [d_j, d_n]$, then $d_i \approx d_j$ and $d_m \approx d_n$.

This rule reflects the notion of co-referencing. It is possible that two different instances mention two seemingly same instances that turn out to be different. But the possibility would be much less if more than one (unique) instances mention two sets of seemingly same but different instances. For example, two different papers may have one author's name in common which actually refers to two different persons; But it rarely happens that two papers have two authors' names in common which refers to four different persons. Ideally, if more frequent primary instances are found pointing to more sets of similar derived instances (which may be implemented by frequent item set mining [14]), the confidence of the results would be much higher.

Fig. 1 serves to illustrate the application patterns of different constraint rules we've defined.

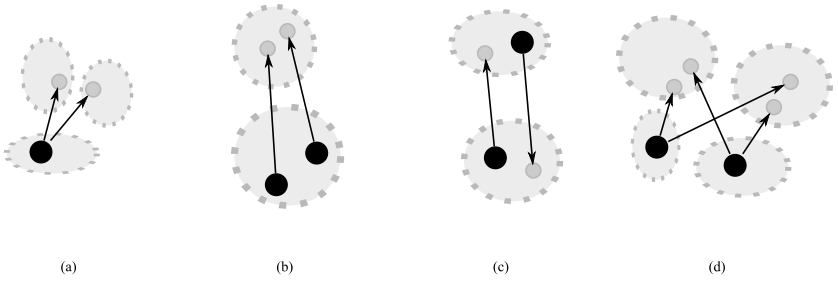


Fig. 1. Illustration of applications of different constraint rules in corresponding situations. (a) derived distinction; (b) primary similarity and derived similarity; (c) reinforced similarity; (d) boosted similarity.

5 Constrained Clustering

This section discusses how the above rules are incorporated in the clustering process. First we present the commonly used canopy clustering method in duplicate detection. Then we focus on our approach.

5.1 Canopy Clustering

Canopy clustering [8] is commonly used in duplicate detection [3,4,5]. It uses two similarity thresholds (T_{tight}, T_{loose}) to judge if an instance is closely/loosely similar to a randomly selected instance that acts as canopy center. All loosely similar instances will fall into this canopy cluster. But those closely similar instances will be removed from the list and never compared to another canopy center. Canopy clustering is very effective in duplicate detection as most instances are clearly non-duplicates and thus fall in different canopies. It is also very efficient since it often uses quick similarity measures such as TFIDF [15] computed using inverted index techniques.

Since the resulting canopies may be still large and overlap with each other, a second stage process such as Greedy Agglomerative Clustering (GAC) or Expectation-Maximization (EM) cluster are usually conducted within each canopy to yield refined results [8].

When canopy clustering is applied to duplicate detection in relational data directly, the performance may not be as good as it is used in normal data. This is because it ignores particular characteristics of relational data. For example, for two derived instances which may represent two different papers of one person, they can be so similar that canopy clustering (even with GAC or EM) treats them as duplicates.

5.2 Canopy Clustering with Constraints

To improve the performance of duplicate detection in relational data, we modified canopy clustering by incorporating the constraints we've defined. The resulting approach can be divided into four steps.

Input: Set of instances $D = \{d_1, d_2, \cdots, d_N\}$;
Similarity threshold T_{tight}, T_{loose}.
Output: Set of canopy clusters $C_1 = \{c_1, c_2, \cdots, c_K\}$.
Begin
$C_1 = \emptyset$; $D_{tmp} = D$;
while $D_{tmp} \neq \emptyset$ **do**
Create a new canopy cluster $c_{canopy} = \emptyset$;
Pick a random d_r in D_{tmp};
Let $c_{canopy} = \{d_i | d_i \in D_{tmp} \wedge sim(d_i, d_r) > T_{loose}\}$
subject to condition:
$\forall d_x, d_y \in c_{canopy}\ (x \neq y) \Rightarrow \{d_z | d_z \rightarrow d_x \wedge d_z \rightarrow d_y\} = \emptyset$;
Let $c_{core} = \{d_i | d_i \in c_{canopy} \wedge sim(d_i, d_r) > T_{tight}\}$;
$D_{tmp} = D_{tmp} - c_{core}$; $C_1 = C_1 + c_{canopy}$;
End while
Output C_1;
End

Fig. 2. Algorithm of step 1

The first step (step 1) is much like the first stage of canopy clustering except that it subjects to the constraint that no any two derived instances from one same instance fall into one same canopy. Fig. 2 shows the algorithm of this step. In the algorithm, function $sim(d_i, d_r)$ computes the degree of similarity between the instance d_i and d_r.

Although each resulting cluster is constrained to contain no two derived instances of one same instance, it still can not guarantee **derived distinction** due to the existence of overlapping canopies. If two clusters, each of which contains a derived instance of one same instance, both have an instance $d_{overlap}$, this instance then actually bridges the two different derived instances when we take a transitive closure. As a result, it violates **derived distinction**.

Step 2 then is designed to ensure **derived distinction** thoroughly. it is done by checking the overlapping instances and only allowing them to be with the most similar derived instance. Fig. 3 shows the algorithm of step 2.

Input: Set of instances $D = \{d_1, d_2, \cdots, d_N\}$;
Set of clusters C_1 generated from step 1.
Output: Set of clusters C_2.
Begin
for each $d_i \in D$ **do**
$D_{derived} = \{d_j | d_i \rightarrow d_j\}$;
for any $d_x, d_y \in D_{derived}$ $(x \neq y)$ **do**
if $\exists d_z \in D$, $c_m, c_n \in C_1$ such that $d_z, d_x \in c_m$ and $d_z, d_y \in c_n$
let $\delta = sim(d_z, d_x) - sim(d_z, d_y)$;
if $\delta > 0$ then remove d_z from c_n else remove d_z from c_m;
end if
end for
end for
Output C_1 as C_2;
End

Fig. 3. Algorithm of step 2

Input:
Set of instances D;
Set of clusters C_2 from step 2.
Output:
Set of duplicate pairs P_1.
Begin
$P_1 = \emptyset$;
for each $c_i \in C_2$ **do**
for any $d_x, d_y \in c_i (x \neq y)$ **do**
//primary similarity
//and derived similarity
if $d_x \rightarrow d_m$ and $d_y \rightarrow d_n$ and $\exists c_j \in C_2$, $c_j(d_m, d_n)$
$P_1 = P_1 + (d_x, d_y) + (d_m, d_n)$;
end if
//reinforced similarity
if $d_x \rightarrow d_m$ and $d_n \rightarrow d_y$ and $\exists c_j \in C_2$, $c_j(d_m, d_n)$
$P_1 = P_1 + (d_x, d_y) + (d_m, d_n)$;
end if
end for
end for
Output P_1;
End

Fig. 4. Algorithm of step 3

Input:
Set of instances D;
Set of clusters C_2 from step 2.
Set of Pairs P_1 from step 3.
Output:
Set of duplicate pairs P_2.
Begin
$P_2 = \emptyset$;
for any $d_x, d_y \in D$ such that
$x \neq y, d_x \not\approx d_y$ **do**
$P_{tmp} = \emptyset$;
while $\exists d_m, d_n, c$ such that
$d_x \rightarrow d_m, d_y \rightarrow d_n, c \in C_2, c(d_m, d_n)$
do
$P_{tmp} = P_{tmp} + (d_m, d_n)$;
end while
if $|P_{tmp}| > 1$ then $P_2 = P_2 + P_{tmp}$;
end for
$P_2 = P_2 + P_1$;
Output P_2;
End

Fig. 5. Algorithm of step 4

The purpose of step 3 is to extract high confident duplicate pairs within each cluster in C_2 by following the definition of **primary similarity**, **derived similarity**, and **reinforced similarity**. In step 4, **boosted similarity** is implemented to extract frequent co-referenced instance pairs as potential duplicates from the clusters. The algorithms of step 3 and 4 are illustrated in Fig. 4 and Fig. 5 respectively. After all the potential duplicate pairs are extracted, a transitive closure is performed to generate the final results.

Please note the constraint rules reflected in these steps are not incompatible with other refinement processes such as GAC. They can be added in the procedure to work together with the constraint rules. For example, GAC can be added after step 2 to further refine clusters.

5.3 Computational Complexity

We informally address the complexity of our approach. The algorithm in step 1 performs a constraint check that normal canopy clustering doesn't have. This extra check does about $O(km^2)$ judgements where k is the number of clusters and m is the average size of each cluster. In the setting of duplicate detection, the size of each cluster usually is not very big ($k \gg m$). The complexity of cluster adjustments in step 2 depends on the number of primary instances (p) and the average size of derived instances a primary instance has (q), which is about $O(pq^2)$. Normally, $n > p \gg q$ where n is the number of all the instances. In step 3, the extraction of potential duplicate pairs out of each cluster performs at the complexity level of $O(km^2+km^2q^2)$ if we include the checking for the derived instances. The complexity in step 4 depends on the implementation. Our simple implementation operates at $O(p^2q^2)$. After all, it should be noted that all the above operations (checking, adjusting, extracting) don't involve very expensive computations. In fact, our experiments reveal that a lot of time is spent in computing the similarity between instances.

6 Experiments

There exist some commonly used data sets for duplicate detection experiments, but data instances in them don't have many types and in-between relations. And mostly they are presented from one unified perspective. This doesn't represent well the real world situations of relational data. Therefore, we collected data from different sources to build the data set for our experiments. The data set is mainly about papers, authors, conferences/jounals, publishers and their relations. Such data is collected from DBLP web site (`http://dblp.uni-trier.de`) and authors' home pages. These data instances are converted into a working format but types, relations and original content values are preserved. Manual labeling work is done to identify the true duplicates among the data for the purpose of evaluation of approaches in the experiments. Totally, there are 278 data instances in the data set referring to 164 unique entities. The size may not be so big, but duplicate detection in it may not be easy since there are a certain amount of different instances with very high similarity, for example, different papers within same research fields and different authors with same/similar names. The distribution of duplicates is not uniform. About two-third of instances have one or two references to their corresponding entities. The most duplicated entity has 13 occurrences.

Same as [8], we use standard metrics in information retrieval to evaluate the performance of clustering approaches for duplicate detection. They are precision, recall and F measure. Precision is defined as the fraction of correct duplicate predictions among all pairs of instances that fall in the same resulting cluster. Recall is defined as the fraction of correct duplicate predictions among all pairs of instances that fall in the same original real duplicate cluster. F measure is the harmonic average of precision and recall.

We evaluate our approach in comparison with the canopy-based greedy agglomerative clustering approach (CB+GAC) [8]. CB+GAC also performs canopy clustering first but with no constraints. It then refine each canopy cluster using GAC: initialize each instance in the canopy to be a cluster, compute the similarity between all pairs of these clusters, sort the similarity score from highest to lowest, and repeatedly merge the two most similar clusters until clusters reach to a certain number. Table 1 shows the evaluation results of different approaches. The two threshold parameters for canopy clustering in this evaluation are set as $T_{tight} = 0.5$ and $T_{loose} = 0.35$, which are obtained through a tuning on a sampled data set. The number of clusters is then automatically determined by the two parameters. In the table, "CB+GAC" is the general clustering approach we have just discussed. "Step 12" is the approach that only performs step 1 and step 2 (refer to Section 5.2) and then returns the resulting clusters. "Step 12+GAC" is the approach that performs GAC after step 1 and step 2. "Step 1234" obviously is the approach that performs all the steps to impose all the constraints we've defined on the clusters. From the table, we can see that by incorporating constraint rules, the overall F measure improves along with the precision. In particular, when all the constraints are applied, the precision increases up to 20%, which indicates that our approach can predict duplicate with very high accuracy.

Table 1. Performance of different approaches

Approach	Precision	Recall	F score
CB+GAC	0.717	0.806	0.759
Step 12	0.728	0.877	0.796
Step 12+GAC	0.784	0.817	0.800
Step 1234	0.921	0.721	0.809

Fig. 6 shows the sensitiveness of precision of different approaches to the loose similarity threshold (T_{loose}) in the canopy clustering. Since in our approach some constraint rules are used to extract duplicate pairs out of working clusters, the quality of the initial canopy clustering may affect the performance. That is, when T_{loose} becomes more loose, each canopy cluster may have more false duplicates, which might affect the performance of those constraint rules used for duplicate extraction. The trend of dropping precision while T_{loose} decreases is well revealed in approach "Step 12". However, the dropping trend of approach "1234" is slightly better than that of "Step 12", which means that constraint rules used in step 3 and 4 can tolerate noisy canopy clusters to certain degrees.

Table 2 shows the precision of detecting duplicated pairs in different steps in our approach. This can be used to roughly estimate contributions of different constraint rules as they are implemented in different steps. The evaluation on our data set shows that the main contribution to the improved precision is made in step 3, where constraint rules of "primary similarity", "derived similariy" and "reinforced similarity" are imposed.

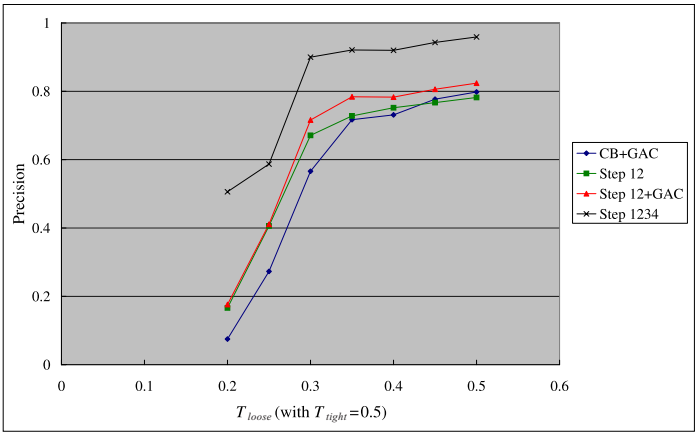


Fig. 6. Sensitiveness of precision to T_{loose} for different approaches

Table 2. Precision of detection of duplicated pairs in different steps

	Step 1	Step 2	Step 3	Step 4
Precision	0.650	0.682	0.881	0.888

7 Conclusions and Future Works

This paper discusses the characteristics of relational data from the perspective of duplicate detection. Based on these characteristics, we have defined constraint rules, which are implemented and incorporated in our cluster-based approach. Experiments show that our approach performs well with improved accuracy in term of precision and recall. Experimental evaluations also reveal that the use of constraint rules increases the precision of duplicate detection for relational data with multiple perspectives.

One of the further studies is to conduct further experiments with larger data sets. Currently, we are keeping collecting data from different sources and converting and labeling them to build larger data sets. Besides the evaluation of accuracy on the large data sets, the efficiency of the approach will be formally evaluated.

Another further study is to design quantitative metrics to reflect characteristics of duplicated relational data. The ideal metrics will act as *soft* constraint rules. Thus, they are expected to be more adaptive to different duplicate problems.

Acknowledgements

The authors sincerely thank the anonymous reviewers for their valuable comments. The work presented in this paper was partially supported by Australian Research Council (ARC) under discovery grant DP0559213.

References

1. Fellegi, I.P., Sunter, A.B.: A theory for record linkage. Journal of the American Statistical Association **64** (1969) 1183–1210
2. Winkler, W.E.: Methods for record linkage and bayesian networks. Technical report, U.S. Census Bureau, Statistical Research Division (2002)
3. Cohen, W.W., Richman, J.: Learning to match and cluster large high-dimensional data sets for data integration. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2002) 475–480
4. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2003) 39–48
5. Singla, P., Domingos, P.: Collective object identification. In Kaelbling, L.P., Saffiotti, A., eds.: IJCAI, Professional Book Center (2005) 1636–1637
6. Culotta, A., McCallum, A.: Joint deduplication of multiple record types in relational data. In: Fourteenth Conference on Information and Knowledge Management (CIKM). (2005)
7. Newcombe, H., Kennedy, J.M., Axford, S.J., James, A.P.: Automatic linkage of vital records. Science **130** (1959) 954–959
8. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2000) 169–178
9. Tejada, S., Knoblock, C.A., Minton, S.: Learning domain-independent string transformation weights for high accuracy object identification. In: KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM Press (2002) 350–359
10. Vapnik, V.N.: The nature of statistical learning theory. 2nd edn. Statistics for engineering and information science. Springer, New York (1999)
11. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Brodley, C.E., Danyluk, A.P., eds.: ICML, Morgan Kaufmann (2001) 282–289
12. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284**(5) (2001) 34–43
13. McGuinness, D.L., Harmelen, F.v.: Owl web ontology language overview. w3c recommendation. http://www.w3.org/tr/2004/rec-owl-features-20040210 (2004)
14. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. SIGMOD Rec. **22**(2) (1993) 207–216
15. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**(5) (1988) 513–523

Understanding Research Field Evolving and Trend with Dynamic Bayesian Networks*

Jinlong Wang[1], Congfu Xu[1,**], Gang Li[2],
Zhenwen Dai[1], and Guojing Luo[1]

[1] Institute of Artificial Intelligence, Zhejiang University
Hangzhou, 310027, China
[2] School of Engineering and Information Technology, Deakin University,
221 Burwood Highway, Vic 3125, Australia
zjupaper@yahoo.com, xucongfu@cs.zju.edu.cn,
gang.li@deakin.edu.au, edmund@zju.edu.cn,
guojingluo@gmail.com

Abstract. In this paper, we proposes a method to understand how research fields evolve through the statistical analysis of research publications and the number of new authors in a particular field. Using a Dynamic Bayesian Network, together with the proposed transitive closure property, a more accurate model can be constructed to better represent the temporal features of how a research field evolves. Experiments on the KDD related conferences indicate that the proposed method can discover interesting models effectively and help researchers to get a better insight looking at unfamiliar research areas.

1 Introduction

Detecting emerging trends and field evolving in scientific disciplines can significantly improve the ability of researchers to catch the wave in a timely manner. Most existing works [1, 2] use statistical topic models such as Latent Dirichlet allocation (LDA) [3] for topic extraction and analysis. Recent work has been concerned with temporal documents [4, 5]. These works can create fine-grained, immediately interpretable topics that are robust against synonymy and polysemy. However, the topic models need a pre-specified number of latent topics, and have to be done through manual topic labeling, which is usually a labor intensive process. More importantly, it is difficult to show the different topics which make up different fields and how these topics evolve and develop from a global view. Mannila et al [6] tried to find one or a (small) set of partial orders that fit the whole data set accurately from an optimization point of view. Due to complexity, only series-parallel orders [7] were considered in [6]. Furthermore,

* Supported by the National Natural Science Foundation of P.R.China (60402010) and Zheiang Provincial Natural Science Foundation of P.R.China (Y105250).
** Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 320–331, 2007.

the Global Partial Orders (GPO) is a qualitative model, local information was sacrificed for global benefits.

In this paper, we attempt to detect how a research field evolves by analyzing the publication track records of authors, especially new authors. The global model constructed from these sequential track records can give us a macro-view of field evolving trend, since "a burst of authors moving into a conference C from some other conference B are actually drawn to topics, which are currently hot at C" as the results in [8]. This could be considered as a graph mining task but we focus on constructing a probabilistic graphical models from sequential data set.

In this paper, we use the Dynamic Bayesian Networks (DBN) [9], which extends the graphical models to accommodate temporal processes represented in the sequential data. We can control the model complexities according to the users' request. With such a method, more detailed information can be obtained in comparison with the GPO model.

However, the current DBN models with the first-order Markov transition can not catch all the information in field evolving trend and it is infeasible if high-order Markov transitions are considered due to the complexity. In this paper a transitive closure method is proposed to capture the evolving across several time slices. This global model constructs a sequential publication record of new authors to highlight how the research field evolves macroscopically. The model is constructed using information from different time periods and this allows a general publication trend to be captured. Finally, this model can be combined with a topic model and topic description to give a user a better understanding of a research field.

2 Problem Statement

Research community is important for academic communications, and many research communities have their individual associations to best foster academic research and collaborations. For example, as the first society in computing, ACM (Association for Computing Machinery) has 34 distinct Speciate Interest Groups (SIGs)[1] in a variety of research fields to address different research interests. They organize and sponsor a large number of leading conferences in many research fields, such as SIGGRAPH, SIGMOD, SIGKDD, SIGIR, etc. Most of these premier conferences represent the frontier of their corresponding sub-fields in computer science, such as SIGGRAPH for computer graphcs, SIGKDD for knowledge discovery and data mining, etc. These conferences attract not only the researchers in their local fields but also large proportion of top researchers from other fields, who publish their research outputs in those top conferences. Through investigating these top conferences, it is possible to gain an understanding of the evolution of a particular research field.

According to the difference of duration time, in one time slot, members in a community can be divided into three types:

[1] http://www.acm.org/sigs/

new author: authors who publish papers in this community for the first time;
regular author: regular contributors to the community;
retiring author: authors who may leave the community soon.

Though regular authors may shape the research field, the new comers are more important in order to sense the emerging research topics. They often not only bring in new resources and thoughts into a community, but also participate in the hottest topics. Especially, in rising and developing fields, the role and influence of new authors are more obvious. Through analyzing this type of new authors by their publication records, we can obtain an understanding of how a research field evolves, and reveal the trend through comparing the models in different time periods.

2.1 Publication Track Records

The objective of this paper is to discover a global model based on the sequential publication records of new authors in the interested community. The definition of the publication track records is as the following.

Definition 1. ***The author's publication track records*** **(APTC)** *is defined as a sequence* $S =< s_1, s_2, \cdots, s_n >(s_i \subseteq C)$*, where* $C = \{c_1, \cdots, c_m\}$ *be the conference set we analyzed, essentially denoting the field. This sequence shows the conference attending history of an author ordered by year.*

An APTC records the research sequence of an author, and this sequence represents their research field during different period. In this paper, we focus on those premier conferences in which the authors or research groups are generally stable, and the authors in those conferences are generally quite focused on their research area, and seldomly publish their papers everywhere.

Definition 2. *For one target conference,* ***the new author's publication track records*** **(NAPTC)** *is a sequence, of the new comer's publishing track before he or she first published a paper on that conference.*

2.2 Problem Formalization

In analyzing, we assume that each NAPTC is generated independently. Since our goal is to analyze the field evolving, we assume each conference focus on one research field. This is usually true in reality for majority of conferences. Especially, in the computer science, most famous conferences focus on one field. Thus, we can use the conference name to represent its research field.

The formal definition of the problem can be denoted as: given the target conference and time, with the new authors' publication track sequence $S = \{S_1, S_2, \cdots, S_m\}$($S_i$ is ith new comer), constructing the global model using Dynamic Bayesian Network (DBN) [10]. The model learned from data aims to find the best model of the joint probability distribution of all conferences C underlying the temporal process. This is a structural learning process. In the next section, we present an unified probabilistic approach to the constructing of a global model.

3 Method

DBN, represents both the inter-slice conditional independence assumptions, and the intra-slice conditional independence assumptions. Among which we only care about the inter-slice links. In this section, we first preprocess the data, then introduce some methods to improve model constructing, and give an algorithm process, finally interpret the model discovered.

3.1 Preprocessing

When the target conference and time are given, the publishing track of new comers compose a subset of C. It often contains hundreds of conferences so the computing will be intractable due to the complexity of BN structure learning. Moreover, there are many conferences that appear only once or twice. This largely increases the computing complexity, but gains little benefit and even destroys the conciseness of the result. Therefore we only focus on top k conferences according to the support number.

Assuming $\{c_1, \cdots, c_k\}$ is the conference variable set analyzed, $c_i \in \{0, 1\}$. $c_i[t]$ is the random variable that denotes the value of the attribute c_i at time t, and $C[t]$ is the set of random variable $c_i[t]$.

In the NAPTC listings, many conferences have multiple representations for each year the conference has run. As such, many feedback loops can emerge which can impact the expression and understanding of the final result. Also when a researcher publishes a paper in one conference, it usually means that he has involved in the corresponding field. As such, in this paper we will only record the first participating.

3.2 Transitive Closure

Table 1. An example

Sequence	Times
a c b	10
a d b	10
a e b	10
a f b	10
a g b	10

After preprocessing, there are k binary random variables $C[t] = \{c_1[t], c_2[t], \cdots, c_k[t]\}$ in each time slice t. Based on the the first-order Markov property, the state sequences can only be represented into a set of consecutive one time slice transition. For example the state sequence $C_1C_2C_3C_4$ can only be divided into three time slice transitions $C_1 \rightarrow C_2$, $C_2 \rightarrow C_3$, $C_3 \rightarrow C_4$. This is usually not accurate enough for our task as shown in the example below.

Considering the data set in Table 1. It is a part of sequence data. The total number of sequences is 1000 and the number of appearance of each node a, b, c, d, e, f and g is 100. In the remaining part of a sequence, there are no one time slice transitions which is the same as Table 1, such as $a \rightarrow c, a \rightarrow d, f \rightarrow b, \cdots$. With a greedy search and BDeu score, we construct DBN. In the result, no transition is significant enough to appear in DBN model so the resulting graph is a set of isolated nodes. This loses one important information the transition from a to b. The reason is that the model

can only deal with transitions for consecutive time slices. However, this transition is valuable for our problem. Considering when a scholar published a paper in ICML 2000, this research field may affect his following research, not only the immediate ones. To solve this problem, we introduce the idea of transitive closure and use *Property 1* to modify the definition of time slices in the first-order markov model.

Property 1. The transition probability of any two random variables across any time slices is equal. The corresponding formula is $P(c_i|c_j) = P(c_i[t]|c_j[t']), i, j \in \{1, \cdots, k\}, t' \in [1, t-1]$.

In our track record sequences, the data is sparse and the sequence is usually short, showing the time span is not large. Furthermore, a conference represents one field only. In this situation, the field transitions almost remain stable, even if spanning limited time slices. Thus, in our two time slice model we define time slice 0 to mean the start of a transition and time slice 1 to mean the destination. The transition can cover any time slices. Instead of dividing the state sequences into a set of one time slice transitions, we now need to generate the transitive closure about the set of one time slice transitions. The state sequence $C_1C_2C_3C_4$ will generate a set of candidate transition pairs $C_1 \rightarrow C_2, C_1 \rightarrow C_3, C_1 \rightarrow C_4, C_2 \rightarrow C_3, C_2 \rightarrow C_4, C_3 \rightarrow C_4$. With the transitive closure, the DBN constructed from Table 1 has the transition $a \rightarrow b$. This property improves the model accuracy.

3.3 The Prior Model

In transitive closure, the first-order extension can not capture all possible transition probabilities, especially in sequence data and the activities occur with sequence, such as $P(C_t|C_{t-2}, C_{t-1})$ (the second-order markov process) and the $P(C_t|C_{t'}, \cdots, C_{t''})(t > t'' > t')$. For a better representation of the sequence, we need to consider these transitions. However, considering all these transitions increases the complexity. The model does not need to be an exact match to, or model all features of, real sequence data, so long as it captures many of the important sequence features. Some methods consider the top k transition. The value of k is hard to define and may lose some important information. Using the global partial order as the prior model, we can effectively obtain the sequence.

As a generative model, global partial order describes a set of sequences using a mixture model of partial orders. The likelihood of a given partial order producing a sequence compatible with it is inversely proportional to the number of total orders compatible with the partial order. The method tries to find a set of partial orders that are specific enough to capture significant ordering information contained in the data. Through using the trivial partial order and an unrestricted partial order, the global optimization model can be found. Using the results to initialize a search over unrestricted partial orders with DBN, we can obtain a good result. Due to the high complexity of global partial order constructed, we restrict the model learning to a small number of nodes $h(<= k)$.

3.4 The Procedure of Constructing DBN

With the descriptions mentioned above, we can construct our DBN as Algorithm 1. In model learning the objective is to maximize the posterior probability $P(M|S)$ of M given S, where S is our sequence data set and M is the model. Since with small amounts of data in our problem, BIC/MDL is known to over-penalize, we use the BDeu scoring metrics [11] in this paper.

Algorithm 1. The procedure of DNB constructing

Data: The conference sequence set (SS) after being preprocessed, parameter k and h.
Result: The global model constructed based on Dynamic Bayesian Network.
begin
  Step 1. Hold the top k conference variables in SS.
  Step 2. Obtain the global partial order M from SS with top h variables.
  Step 3. Transitive closure computation.
  for $\forall s \in SS$ **do**
    Divide the sequence s into binary transition pairs;
    Count their transitive closure;
    Store the transitive closure of s in sequence set SSC;
  Step 4. Construct Dynamic Bayesian Network on SSC.
  while *true* **do**
    generate all the network structures M' from M;
    for $\forall M'_i \in M'$ **do**
      computing its BDeu score;
    let i be the index such that M'_i has the highest BDeu score.
    if *the BDeu score of* M'_i *is higher than* M **then**
      $M = M'_i$;
    else
      break while;
  Step 5. Draw the DBN model as a graph;
  Represent the inference probability.
end

In step 1, we preprocess the sequence and hold the top k conference variables. Step 2 constructs the global partial order from SS with the top h variable, then uses this as the prior model. With step 3, the conference sequences are transformed with the transitive closure. In step 4, based on the transitive closure and the prior model, we can construct the Dynamic Bayesian Network. In the search process, we use the greedy search with random restarts [12], which can restart from another random graph to escape the local maximum. In step 5, we draw the DBN with the top score value model as a brief graph, and interpret the model through the CPT or influence scores [13].

3.5 Network Interpretation

In the network interpretation we only care about the field (conference variable) evolving represented by inter-slice links. Thus in the brief graph the nodes are defined as the random variables and the links are defined as the inter-slice links. At a quantitative level, relationships between variables are described by a family of joint probability distributions (conditional probability table, CPT) that are consistent with the independence assertions embedded in the graph. Sometimes, people usually want to know the influence that a parent gives to its child, we consider the influence score [13] as an alternative choice. The influence score is arranged between -1 and 1. Positive numbers represent activating relationships of a parent on a child, while negative numbers represent repressing relationships.

In our model based on DBN, in general, each slice can have any number of state variables. If there is a node connected by several nodes, we need to carefully explain it as the value of the node is influenced by those nodes connected to it. The quantitative analysis of the influence should rely on the CPTs and with the help of influence score.

Additionally, in our graph, positive correlation in some degree means the high probability of the sequential appearance comparing with negative correlation and independence. So from the positive correlations a set of binary orders of the nodes covering one time slice can be created, and with concatenating these binary orders together a global order can be generated. Therefore the nodes on the bottom of the global order usually appear later than the nodes above them.

4 Experiments

We apply the method presented in Section 3 to construct our DBN model. We also use the LDA model[2] for topic discovery. Empirical results show that our model provides a compact representation, which is better than global partial order model and current DBN method. Through our model, we can provide a potential source for understanding the sequential data deeper and catching how a research field evolves and develops better.

4.1 Data Preparation

In the experiments, we use the two data sets, the first is the DBLP set of datasets[3]. Through DBLP(by October 2006), we can extract the publication track sequence of authors. In these data, the duplicate names are only a tiny part of the whole dataset, thus will not be a problem in our model construction.

The second data set consists of the abstracts in SIGKDD conference proceedings from 2001 to 2006. Through these data, we can extract the topics for better explaining our model's advantage. All the abstracts were crawled from the ACM

[2] http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm
[3] http://dblp.uni-trier.de/

digital library[4]. For better discovering the topic, we filter some phases, such as "the", "a", etc., which may affect the accuracy of the results.

4.2 Global Model

This section compares the all global models as the following:

1. Global partial order (GPO);
2. **D**ynamic **B**ayesian **N**etwork (DBN);
3. DBN with **T**ransitive **C**losure (DBN-TC);
4. DBN with **T**ransitive **C**losure and **P**rior **N**etwork (DBN-TCPN);

The GPO model is constructed based on [6], the others DBN, DBN-TC and DBN-TCPN are constructed based on Dynamic Bayesian Network. The DBN model constructed without the transitive closure is based on the first-order markov. The DBN-TC model is based on DBN with the transitive closure. The DBN-TCPN is the model proposed in this paper, with transitive closure and prior network GPO. The sequence data are gathered from publication track sequences of new authors in SIGKDD 2006. The prior network presents the global partial order model with top 12 conference variables. And the DBN models are constructed from the top 20 conference variables.

Fig.1 shows the four models. Classifying the edges into positive and negative correlation according to the influence score [13], all three DBN models have no negative edges. The DBN (DBN, DBN-TC and DBN-TCPN) models all present more information than GPO model. In the following, we describe our model DBN-TCPN and use it to understand how research field evolves.

4.3 Model Detail

In this subsection, we describe this model in detail, including both the topology and probabilistic table. Fig.1(d) or Fig.2(d) represents the topology of publication track records with new comers in SIGKDD 2006. As the model does not have negative correlations. This is quite convenient for us to gain the sequential feature of the conferences. The figure shows many application fields. Especially www based research conferences appear in the low level in the network, such as SIGIR, CIKM, WWW, APWeb. It also indicates how the KDD research field develops and is enlarged, due to the emergence of ICDM and PAKDD, two other conferences in the KDD field.

From the part of the corresponding probability table shown in the Fig.1(d), it is evident that many people change research field from theory conference to application conference, such as the CPT for the WWW conference node, the value "1" means that researchers of the discrete algorithm symposium SODA, a famous forum focusing on discrete problems, bring their research into WWW when they have no Expert Systems field. And from the CPT for the ICDM conference node, multimedia domain is related positively to the data mining ICDM.

[4] http://portal.acm.org/dl.cfm

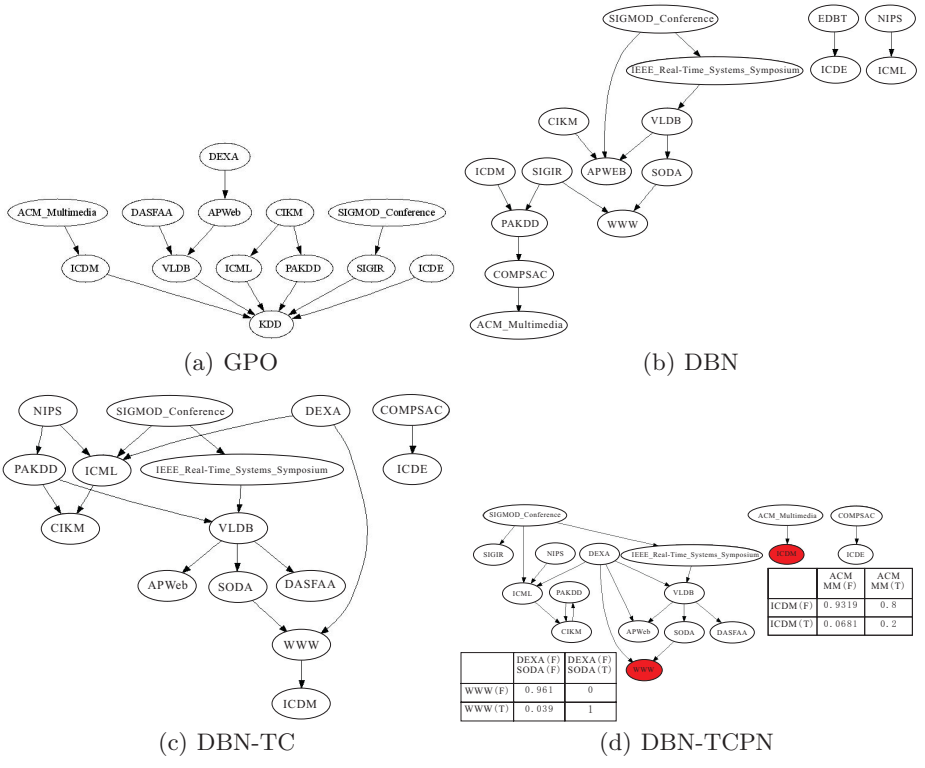


Fig. 1. Global model comparision

4.4 Comparision with Topic Model

In this section, we use our DBN-TCPN model (DBN with **T**ransitive **C**losure and **P**rior **N**etwork) constructed in different period in SIGKDD conference to reveal the evolving and trend in data mining and knowledge discovery. For better explanations, we also compare it with the topic model.

In the model construction, the prior network is with top 12 conference variables. And the DBN-TCPN models are constructed with top 20 variables except for the 2002 year with top 22 (This is due to that the conference variable support counts are difficult to be distinguished). From the models in Fig.2, where isolate variables have not been shown, we can find that the model constructed is more and more sparse gradually as shown by the statistical property shown in Fig.2. Such as in the 2000, 2004 and 2006 year, the variable number is all 20, but the edge number decreases from 30 to 25 to 18. This shows that the effect of the KDD field is larger and larger, attracting various fields' researchers.

Analyzing the figure closely, Fig.2(a) indicates that new comers in 2000 usually come from the database field (SIGMOD, ICDE, PODS, VLDB), artificial intelligence (AAAI/IAAI, IJCAI, UAI) and machine learning (ICML). These conferences appear in the low-level in the graph. This indicates that they have

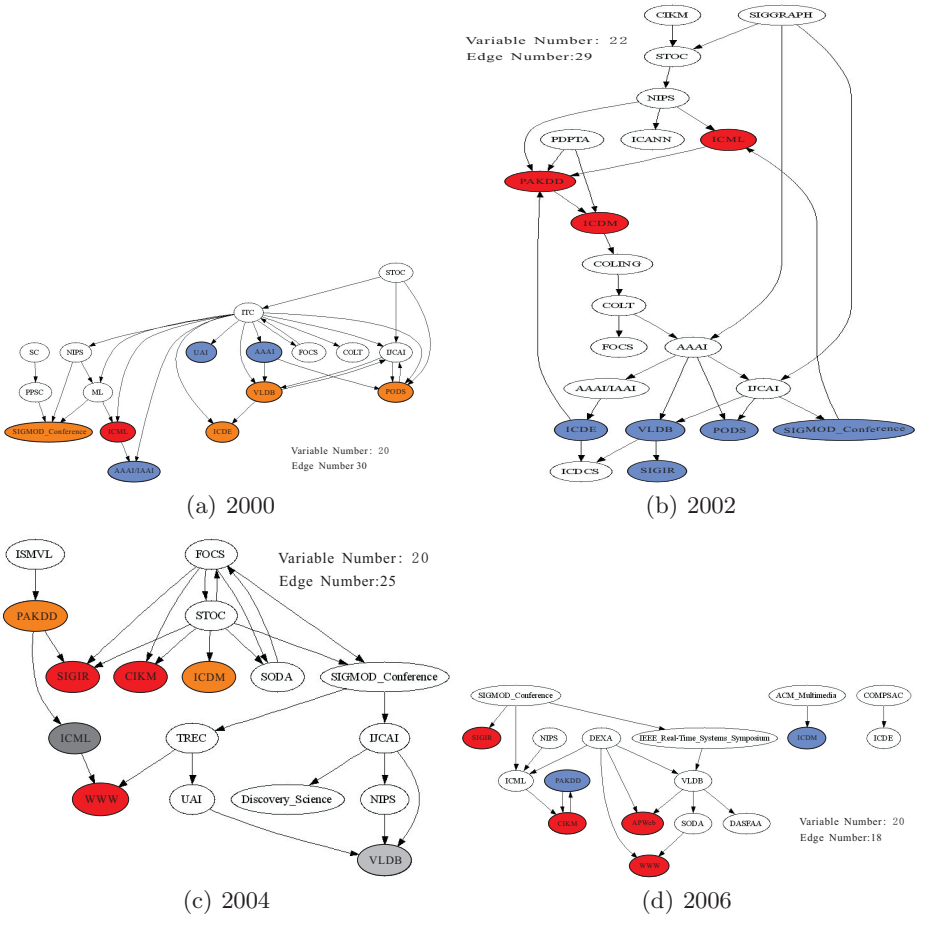


Fig. 2. Global model with different time in SIGKDD. In the figures, we use different color to differentiate different fields' conferences, which may be interesting.

some other domain background, and the research field evolves from theoretical computing to DB, AI and ML, with further transforms to DM and knowledge discovery (KDD). In that period, the DM and KDD fields where in their infancy, and many algorithmic and theoretic problems needed to be studied. In 2002 (Fig.2(b)) the DB field maintains a presence in KDD, and the KDD field has developed, some forums (PAKKD and ICDM) have also attracted the researchers from other fields. Additionally, the field closed to SIGIR emerged. In 2004, KDD continues developing, and application fields play an important role in KDD, they introduce the web application to web mining using a machine learning method, and this trend continues in 2006. Moreover, the number of KDD applications has increased in relation to other research topics such as multimedia (ACM_Multimedia), SIGIR and WWW.

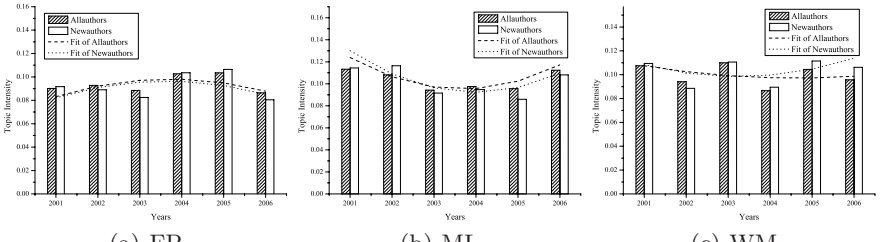
For a better comparison, we used the yearly data of a target conference to analyze trends in topics over time based on topic model. Using the topic obtained earlier, the documents were partitioned by year, and for each year all the documents were assigned to the topic using the model. These fractions provide useful indicators of relative topic popularity in research literatures in recent years. And for better analysis, we respectively compute the one of all authors and new authors.

Table 2 lists the 10 topics analyzed based on SIGKDD abstract data from 2001 to 2006.

Table 2. Topics discovered with manual labels

Topic No.	manual namings	Topic No.	manual namings
0	classification	5	web mining
1	graph mining	6	clustering
2	network learning	7	probabilistic model
3	application based on rule	8	text mining
4	real application	9	algorithm design

For convenience, we combine the related topics, such as topic No. 0 (classification), topic No. 6 (clustering) as machine learning topics, topic No. 2 and No. 5 for web mining, and average their topic intensity. Figure 3 shows topic intensity of all papers by new authors and all authors in Machine Learning (ML), Algorithm Design (AD) and Web Mining (WM). To represent the results, we use a polynomial function to show that the data fit the line in Figure 3. The results indicate that the topics of new authors are consistent with that of all authors, and this is consistent with our assumptions and model described above. However, we cannot see KDD development and other related field evolving information from how a topic changes, which makes it difficult to understand how a particular research field evolves. If these topic models can be integrated with a global model, the quality of prediction can be improved.



(a) FP (b) ML (c) WM

Fig. 3. Topic Model

5 Conclusions

Understanding how a research field evolves is essential for scientists, analysts and decision makers to identify emerging trends in the body of scientific literature. This paper provides a method to help researchers understand unfamiliar subject areas and guide them toward hot topics and trends using a global model. This was accomplished by combining a Dynamic Bayesian Network with the proposed transitive closure property, to more accurately model the temporal features of how a research field evolves. Through the introduction of Global Partial Order (GPO) model (a good global model), we can synthesize the ordering information. Models constructed are compared in order to identify change as a sign of an emerging trend. The experimental results with SIGKDD show our model is effective. Also the models constructed represent the research trend accurately in data mining and knowledge discovery field. Especially with the comparative analysis between our model and topic model in the experimental data set, the result shows the consistent and our model can reveal more trend information than topic model.

References

[1] Steyvers, M., Smyth, P., Rosen-Zvi, M., Griffiths, T.: Probabilistic author-topic models for information discovery. In: KDD 2004. (2004) 306–315
[2] Griffiths, T., Steyvers, M.: Finding scientific topics. In: PNAS. Volume 101(suppl.1). (2004) 5228–5235
[3] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
[4] Wang, X., McCallum, A.: Topics over time: a non-markov continuous-time model of topical trends. In: KDD 2006. (2006) 424–433
[5] Blei, D.M., Lafferty, J.D.: Dynamic topic models. In: ICML 2006, New York, NY, USA, ACM Press (2006) 113–120
[6] Mannila, H., Meek, C.: Global partial orders from sequential data. In: KDD 2000. (2000) 161–168
[7] Valdes, J., Tarjan, R.E., Lawler, E.L.: The recognition of series parallel digraphs. In: STOC 1979, New York, NY, USA, ACM Press (1979) 1–12
[8] Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: Membership, growth, and evolution. In: SIGKDD 2006. (2006) 44–54
[9] Friedman, N., Murphy, K.P., Russell, S.J.: Learning the structure of dynamic probabilistic networks. In: UAI. (1998) 139–147
[10] Neapolitan, R.E.: Learning Bayesian Networks. Prentice Hall (2003)
[11] Heckerman, D., Geiger, D., Chickering, D.M.: Learning bayesian networks: The combination of knowledge and statistical data. Machine Learning **20** (1995) 197–243
[12] Heckerman, D.: A tutorial on learning with Bayesian networks. In Jordan, M., ed.: Learning in graphical models. MIT Press, Cambridge, MA, USA (1999) 301–354
[13] Yu, J., Smith, V.A., Wang, P.P., Hartemink, A.J., Jarvis, E.D.: Using bayesian network inference algorithms to recover molecular genetic regulatory networks. In: ICSB 2002. (2002)

Embedding New Data Points for Manifold Learning Via Coordinate Propagation

Shiming Xiang[1], Feiping Nie[1], Yangqiu Song[1],
Changshui Zhang[1], and Chunxia Zhang[2]

[1] State Key Laboratory of Intelligent Technology and Systems,
Department of Automation, Tsinghua University, Beijing 100084, China
{xsm, zcs}@mail.tsinghua.edu.cn, {nfp03, songyq99}@mails.tsinghua.edu.cn
[2] School of Computer Science, Software School,
Beijing Institute of Technology, Beijing 100081, China
cxzhang@bit.edu.cn

Abstract. In recent years, a series of manifold learning algorithms have been proposed for nonlinear dimensionality reduction (NLDR). Most of them can run in a batch mode for a set of given data points, but lack a mechanism to deal with new data points. Here we propose an extension approach, i.e., embedding new data points into the previously-learned manifold. The core idea of our approach is to propagate the known coordinates to each of the new data points. We first formulate this task as a quadratic programming, and then develop an iterative algorithm for coordinate propagation. Smoothing splines are used to yield an initial coordinate for each new data point, according to their local geometrical relations. Experimental results illustrate the validity of our approach.

1 Introduction

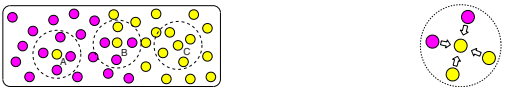
Recently, some manifold learning algorithms have been proposed for nonlinear dimensionality reduction (NLDR). Typical algorithms include Isomap [1], local linear embedding (LLE) [2], Laplacian eigenmap [3], local tangent space alignment (LTSA) [4], charting [5], Hessian LLE (HLLE) [6], semi-definite embedding [7], conformal eigenmap [8], spline embedding (SE) [9], etc. Real performances on many data sets show that they are effective methods to discover the underlying structure hidden in the high-dimensional data set.

All the manifold learning algorithms are initially developed to obtain a low-dimensional embedding for a set of given data points. The problem in general is formulated as an optimization problem, in which the low-dimensional coordinates of all the given data points need to be solved. Matrix eigen-decomposition or other mathematical optimization tools (for instance, the semidefinite programming [7]) are used to obtain the final results, i.e., the intrinsic embedding coordinates. Accordingly, the algorithms run in a batch mode, once for all the input data points. When new data points arrive, one needs to rerun the algorithm with all data points. In applications, however, rerunning the algorithm may become impractical as more and more data points are collected sequentially. On

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 332–343, 2007.

the other hand, *rerunning* means the previous results are simply discarded. This may be very wasteful in computation.

Our interest here is to embed the new data points into the previously-learned results. This is also known as out-of-sample problem. In literature, out-of-sample extensions for LLE, Isomap, Laplacian Eigenmap are given by Bengio et al., using kernel tricks [10]. This problem is further formulated as an incremental learning problem, and extensions for LLE and Isomap are given by several researchers [11,12]. These approaches are suitable for once dealing with one new data point. For more than one new data points, however, we need to embed them one by one.



(a) (b)

Fig. 1. (a) The task of embedding new data points; (b) coordinate propagation in a neighborhood

Let us use Fig. 1 to explain our motivation. We are given two sets of data points which are well sampled from a manifold embedded in a high-dimensional Euclidean space. The low-dimensional embedding coordinates of one data set are also given, saying in Fig. 1(a), the data points with (dark) purple color. Under these conditions, our task is to embed the new data points (yellow points). In this work setting, the coordinates of the neighbors of a new data point may be known, partly known or even all unknown (Fig. 1(a)).

Our idea to solve this problem is to propagate the known coordinates to the new data points. To this end, we first consider this problem in view of coordinate reconstruction in the intrinsic space and formulate it as a quadratic programming. In this way, we can get a global embedding for the new data points. Then, we develop an iterative algorithm through a regularization framework. Through iterations, each new data point gradually obtains a coordinate (Fig. 1(b)). We call this process *coordinate propagation*. We also use smoothing splines to generate an initial coordinate for each new data point to speed up the precess.

2 Model and Algorithm

2.1 Problem Formulation

The NLDR problem. Given a set of data points $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^m$, which lie on a manifold M embedded in a m-dimensional Euclidean space. The goal is to invert an underlying generative model $\mathbf{x} = f(\mathbf{y})$ to find their low-dimensional parameters (embedding coordinates) $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n\} \subset \mathbb{R}^d$ with $d < m$. In this form, NLDR is also known as manifold learning.

The out-of-sample extension problem. Given a set of data points $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_l, \mathbf{x}_{l+1}, \cdots, \mathbf{x}_n\} \subset \mathbb{R}^m$ and the low-dimensional embedding coordinates $\mathcal{Y}_L = \{\mathbf{y}_1^0, \cdots, \mathbf{y}_l^0\} \subset \mathbb{R}^d$ learned from the first l data points. The goal is to obtain the low-dimensional coordinates $\mathcal{Y}_U = \{\mathbf{y}_{l+1}, \cdots, \mathbf{y}_n\}$ of the rest $n-l$ data points, according to their relevances to the first l data points.

2.2 Model

A large family of nonlinear manifold learning algorithms can be viewed as the approaches based on minimizing the low-dimensional coordinate reconstruction error. The algorithms in this family include LLE, Laplacian eigenmap, LTSA, and spline embedding (SE). The optimization problem can be uniformly formulated as follows:

$$\begin{array}{ll} min & tr(Y^T M Y) \\ s.t. & Y^T C Y = I \end{array} \tag{1}$$

where tr is a trace operator, M is a $n \times n$ matrix which is calculated according to the corresponding geometrical preserving criterion, C is a $n \times n$ matrix used to constrain Y to avoid degenerate solutions, and $Y = [\mathbf{y}_1, \cdots, \mathbf{y}_n]^T$ is a $n \times d$ matrix to be solved, in which $\mathbf{y}_i$ is a d-dimensional embedding coordinate of $\mathbf{x}_i$ $(i = 1, \cdots, n)$. That is, each row of Y corresponds to a low-dimensional coordinate of a data point in $\mathcal{X}$.

Specifically, in LLE, $M = (I-W)^T(I-W)$; in Laplacian eigenmap, $M = D - W$; in LTSA, $M = S^T W^T W S$, and in SE, $M = S^T B S$. In these algorithms, C is a $n \times n$ identity matrix. Problem (1) can be easily solved via matrix eigendecomposition.

Now we use problem (1) to solve the out-of-sample problem. Introducing the known low-dimensional embedding coordinates of the first l data points in $\mathcal{X}$, naturally we can obtain a linearly constrained optimization problem:

$$\begin{array}{ll} min & tr(Y^T M Y) \\ s.t. & \mathbf{y}_i = \mathbf{y}_i^0, \quad i = 1, 2, \cdots, l \end{array} \tag{2}$$

It seems that problem (2) is very complex since the variable to be optimized is a matrix which has $d \times n$ unknown components. Directly solving it may be very expensive due to different constraints from problem (1).

Now we rewrite Y in terms of column vectors, and denote it by $Y = [\tilde{\mathbf{y}}_1, \cdots, \tilde{\mathbf{y}}_d]$, in which $\tilde{\mathbf{y}}_i \in \mathbb{R}^n$ $(i = 1, \cdots, d)$ is the i-th coordinate component vector of all the n data points. Then

$$tr(Y^T M Y) = \sum\nolimits_{i=1}^{d} \tilde{\mathbf{y}}_i^T M \tilde{\mathbf{y}}_i \tag{3}$$

We can see that $\tilde{\mathbf{y}}_1, \cdots, \tilde{\mathbf{y}}_d$ are decoupled from each other in Eq. (3). We further write out the d coordinate components of $\mathbf{y}_i^0$ and let $\mathbf{y}_i^0 = [f_i^0(1), \cdots, f_i^0(d)]^T$, $i = 1, \cdots, l$. Based on the Lagrange multiplier method, problem (2) can be converted, equivalently, into the following d subproblems, each of which is used to optimize a coordinate component vector:

$$\begin{cases} min & \mathbf{y}^T M \mathbf{y} \\ s.t. & y_i = f_i^0(1), \quad i = 1, 2, \cdots, l \\ \cdots & \\ min & \mathbf{y}^T M \mathbf{y} \\ s.t. & y_i = f_i^0(d), \quad i = 1, 2, \cdots, l \end{cases} \tag{4}$$

where $\mathbf{y} = [y_1, \cdots, y_n]^T$ is a n-dimensional vector to be solved. Note that each subproblem is a convex quadratic programming (QP) since M is a positive semidefinite matrix[1]. Therefore, we can easily solve them.

To reduce the number of variables to be solved, we write M as follows [13]:

$$M = \begin{pmatrix} M_{ll} & M_{lu} \\ M_{ul} & M_{uu} \end{pmatrix} \tag{5}$$

where M_{ll} is a $l \times l$ sub-block, M_{lu} is a $l \times (n-l)$ sub-block, M_{ul} is a $(n-l) \times l$ sub-block, and M_{uu} is a $(n-l) \times (n-l)$ sub-block. Let $\mathbf{y}_l = [y_1, \cdots, y_l]^T$ and $\mathbf{y}_u = [y_{l+1}, \cdots, y_n]^T$. Then

$$\mathbf{y}^T M \mathbf{y} = \mathbf{y}_u^T \cdot M_{uu} \cdot \mathbf{y}_u + \mathbf{y}_l^T (M_{lu} + M_{ul}^T) \mathbf{y}_u + \mathbf{y}_l^T \cdot M_{ll} \cdot \mathbf{y}_l$$

Note that $\mathbf{y}_l$ is known in each subproblem. Substituting it into the corresponding objective function, problem (4) can be further reduced to the following QP problems:

$$\begin{cases} min & \mathbf{y}_u^T \cdot M_{uu} \cdot \mathbf{y}_u + \mathbf{h}_1^T \cdot \mathbf{y}_u \\ \cdots & \\ min & \mathbf{y}_u^T \cdot M_{uu} \cdot \mathbf{y}_u + \mathbf{h}_d^T \cdot \mathbf{y}_u \end{cases} \tag{6}$$

where $\mathbf{h}_i$ $(\in \mathbb{R}^{n-l}, i = 1, \cdots, d)$ is calculated according to $(M_{ul} + M_{lu}^T)\mathbf{y}_l$. Now each QP subproblem in (6) has only $n-l$ variables to be solved. Meanwhile, M_{uu} is also positive semidefinition[2]. Thus, each QP in (6) is a convex QP, which has a global optimum.

Finally, we can combine the d global optima of problem (6) together into $n-l$ d-dimensional coordinates. In this way, we achieve a low-dimensional global embedding for $n-l$ new data points.

2.3 Iterative Algorithm for Coordinate Propagation

In this subsection, we develop an iterative algorithm for solving the out-of-sample extension problem. The iterative algorithm can reduce the computational complexity and need much less computation resources. In an iterative framework, it would be possible for us to embed a very large number of new data points.

[1] This can be easily justified in LLE, LTSA, and SE. In Laplacian eigenmap, M is a Laplacian matrix, which is also positive semidefinition. Actually, for any vector $\mathbf{x} = [x_1, \cdots, x_n]^T \in \mathbb{R}^n$, $\mathbf{x}^T M \mathbf{x} = \mathbf{x}^T (D - W)\mathbf{x} = \frac{1}{2} \sum_{i,j} (x_i - x_j)^2 w_{ij}$. Since each component w_{ij} in W is nonnegative, then $\mathbf{x}^T M \mathbf{x} \geq 0$.

[2] For any $\mathbf{x} = [0, \cdots, 0, x_{l+1}, \cdots, x_n]^T$, since M is positive semidefinition, we have $\mathbf{x}^T M \mathbf{x} = [x_{l+1}, \cdots, x_n] \cdot M_{uu} \cdot [x_{l+1}, \cdots, x_n]^T \geq 0$. This indicates that M_{uu} is also positive semidefinition.

Here we consider one of the subproblems in problem (4) since they have the same form. For convenience, we omit the superscripts and the subscripts in the constraints, and rewrite the problem as follows:

$$\begin{array}{ll} min & \mathbf{y}^T M \mathbf{y} \\ s.t. & y_i = f_i, \quad i = 1, 2, \cdots, l \end{array} \tag{7}$$

Converting the hard constraints in (7) into soft constraints and introducing a predicting term for the new data points, we have the following regularization representation:

$$min \quad \mathbf{y}^T M \mathbf{y} + \mu_1 \sum\nolimits_{i=1}^{l} (y_i - f_i)^2 + \mu_2 \sum\nolimits_{i=l+1}^{n} (y_i - g_i)^2 \tag{8}$$

where $\mu_1 > 0$ and $\mu_2 > 0$ are regularization parameters, and g_i is a predicted value for y_i, $i = l+1, \cdots, n$. Here, g_i will be evaluated from the neighbors of $\mathbf{x}_i$. We use spline interpolation to solve this problem (in Section 3).

In (8), the first term is the smoothness constraint, which means that the embedding coordinates should not change too much between neighboring data points. The second term is the fitting constraint, which means that the estimated function y should not change too much from the given values. The third term is the predicting constraint, which means that function y should not bias too much from the predicted values. The trade-off among these three constraints are stipulated by μ_1 and μ_2. We can see that the second term is equivalent to the hard constraints in (7) in case of $\mu_1 \to \infty$.

Let us assume for the moment that each g_i is known. Differentiating the objective function with respect to $\mathbf{y} \stackrel{\wedge}{=} [\mathbf{y}_l^T, \mathbf{y}_u^T]^T$, we have

$$\begin{cases} M_{ll}\mathbf{y}_l + \mu_1 \mathbf{y}_l + M_{lu}\mathbf{y}_u - \mu_1 \mathbf{f}_l = 0 \\ M_{uu}\mathbf{y}_u + \mu_2 \mathbf{y}_u + M_{ul}\mathbf{y}_l - \mu_2 \mathbf{g}_u = 0 \end{cases} \tag{9}$$

where $\mathbf{f}_l = [f_1, \cdots, f_l]^T \in \mathbb{R}^l$ and $\mathbf{g}_u = [g_{l+1}, \cdots, g_n]^T \in \mathbb{R}^{n-l}$. For an out-of-sample extension problem, we need not solve $\mathbf{y}_l$ since $\mathbf{y}_l = \mathbf{f}_l$ is known. Thus we only need to consider $\mathbf{y}_u$. Here we can see that the second equation in (9) is equivalent to the subproblem in (6) when $\mu_2 = 0$. Now it can be transformed into

$$\mathbf{y}_u = \frac{1}{1+\mu_2} S \mathbf{y}_u - \frac{1}{1+\mu_2} M_{ul} \mathbf{f}_l + \frac{\mu_2}{1+\mu_2} \mathbf{g}_u$$

where $S = I - M_{uu}$ and I is a $(n-l) \times (n-l)$ identity matrix. Let us introduce two new variables: $\alpha = \frac{1}{1+\mu_2}$ and $\beta = 1 - \alpha$. Then we can get an iteration equation:

$$\mathbf{y}_u^{(t+1)} = \alpha S \mathbf{y}_u^{(t)} - \alpha M_{ul} \mathbf{f}_l + \beta \mathbf{g}_u \tag{10}$$

Some Remarks. (1). According to the theories of linear algebra, the sequence $\{\mathbf{y}_u^{(t)}\}$ generated by Eq. (10) is convergent if and only if the spectral radius of αS is less than one, i.e., $\rho(\alpha S) < 1$. Note that the spectral radius of a matrix is less than any kinds of operator norms[3]. Here we can take $\alpha = 1/(||S||_1 + 1)$. Thus

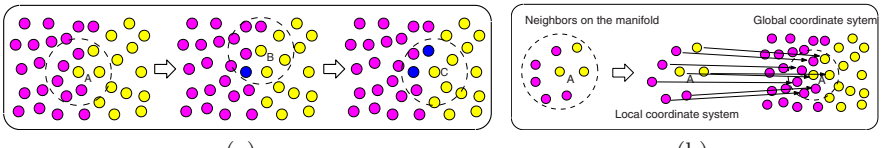
[3] $||A||_1 = \max\limits_j \{\sum_{i=1}^m |a_{ij}|, j = 1, \cdots, n\}$, $||A||_\infty = \max\limits_i \{\sum_{j=1}^n |a_{ij}|, i = 1, \cdots, m\}$, etc.

$\rho(\alpha S) < 1$. (2). In Eq. (10), the first term is the contribution from the new data points, while the second term is the contribution from the previously learned data points on the manifold. Note that these contributions are decreased since $\alpha < 1$ holds. If we only consider these two terms, then the sequence will converge to $\mathbf{y}_u^* = -\alpha(I - \alpha S)^{-1} M_{ul} f_l$. This may be far from the optimization optimum $-M_{uu}^{-1} M_{ul} f_l$, especially when α is a small number. To avoid attenuations, we introduce a compensation term, i.e., the third term in Eq. (10). Here we call it a prediction to the new data points in the iterative framework, and its contribution to $\mathbf{y}_u$ is stipulated by the positive parameter β, which is a parameter in $(0, 1)$. Therefore, it is necessary for us to get a good prediction to $\mathbf{g}_u$. We will evaluate it along the manifold via spline interpolation on the neighbors.

The steps of the iterative algorithm can be summarized as follows:

(1) Provide an initial coordinate component vector $\mathbf{g}_u^{(0)}$ (in Section 3); Let $i = 0$.
(2) Let $\mathbf{g}_u$=$\mathbf{g}_u^{(i)}$ and run the iteration equation (10) to obtain a $\mathbf{y}_u^*$.
(3) Justify if it is convergence.
(4) Predict a $\mathbf{g}_u^{(i+1)}$ according to $\mathbf{y}_u^*$ and $\mathbf{f}_l$ (in Section 3).
(5) $i = i + 1$, go to step (2).

Through the iterations, each new data point gradually receives a value (here it is a coordinate component). To construct d coordinate component vectors, we need to perform the iterative algorithm d times. In this way, the known coordinates are finally propagated to the new data points.



(a) (b)

Fig. 2. Coordinate propagation. (a) The first three steps of coordinate prediction; (b) Two steps of mapping the neighbors on the manifold to the global coordinate system.

In computation, we set the maximum iteration times to 100 when performing the iteration Equation (10). Now a task to be solved is to provide an initial $\mathbf{g}_u^{(0)}$ and generate a $\mathbf{g}_u^{(i+1)}$ in step (4). Details will be introduced in Section 3.

3 Predicting Coordinates Via Smoothing Splines

We first discuss how to provide a $\mathbf{g}_u^{(0)}$. Fig. 2(a) is used to explain our idea. There we first select to predict the new data point "A" since it has the maximum number of neighbors with known coordinates. After "A" is treated, then we select "B". After "A" and "B" have been treated, "C" is one of the candidates in next time.

In the above process, a basic task can be summarized as follows. Given a new data point $\mathbf{x} \in \mathbb{R}^m$ and its k neighbors $\{\mathbf{x}_1, \cdots, \mathbf{x}_r, \mathbf{x}_{r+1}, \cdots, \mathbf{x}_k\}$. We assume that the low-dimensional coordinates of the first r data points are known. The task is to generate a coordinate for the center point $\mathbf{x}$.

Our method includes two steps (Fig. 2(b)). (1). Construct a local coordinate system to represent $\mathbf{x}$ and its k neighbors, and calculate $r+1$ local coordinates $\mathbf{t}, \mathbf{t}_1, \cdots, \mathbf{t}_r \in \mathbb{R}^d$. (2). Construct a function $g : \mathbb{R}^d \to \mathbb{R}$ through which we can get a value $f = g(\mathbf{t})$ for $\mathbf{x}$. Here f can be considered as a low-dimensional coordinate component. Furthermore, g should meet the following conditions:

$$f_j = g(\mathbf{t}_j), \quad j = 1, 2, \cdots, r \tag{11}$$

where f_j is a known coordinate component of $\mathbf{x}_j$ $(j = 1, \cdots, r)$. Actually, we use g to map the local coordinate $\mathbf{t}$ of $\mathbf{x}$ into the global coordinate system with lower dimensionality, in which the original data points are represented.

Suppose that the n data points in $\mathcal{X}$ are densely sampled from the manifold, then the tangent space of the manifold M at $\mathbf{x} \in \mathcal{X}$ can be well estimated from the neighbors of $\mathbf{x}$ [14]. We use this subspace to define the local coordinates [4,6]. To be robustness, we simultaneously coordinatize the $k+1$ data points. Note that the computation for the rest data points $\mathbf{x}_{r+1}, \cdots, \mathbf{x}_k$ is also necessary for us to generate $\mathbf{g}_u^{(i+1)}$. After the local coordinates are evaluated, then we need to map them into the global coordinate system. Fig. 2(b) shows the above process.

To satisfy the conditions in (11), spline regression method is used to construct the function g. The spline we use is developed from the Sobolev space, and has the following form [15,16]:

$$g(\mathbf{t}) = \sum\nolimits_{j=1}^{r} \alpha_j \, \phi_j(\mathbf{t}) + \sum\nolimits_{i=1}^{p} \beta_i \, p_i(\mathbf{t}) \tag{12}$$

where the first term is a linear combination of r Green's functions $\phi_j(\mathbf{t})$ $(j = 1, \cdots, r)$, and the second term is a polynomial in which all $p_i(\mathbf{t})$, $i = 1, \cdots, p$, constitute a base of a polynomial space. Here we take the one-degree polynomial space as an example to explain $p_i(\mathbf{t})$. Let $\mathbf{t} = [t_1, t_2]^T$ in the case of $d = 2$, we have $p_1(\mathbf{t}) = 1$, $p_2(\mathbf{t}) = t_1$ and $p_3(\mathbf{t}) = t_2$. In this case, p is equal to 3.

In addition, the Green's function $\phi_j(\mathbf{t})$ is a general radical basis function [15,16]. For instances, in the case of $d = 2$, $\phi_j(\mathbf{t}) = (||\mathbf{t} - \mathbf{t}_j||)^2 \cdot \log(||\mathbf{t} - \mathbf{t}_j||)$; in the case of $d = 3$, $\phi_j(\mathbf{t}) = ||\mathbf{t} - \mathbf{t}_j||$.

To avoid degeneracy, we add the following conditionally positive definition constraints [17]:

$$\sum\nolimits_{j=1}^{r} \alpha_j \cdot p_i(\mathbf{t}_j) = 0, \quad i = 1, \cdots, p \tag{13}$$

Now substituting the interpolation conditions (11) into Eq. (12) and Eq. (13), we can get a linear system for solving the coefficients $\alpha \in \mathbb{R}^r$ and $\beta \in \mathbb{R}^p$:

$$\begin{pmatrix} K & P \\ P^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix} \tag{14}$$

where K is a $r \times r$ symmetrical matrix with elements $K_{ij} = \phi(||\mathbf{t}_i - \mathbf{t}_j||)$, P is a $r \times p$ matrix with elements $P_{ij} = p_i(\mathbf{t}_j)$, and $\mathbf{f} = [f_1, \cdots, f_r]^T \in \mathbb{R}^r$.

We point out that g is a smooth function and $f_j = g(\mathbf{t}_j)$ holds for all j, $j = 1, \cdots, r$. Faithfully satisfying the given conditions in (11) is necessary for us to rely on it to interpolate a new point $\mathbf{x}$.

To avoid error accumulation, the above performance is only used to yield a coordinate component for the center point $\mathbf{x}$ and not used to map the rest new data points $\mathbf{x}_{r+1}, \cdots, \mathbf{x}_k$. To get the d coordinate components of $\mathbf{x}$, we need to construct d splines. That is, we need to solve Eq. (14) d times. Note that in each time the coefficient matrix keeps unchanged since it is only related to the r local coordinates $\mathbf{t}_j$ $(j = 1, \cdots, r)$. Finally, according to the steps as illustrated in Fig. 2, each new data point can get an initial coordinate.

To predict a new vector $\mathbf{g}_u^{(i+1)}$ during the iterations, we only need to set $r = k$ and perform the above algorithm again for each new data point.

(a) (b)

Fig. 3. (a): The 1200 data points sampled from a S-surface; (b): The intrinsic structure of the data points in (a), i.e., a 2-dimensional rectangle

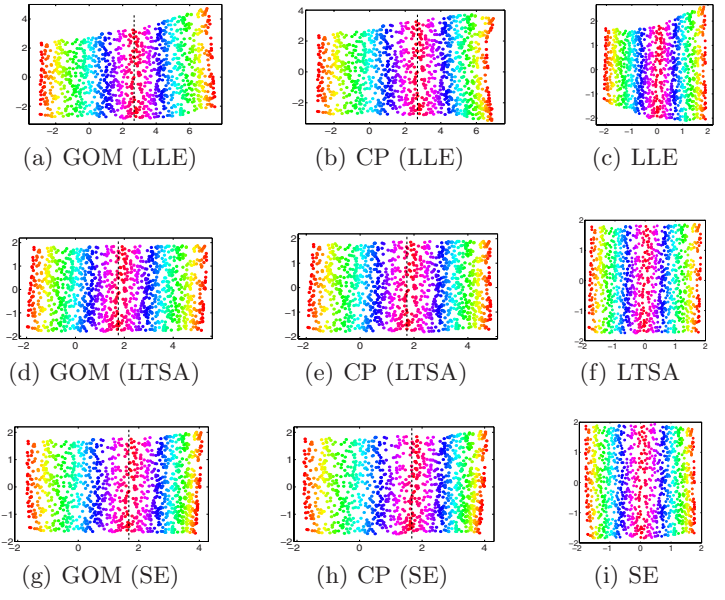
4 Experimental Results

We evaluated the algorithm on several data sets including toy data points and real-world images. Here we give some results obtained by the global optimization model (GOM) and the coordinate propagation (CP). These experiments can give us a straightforward explanation on the data and the learned results. In addition, the computation complexity is also analyzed in this section.

Fig. 3(a) illustrates 1200 data points sampled from a S-surface. Among these data points, 600 data points below the horizontal plane "p" are treated as original data points, and the rest 600 data points above the plane "p" are treated as new data points. The intrinsic manifold shape hidden in these data points is a rectangle (Fig. 3(b)). The sub-rectangle located in the left of the center line in Fig. 3(b) can be viewed as the 2-dimensional (2D) parameter domain of the original data pints. Our goal is to use the new data points to extend it to the right sub-rectangle.

We use LLE, LTSA and SE to learn the original data points. The results with $k = 12$ nearest neighbors are shown in the left region of the dash line in Fig. 4(a)/4(b), Fig. 4(d)/4(e), and Fig 4(g)/4(h), resepctively. Naturally, the learned structure is only a part of the whole manifold shape. The intrinsic manifold structure hidden in these 600 original data points is a small rectangle.

The new data points are embedded into the right region by GOM and CP. In Fig. 4(a) and Fig. 4(b), the M matrix in Eq. (5) is calculated according to LLE with $k = 12$, i.e., $M = (I - W)^T(I - W)$. In Fig. 4(d) and Fig. 4(e), the M matrix is calculated according to LTSA with $k = 12$, i.e., $M = S^T W^T W S$; In



(a) GOM (LLE) (b) CP (LLE) (c) LLE

(d) GOM (LTSA) (e) CP (LTSA) (f) LTSA

(g) GOM (SE) (h) CP (SE) (i) SE

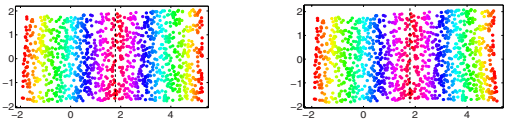
Fig. 4. The 2-dimensional embedding results of 1200 data points as illustrated in Fig. 3(a). The center lines are drawn manually.

Fig 4(g) and Fig. 4(h), the M matrix is calculated according to SE with $k = 12$, i.e., $M = S^T BS$. From the ranges of the learned coordinates, we can see that the learned manifold shape is extended along the right direction by the new data points.

For comparison, Fig. 4(c), Fig. 4(f) and Fig. 4(i) show the 2D embedding results of all the 1200 data points directly by LLE, LTSA and SE. As can be seen, the results are confined into a square region, not extended to be a long rectangle, which is the real low-dimensional structure hidden in the data points.

Fig. 5 shows the results by GOM and CP, which use a combination of SE and LLE. That is, the original data points are learned by SE to get their low-dimensional coordinates, but the M matrix in Eq. (5) is calculated via LLE. Compared with the results purely based on LLE (see Fig. 4(a) and Fig. 4(b)), here the shape of the manifold is better preserved.

Fig. 6 shows two experiments on image data points. In Fig. 6(a) and Fig. 6(b), a face moves on a noised background image from the top-left corner to the bottom-right corner. The data set includes 441 images, each of which includes 116×106 grayscale pixels. The manifold is embedded in $\mathbb{R}^{12296}$. Among these images, 221 images are first learned by SE with $k = 8$. The learned results are shown with (red) filled squares. The rest data points are treated as new data points. From Fig. 6(a) and Fig. 6(b), we can see that they are faithfully embedded into the previously learned structure by GOM and CP. Here, the M matrix in Eq. (5) is calculated according to SE.



(a) GOM (b) CP

Fig. 5. The embedding results by GOM and CP. The original data points are learned by SE, while the M matrix in Eq. (5) for GOM and CP is calculated via LLE.

(a) GOM (b) CP (c) GOM (d) CP

Fig. 6. The embedding results by GOM and CP on two image data sets. The original data points are learned by SE. The M matrix in Eq. (5) is calculated according to SE. Representative images of some new data points are shown at the side of the corresponding circle points.

In Fig. 6(c) and Fig. 6(d), 400 color images are treated, which are taken from a teapot via different viewpoints aligning in a circle. The size of the images is 76×101. The manifold is embedded in $\mathbb{R}^{23028}$. Among the images, 200 images are first learned by SE with $k = 5$, and the results are illustrated with (red)filled circles. The rest 200 images are treated as new data points. They are embedded into the right positions by GOM and CP, using SE to calculate the M matrix in Eq. (5).

Computation Complexity. Both GOM and CP require to calculate the M matrix in Eq. (5). Differently, in GOM we need to solve d QP problems. The computation complexity is $O(d(n-l)^3)$. In CP, we need to perform the singular value decompositions (SVD) of $n-l$ matrices in $\mathbb{R}^{(k+1)\times(k+1)}$ when computing $k+1$ local coordinates in tangent space [4,6] for each of $n-l$ new data points. We also need to solve $d \times (n-l)$ linear systems formulated as Eq. (14). The computation complexity of SVD is $O((k+1)^3)$, while that of the linear system is near to $O((k+d+1)^2)$ (using Gauss-Seidel iteration). In addition, the computation complexity in Eq. (12) is near to $O(k+d+1)$. Thus, totally the complexity in each a coordinate propagation is about $O((n-l)[(k+1)^3) + (k+d+1)^2 + k + d + 1])$. Compared with $O(d(n-l)^3)$ in GOM method, the computation complexity in CP is only linear to the number of new data points.

In most experiments, the real performance of CP is convergent when iteration counter i is equal to one. That is, we only need to provide $\mathbf{g}_u^{(0)}$ and $\mathbf{g}_u^{(1)}$ once and the convergence is achieved during iterating Eq. (10). This is reasonable since the data points are assumed to be well sampled in manifold learning. Thus we can get a good prediction to each new data point along the manifold via spine interpolation.

5 Related Work on Semi-supervised Manifold Learning

A parallel work related to out-of-sample extension for manifold learning is the semi-supervised manifold learning [13,18,19]. In a semi-supervised framework, the coordinates of some landmark points are provided to constrain the shape to be learned. The landmark points are usually provided according to prior knowledge about the manifold shape or simply given by hand. The goal is to obtain good embedding results via a small number of landmark points. In generally, it is formulated as a transductive learning problem. Intrinsically, the corresponding algorithm runs in a batch mode. In contrast, out-of-sample extension starts with a known manifold shape which is learned from the original data points, and focuses on how to embed the new data points, saying, in a dynamic setting which are collected sequentially. During embedding, the coordinates of previously learned data points can maintain unchanged.

6 Conclusion

We have introduced an approach to out-of-sample extension for NLDR. We developed the global optimization model and gave the coordinate propagation algorithm. Promising experimental results have been presented for 3D surface data and high-dimensional image data, demonstrating that the framework has the potential to embed effectively new data points to the previously learned manifold, and has the potential to use the new points to extend an incomplete manifold shape to a full manifold shape. In the future, we would like to automatically introduce the edge information about the manifold shape from the data points so as to obtain a low-dimensional embedding with better shape-preserving.

Acknowledgements

This work is supported by the Projection (60475001) of the National Nature Science Foundation of China. The anonymous reviewers have helped to improve the representation of this paper.

References

1. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science. **290** (2000) 2319-2323
2. Roweis, S.T., Saul, L. K.: Nonlinear dimensionality reduction by locally linear embedding. Science. **290** (2000) 2323-2326
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation. **15**(6) (2003), 1373-1396
4. Zhang, Z.Y., Zha, H. Y.: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM Journal on Scientific Computing. **26**(1) (2004) 313-338

5. Brand, M.: Charting a manifold. Advances in Neural Information Processing Systems 15. Cambridge, MA: MIT Press, (2003) 985-992
6. Donoho, D.L., Grimes, C. E.: Hessian eigenmaps: locally linear embedding techniques for highdimensional data. Proceedings of the National Academy of Arts and Sciences. **100** (2003) 5591-5596
7. Weinberger, K.Q., Sha, F., Saul, L.K.: Learning a kernel matrix for nonlinear dimensionality reduction. Proceedings of International Conference on Machine learning. Banff, Canada (2004) 888-905
8. Sha, F., Saul L. K.: Analysis and extension of spectral methods for nonlinear dimensionality reduction. International Conference on Machine learning. Bonn, Germany (2005) 784-791
9. Xiang, S.M., Nie, F.P., Zhang, C. S., Zhang, C.X.: Spline embedding for nonlinear dimensionality reduction. European conference on Machine Learning, Berlin, Germany (2006) 825-832
10. Bengio, Y., Paiement, J., Vincent, P.: Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps and spectral clustering. Advances in Neural Information Processing Systems 16. Cambridge, MA: MIT Press (2004).
11. Law, M., Jain, A.K.: Incremental nonlinear dimensionality reduction by manifold learning. IEEE Transactions on Pattern Analysis and Machine Intelligence. **28**(3) (2006) 377-391
12. Kouropteva, O., Okun, O., Pietikäinen, M.: Incremental locally linear embedding. *Pattern Recognition*, **38**(10) (2005) 1764-1767
13. Yang, X., Fu, H.Y., Zha, H.Y., Barlow, J.: Semi-supervised dimensionality reduction. International Conference on Machine Learning. Pittsburgh, USA. (2006)
14. Min, W., Lu, K., He, X. F.: Locality pursuit embedding. Pattern recognition. **37**(4) (2004) 781-788
15. Duchon, J.: Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: Schempp, W., Zeller, K. (eds): Constructive Theory of Functions of Several Variables. Berlin: Springer-Verlag. (1977) 85-100
16. Wahba, G.: Spline models for observsatonal data. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM Press. (1990)
17. Yoon, J.: Spectral approximation orders of radial basis function interpolation on the Sobolev space. SIAM Journal on Mathematical Analysis, **33**(4):946-958
18. Ham, J., Lee, L., Saul, L.: Semisupervised alignment of manifolds International Workshop on Artificial Intelligence and Statistics. Barbados, West Indies (2004) 120-127
19. Gong, H.F., Pan, C.H., Yang, Q., Lu, H.Q., Ma, S.D.: A Semi-supervised framework for mapping data to the intrinsic manifold. International Conference on Computer Vision. Beijing, China (2005) 98-105

Spectral Clustering Based Null Space Linear Discriminant Analysis (SNLDA)

Wenxin Yang[1] and Junping Zhang[2,3,⋆]

[1] Department of Electronic Engineering
Fudan University, Shanghai 200433, China
soloistyang@gmail.com
[2] Shanghai Key Lab. of Intelligent Information Processing
Department of Computer Science and Engineering
Fudan University, Shanghai 200433, China
jpzhang@fudan.edu.cn
[3] State Key Laboratory of Rail Traffic Control and Safety
Beijing Jiaotong University, China

Abstract. While null space based linear discriminant analysis (NLDA) obtains a good discriminant performance, the ability easily suffers from an implicit assumption of Gaussian model with same covariance each class. Meanwhile, mixture model discriminant analysis, which is a good way for processing issues on multiple subclasses in each class, depends on human experience on the number of subclasses and has a highly complex iterative process. Considering the cons and pros of the two mentioned approaches, we therefore propose a new algorithm, called **S**pectral clustering based **N**ull space **L**inear **D**iscriminant **A**nalysis (SNLDA). The main contributions of the algorithm include the following three aspects: 1) Employing a new spectral clustering method which can automatically detect the number of clusters in each class. 2) Finding a unified null space for processing multi-subclasses issues with eigen-solution technique. 3) Refining the calculation of the covariance matrix in a single sample subclass. The experimental results show the promising of the proposed SNLDA algorithm.

1 Introduction

A large number of subspace methods have been proposed for processing high dimensional data in last decades. Among these methods, Principal Component Analysis (henceforth PCA), which is to find an optimal set of projection directions in the sample space and maximize the covariance of the total scatter across all samples, has difficulties in solving nonlinear problems. Linear Discriminant Analysis (LDA), which attempts to maximize inter-class distances and minimize intra-class distances simultaneously, always suffers from a small sample size (SSS) problem especially for high dimensional data. Null Space Linear Discriminant Analysis (NLDA) in which the null space of the intra-class scatter matrix $\boldsymbol{S}_W$ is preserved and then projected to the inter-lass scatter matrix $\boldsymbol{S}_B$ [1], can obtain a high classification accuracy than PCA and LDA because

⋆ Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 344–354, 2007.

of saving more discriminant information. However, a disadvantage of the NLDA algorithm as well as LDA, is that each class is implicitly assumed to subject to Gaussian distributions with equal covariance.

On the other hand, mixture models based approaches, such as Multiple Discriminant Analysis (MDA) [2] and Multimodal Oriented Discriminant Analysis (MODA) [3], assume that training samples of each class are generated from a mixture model constituted of multiple Gaussian subclasses with different covariances, and extract discriminant information from the subclasses which are obtained by cluster analysis. While these methods obtain better discriminant performance than the traditional single model ones, there are still some drawbacks among them. Firstly, the number of subclasses of training data needs to be manually assigned. Secondly, the employed iterative algorithm has a high computational complexity and low convergence rate. Thirdly, the multimodal methods suffer by the SSS problem more seriously when the number of training samples is small enough (e.g. $= 2$).

To address the aforementioned issues, we propose the SNLDA algorithm. First of all, a new spectral clustering method proposed by Lihi Z.M. & Pietro P. [5] is introduced for automatical detecting the number of subclasses of each class. Then, covariances from different subclasses are unified for modeling a null space. Finally, principal feature vectors are extracted from the null space with eigen-solution approach. Without the iterative procedure employed by the MDA and MODA, the proposed SNLDA algorithm builds a system with higher recognition performance and less computational complexity. Meanwhile, the SSS problem which cannot be solved by the mentioned multimodal methods is circumvented through the SNLDA algorithm.

The rest of the paper is organized as follows: Section 2 is the details of the proposed SNLDA method. The experiment results are reported in Section 3. In Section 4, we end up this paper with a conclusion.

2 The Proposed SNLDA Method

In this section, a new spectral clustering algorithm that can automatically detect the number of clusters (or subclasses) in each class is first introduced. Then the details of deriving the unified null space will be proposed. Finally, a pseudo-code of the proposed algorithm and some refinements will be given.

2.1 Spectral Clustering

Generally speaking, an underlying distribution of data can be properly approximated by a mixture of Gaussian models with cluster analysis. To achieve this task, the MODA algorithm introduces multiclass spectral clustering [4] which employs an iterative procedure with non-maximum suppression and uses singular value decomposition (SVD) to recover the rotation $\boldsymbol{R}$. However, the method requires the pre-assignment of the number of clusters and easily gets stuck in local minima [4].

In this section, a new spectral clustering method proposed by Lihi Z.M. & Pietro P. [5] is introduced. Assuming that the eigenvector $\boldsymbol{X} \in \Re^{n \times C}$ in an ideal case is polluted by a linear transformation $\boldsymbol{R} \in \Re^{C \times C}$, the method can recover the rotation $\boldsymbol{R}$

through a gradient descent scheme, the corresponding cost function J to be minimized is defined as:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{C} \frac{Z_{ij}^2}{M_i^2} \tag{1}$$

Where $\boldsymbol{Z}$ is the rotated eigenvector, n is the number of points, C means the possible group number, $M_i = \max_j Z_{ij}$, i and j denote the row and the column of matrix $\boldsymbol{Z}$, respectively. As a result, the number of clusters in each class could be automatically estimated without local minimum. It is noticeable that the spectral clustering method can perform quite well even for small sample sizes. The details on the introduction of the method can be seen in [5].

2.2 Deriving the Unified Null Space

After cluster analysis, each class will be automatically divided into several Gaussian clusters. To measure the distance between two different normal distributions $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i^{r_1}, \boldsymbol{\Sigma}_i^{r_1})$ and $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j^{r_2}, \boldsymbol{\Sigma}_j^{r_2})$, the Kullback-Leibler (KL) divergence is first defined as follows [3]:

$$\begin{aligned} D_{KL} &= \int d\boldsymbol{x}(\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i^{r_1}, \boldsymbol{\Sigma}_i^{r_1}) - \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j^{r_2}, \boldsymbol{\Sigma}_j^{r_2})) \log \frac{\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i^{r_1}, \boldsymbol{\Sigma}_i^{r_1})}{\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j^{r_2}, \boldsymbol{\Sigma}_j^{r_2})} \\ &= \boldsymbol{tr}((\boldsymbol{\Sigma}_i^{r_1})^{-1}\boldsymbol{\Sigma}_j^{r_2} + (\boldsymbol{\Sigma}_j^{r_2})^{-1}\boldsymbol{\Sigma}_i^{r_1} - 2\boldsymbol{I}) \\ &\quad + (\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T((\boldsymbol{\Sigma}_i^{r_1})^{-1} + (\boldsymbol{\Sigma}_j^{r_2})^{-1})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2}) \end{aligned} \tag{2}$$

Where $\boldsymbol{x} \in \Re^d$ is the training sample, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote mean and covariance, the superscript and subscript of each symbol indicate the index of cluster and class, respectively. For example, $\boldsymbol{\mu}_i^{r_1}$ and $\boldsymbol{\Sigma}_i^{r_1}$ denote the mean and the covariance of the r_1-th cluster in the i -th class. The symbol "$\boldsymbol{tr}$" denotes the trace of matrix. Our aim is to find a linear transformation $\boldsymbol{B} \in \Re^{d \times k}$ (i.e. normalization factor) so that for all clusters,

$$\boldsymbol{B}^T\boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{B}^T\boldsymbol{\mu}_i^r, \boldsymbol{B}\boldsymbol{\Sigma}_i^r\boldsymbol{B}^T) \quad \forall i, r$$

can maximizes the KL divergence among different clusters under the low dimensional subspace, namely:

$$\begin{aligned} E(\boldsymbol{B}) &= \sum_i \sum_{j \neq i} \sum_{r_1 \in C_i} \sum_{r_2 \in C_j} \boldsymbol{D}_{KL} \\ &\propto \sum_i \sum_{j \neq i} \sum_{r_1 \in C_i} \sum_{r_2 \in C_j} \boldsymbol{tr}((\boldsymbol{B}^T\boldsymbol{\Sigma}_i^{r_1}\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{\Sigma}_j^{r_2}\boldsymbol{B}) \\ &\quad + (\boldsymbol{B}^T\boldsymbol{\Sigma}_j^{r_2}\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{\Sigma}_i^{r_1}\boldsymbol{B})) + (\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T\boldsymbol{B}((\boldsymbol{B}^T\boldsymbol{\Sigma}_i^{r_1}\boldsymbol{B})^{-1} \\ &\quad + (\boldsymbol{B}^T\boldsymbol{\Sigma}_j^{r_2}\boldsymbol{B})^{-1})\boldsymbol{B}^T(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2}) \\ &= \sum_i \sum_{r_1 \in C_i} \boldsymbol{tr}((\boldsymbol{B}^T\boldsymbol{\Sigma}_i^{r_1}\boldsymbol{B})^{-1}(\boldsymbol{B}^T \sum_{j \neq i} \sum_{r_2 \in C_j} ((\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T \\ &\quad + \boldsymbol{\Sigma} j^{r_2})\boldsymbol{B})) \end{aligned}$$

$$= \sum_{i} \sum_{r_1 \in C_i} \boldsymbol{tr}((\boldsymbol{B}^T \boldsymbol{\Sigma}_i^{r_1} \boldsymbol{B})^{-1} (\boldsymbol{B}^T \boldsymbol{A}_i \boldsymbol{B})) \tag{3}$$

$$\boldsymbol{A}_i = \sum_{j \neq i} \sum_{r_2 \in C_j} ((\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T + \boldsymbol{\Sigma}_j^{r_2})$$

It is difficult to directly optimize the energy function in Eq. (3) [3], because second-order type of gradient methods do not scale well for a large size of matrix. As a result, the MDA algorithm applies the EM algorithm in the optimization procedure. De la Torre & Kanade proposed a bound optimization method called Iterative Majorization for monotonic reducing the value of the energy function [3]. However, when using these EM-like iterative algorithms, complexities in time and storage are quite high. Also, if data belong to a single sample cluster, discriminant information will be lost due to the fact that all zero intra-cluster scatter matrix will be produced.

Eq.(3) cannot be solved by an eigen-solution like the traditional LDA because there are many different normalization factors $(\boldsymbol{B}^T \boldsymbol{\Sigma}_i^{r_1} \boldsymbol{B})^{-1}$ [3]. In order to eliminate the influence of the normalization factors, our basic idea is to first project different intra-cluster scatter matrix of each cluster to a unified null space, then get the feature vectors by maximizing the inter-cluster scatter matrix in the null space of the intra-cluster one. For the sake of unifying each intra-cluster scatter matrix, a proposition must be proposed first:

Proposition. *Suppose the training set is composed of a total of C clusters. Let $\boldsymbol{\Sigma}$ denote the sum covariance of all clusters $\boldsymbol{\Sigma} = \sum_{n=1}^{C} \boldsymbol{\Sigma}_n$. If the orthnormal bases $\boldsymbol{B}$ can project the sum covariance $\boldsymbol{\Sigma}$ to its null space, namely, $\boldsymbol{B}^T \boldsymbol{\Sigma} \boldsymbol{B} = 0$ and $(\boldsymbol{B}^T \boldsymbol{B} = \boldsymbol{I})$, it can also project each sub covariance $\boldsymbol{\Sigma}_n$ to its null space, which is $\boldsymbol{B}^T \boldsymbol{\Sigma}_n \boldsymbol{B} = 0$.*

A proof on the proposition can be seen in the appendix. For each covariance matrix $\boldsymbol{\Sigma}_i^r$, which belongs to the r-th cluster in the i-th class, we have

$$\boldsymbol{\Sigma} = \sum_{i} \sum_{r} \boldsymbol{\Sigma}_i^r \tag{4}$$

then we can get the orthonormal bases $\boldsymbol{B}$ which can project $\boldsymbol{\Sigma}$ to its null space

$$\boldsymbol{B}^T \boldsymbol{\Sigma} \boldsymbol{B} = 0 \quad (\boldsymbol{B}^T \boldsymbol{B} = \boldsymbol{I}) \tag{5}$$

Under the mentioned proposition, the orthonormal bases $\boldsymbol{B}$ can also project each covariance $\boldsymbol{\Sigma}_i^r$ to its null space

$$\boldsymbol{B}^T \boldsymbol{\Sigma}_i^r \boldsymbol{B} = 0 \tag{6}$$

Therefore, we can modify the objective function Eq.(3) as

$$\begin{aligned} E(\boldsymbol{B}) &= \sum_{i} \sum_{j \neq i} \sum_{r_1 \in C_i} \sum_{r_2 \in C_j} \boldsymbol{D}_{KL} \\ &\propto \sum_{i} \sum_{r_1 \in C_i} \boldsymbol{tr}((\boldsymbol{B}^T \boldsymbol{\Sigma}_i^{r_1} \boldsymbol{B})^{-1} (\boldsymbol{B}^T (\sum_{j \neq i} \sum_{r_2 \in C_j} (\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T \\ &\quad + \boldsymbol{\Sigma}_j^{r_2}) \boldsymbol{B})) \end{aligned}$$

$$
\begin{aligned}
&= \arg \max_{|\boldsymbol{B}^T \Sigma \boldsymbol{B}=0|} |\sum_i \sum_{r_1 \in C_i} (\boldsymbol{B}^T \sum_{j \neq i} \sum_{r_2 \in C_j} (\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T \boldsymbol{B} \\
&\quad + \boldsymbol{B}^T \boldsymbol{\Sigma}_j^{r_2} \boldsymbol{B})| \\
&= \arg \max_{|\boldsymbol{B}^T \Sigma \boldsymbol{B}=0|} |\sum_i \sum_{r_1 \in C_i} (\boldsymbol{B}^T \sum_{j \neq i} \sum_{r_2 \in C_j} (\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T \boldsymbol{B})| \\
&= \arg \max_{|\boldsymbol{B}^T \Sigma \boldsymbol{B}=0|} |\boldsymbol{B}^T \sum_i \sum_{r_1 \in C_i} \sum_{j \neq i} \sum_{r_2 \in C_j} (\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T \boldsymbol{B}| \\
&= \arg \max_{|\boldsymbol{B}^T \Sigma \boldsymbol{B}=0|} |\boldsymbol{B}^T \boldsymbol{S} \boldsymbol{B}| \qquad (7) \\
\boldsymbol{\Sigma} &= \sum_i \sum_{r_1} \boldsymbol{\Sigma}_i^{r_1} \\
\boldsymbol{S} &= \sum_i \sum_{r_1 \in C_i} \sum_{j \neq i} \sum_{r_2 \in C_j} (\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})(\boldsymbol{\mu}_i^{r_1} - \boldsymbol{\mu}_j^{r_2})^T
\end{aligned}
$$

It is obviously that the different normalization factor $(\boldsymbol{B}^T \boldsymbol{\Sigma}_i^{r_1} \boldsymbol{B})^{-1}$ has been replaced by the unified null space $\boldsymbol{B}^T \boldsymbol{\Sigma} \boldsymbol{B} = 0$, and the orthonormal bases $\boldsymbol{B}$ can minimize each covariance $\boldsymbol{\Sigma}_i^r$ onto zero and avoid the numerical influence of the ratio with Eq. (7). Meanwhile, an eigen-solution way can be applied for optimizing the energy function instead of the EM-like algorithms. Therefore, the proposed method has less computational complexity than MDA and MODA which use iterative optimization strategy.

2.3 Further Refinements and a Pseudo-code of the SNLDA Algorithm

If a cluster consists of only one sample which is often happened in many databases (e.g. face recognition), all the elements in the intra-cluster scatter matrix will be equal to zero. Hence a modification in the definition of cluster covariance is proposed:

$$\boldsymbol{\Sigma}_i^r = (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \qquad (8)$$

where $\boldsymbol{x}$ is the only data point of the r-th cluster in the i-th class, $\boldsymbol{\mu}_i$ indicates the mean of the i-th class. Through the replacement, the rank of the modified intra-cluster covariance will not become 0 but 1, and discriminant information can be preserved. It should be pointed out that, because the spectral clustering demands the number of samples in each class is not less than 2, we do not take the instance of a single training sample per class into account in this paper.

Furthermore, some of the other literatures still mention that the null space of the inter-class matrix is no use for discriminant analysis [6], and therefore project the observation space to the null space of the intra-class scatter matrix. Under this conception, we redefine the total inter-cluster covariance $\boldsymbol{S}$ as $\boldsymbol{S}_{raw}$. The details of the SNLDA algorithm are described in Tab.1.

It also should be mentioned that, while the proposed unified null space removes the different covariances components of each cluster, the SNLDA algorithm can still solve the issue of non-Gaussian covariance. We have designed a simulated database to

Table 1. A Pseudo-Code of The SNLDA Algorithm

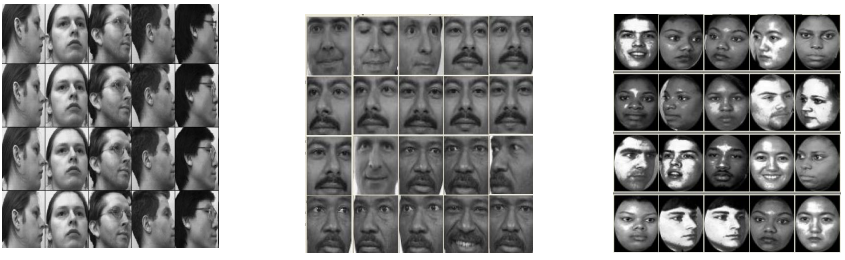
1 . Performing spectral clustering for each class by the proposed SNLDA method.
2 . Calculating the covariance $\mathbf{\Sigma_i^r}$ of each cluster, then summarizing all the covariance matrices according to the following eqaution.
$\mathbf{\Sigma} = \sum_{\mathbf{i}} \sum_{\mathbf{r}} \mathbf{\Sigma_i^r}$
3 . Finding the orthonormal bases $\boldsymbol{B}_{null}$ which project $\mathbf{\Sigma}$ onto its null space.
$(\boldsymbol{B}_{null})^T \mathbf{\Sigma} \boldsymbol{B}_{null} = 0 \quad ((\boldsymbol{B}_{null})^T \boldsymbol{B}_{null} = I)$
4 . Calculating the inter-cluster covariance matrix $\boldsymbol{S}$ as follows:
$\boldsymbol{S} = \sum_i \sum_{r_1 \in C_i} \sum_{j \neq i} \sum_{r_2 \in C_j} (\mu_i^{r_1} - \mu_j^{r_2})(\mu_i^{r_1} - \mu_j^{r_2})^T$
5 . Finding the observation space $\boldsymbol{S}_{raw}$ by keeping the non-zero eigenvalues and eigenvectors in $\boldsymbol{S}$'s eigen-decomposition.
$[\boldsymbol{U}, \boldsymbol{V}] = eig(\boldsymbol{S})$
$\boldsymbol{U}_{raw} \in \boldsymbol{U}^{d \times k}, \boldsymbol{V}_{raw} \in \boldsymbol{V}^{k \times k}$
$\boldsymbol{S}_{raw} = \boldsymbol{U}_{raw} \boldsymbol{V}_{raw} \boldsymbol{U}_{raw}^T$
Where $k = rank(\boldsymbol{S})$, d is the dimension of a single sample.
6 . Rebuilding the objective function by projecting $\boldsymbol{S}_{raw}$ to the sum covariance $\mathbf{\Sigma}$'s null space.
$\boldsymbol{S}_{op} = (\boldsymbol{B}_{null})^T \boldsymbol{S}_{raw} \boldsymbol{B}_{null}$
7 . Choosing the eigenvectors $\boldsymbol{B}_{com}$ corresponding to the first m largest eigenvalues of $\boldsymbol{S}_{op}$.
$(\boldsymbol{B}_{com})^T \boldsymbol{S}_{op} \boldsymbol{B}_{com} = \mathbf{\Lambda}$
8 . The feature vector $\boldsymbol{B}$ for discriminant analysis is described as
$\boldsymbol{B} = \boldsymbol{B}_{null} \boldsymbol{B}_{com}$

test the non-Gaussian problem in Section 3.2, the experimental results can support our viewpoint.

3 Experiments

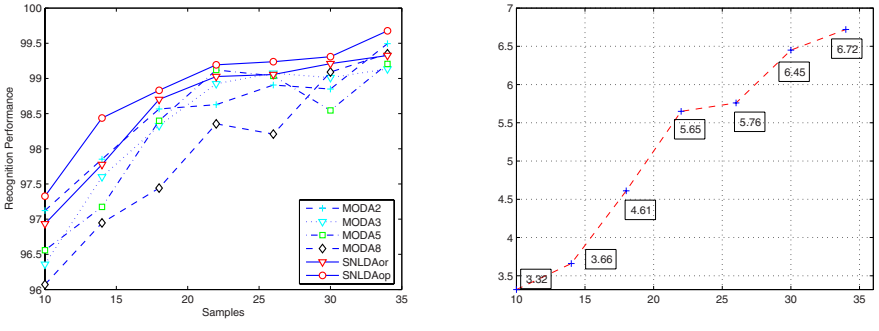
In this section, four databases, including a multi-view UMIST face database [7], two FERET face databases [8], and a simulated database, are used for evaluating the performance of the proposed SNLDA algorithm. Some examples of the three face databases are illustrated in Fig.1. In the UMIST database, 10 subjects with 54 images per person are randomly selected and each image is resized to 32×30 pixels. And both of the two FERET databases include 40 subjects, each with 10 images of the face. The first database (FERET1) is mainly composed of frontal images while in the second one (FERET2), large pose variation is introduced. The images in FERET1 are cropped to 30×30 pixels for removing the influences of background and hairstyle. In the FERET2 database, images are zoomed into 24×36 pixels followed by ellipse masking. Meanwhile, all the face images are roughly and manually aligned.

For all the experiments, each database is randomly divided into a training set and a test set without overlapping. The NN (nearest neighbor) algorithm will be applied for classification as soon as the dimension reduction is achieved. All the reported results is the average of 20 repetitions under the mentioned procedure.



(a) The UMIST Face Database (b) The FERET-1 Database (c) The FERET-2 Database

Fig. 1. Examples of The Three Face Databases



(a) (b)

Fig. 2. In left figure, $MODA_2 \sim MODA_8$ are the cluster number manually assigned model. $SNLDA_{op}$ redefines the covariance in single-sample clusters, while $SNLDA_{or}$ does not. The right figure denotes the average number of clusters detected by the SNLDA algorithm.

3.1 The UMIST Database

In the first subsection, we attempt to employ the UMIST database to compare the accuracy between the SNLDA algorithm and the MODA algorithm in which the number of clusters is manually set to be 2, 3, 5 and 8, respectively. The results on the average recognition rate versus the number of training samples of each class are shown as in Fig.2(a).

It can be seen from Fig.2(a) that when the MODA algorithm is applied, the highest recognition rates do not always correspond to a fixed number of clusters. Compared with the MODA algorithm, SNLDA can be self-tuned as the number of training samples varies and sounds more stable. What's more, the accuracy of the refined $SNLDA_{op}$ algorithm is always better than those of the other algorithms even if the number of training samples is fewer. It is clear that the mentioned disadvantage of the $SNLDA_{or}$ algorithm is partially overcame by the the $SNLDA_{op}$.

For better understanding the clustering algorithm in the SNLDA algorithm, a curve on the average detected number of clusters for the database is also illustrated in Fig 2(b). From the figure it can be found that as the number of training samples increases, the number of clusters also fluctuates. Also, none of the average cluster number is close to 1 or the upper bound of the class which is an important condition for ensuring the stability of subsequent classification. Therefore, the spectral clustering method used by the SNLDA algorithm is quite fit for data with small sample size and the proposed SNLDA approach is better in accuracy than the traditional MODA one.

3.2 Non-gaussian Simulated Database

In this subsection, a simulated database is generated for evaluating the discriminant ability of the proposed SNLDA algorithm under the condition of non-Gaussian covariance. Here 200 samples from five different 200-dimensional ($d = 200$) Gaussian classes were generated. Each sample of the c-th class is generated as $\boldsymbol{x}_i = \boldsymbol{B}_c\boldsymbol{c} + \boldsymbol{\mu}_c + \boldsymbol{n}$, where $\boldsymbol{x}_i \in \Re^{200}$. And each element of random matrix $\boldsymbol{B}_c \in \Re^{200\times 60}$ is generated from $\mathcal{N}(0, \boldsymbol{I})$, $\boldsymbol{c} \in \mathcal{N}_{60}(0, \boldsymbol{I})$, $\boldsymbol{n} \in \mathcal{N}_{200}(0, \boldsymbol{I})$. The means of five classes are $\boldsymbol{\mu}_1 \in 4[\overrightarrow{\mathbf{1}}_{200}]^T$, $\boldsymbol{\mu}_2 \in 4[\overrightarrow{\mathbf{0}}_{200}]^T$, $\boldsymbol{\mu}_3 \in -4[\overrightarrow{\mathbf{0}}_{100}\ \overrightarrow{\mathbf{1}}_{100}]^T$, $\boldsymbol{\mu}_4 \in 4[\overrightarrow{\mathbf{1}}_{100}\ \overrightarrow{\mathbf{0}}_{100}]^T$, $\boldsymbol{\mu}_5 \in -4[\overrightarrow{\mathbf{1}}_{50}\ \overrightarrow{\mathbf{0}}_{50}\ \overrightarrow{\mathbf{1}}_{50}\ \overrightarrow{\mathbf{0}}_{50}]^T$, respectively [3]. To delicate the performance on no-Gaussian covariance, we give a contrast of the classical PCA (Principal Component Analysis) algorithm, the $MODA_2$ algorithm and the $SNLDA_{op}$ algorithm. The experimental results are shown in Tab.2.

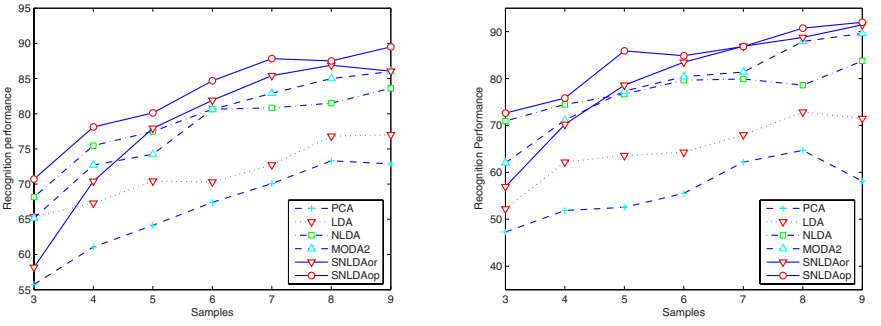
Table 2. A Comparison of Recognition Rate

Samples	2	6	10	14	18
PCA	55.36±4.89	57.88±7.76	66.06±5.97	70.88±7.29	72.86±5.50
$MODA_2$	$\mathbf{55.58 \pm 4.25}$	64.12±2.08	76.33±4.07	77.69±1.84	83.18±4.50
$SNLDA_{op}$	55.47±4.16	$\mathbf{67.29 \pm 3.90}$	$\mathbf{76.50 \pm 4.64}$	$\mathbf{82.92 \pm 2.72}$	$\mathbf{85.59 \pm 3.31}$

In the aspect of recognition rate, the SNLDA algorithm outperforms PCA by an average of 10% and the MODA algorithm by an average of 2.8%. An exception is that in the first column of the table, the accuracy of the three algorithms is almost the same. It is shown that when the number of training samples are small (e.g.$leq2$), the accuracy of the three mentioned algorithms are similar. The experiment results show that the SNLDA algorithm can get better performance than others even in non-Gaussian covariance conditions.

3.3 The FERET Databases

Finally, a comparative experiment among PCA, LDA (Linear Discriminant Analysis), NLDA (Null-space Linear Discriminant Analysis), MODA and the proposed SNLDA algorithm is carried out on the two FERET face databases. The results are shown in Fig.3. Considering the limitation of the paper's size, furthermore, only a table on



(a) The FERET-1 Face Database (b) The FERET-2 Face Database

Fig. 3. A comparison among different discriminant analysis methods

recognition rates with standard deviations of the FERET-2 database is tabulated in Tab.3. The results about the FERET-1 database are similar to those about the FERET-2 database.

From Fig. 3 and Tab.3, we can see that the performance of $SNLDA_{op}$ is always the best. When the number of samples is over 5, furthermore, the performance of two SNLDA algorithms are higher than the traditional methods. When 3 or 4 samples per subject are regarded as training samples, however, the multi-classes method MODA and $SNLDA_{or}$ are not ideal due to the influence of many single sample clusters. In addition, the MODA algorithm can outperform the $SNLDA_{or}$ algorithm in these conditions, that ascribe to the iterative procedure which can extract much information from the different normalization factors. Totally, the recognition rate of the $SNLDA_{op}$ algorithm is $4\% \sim 7\%$ higher than the traditional MODA and NLDA, almost 17% higher than PCA and LDA.

Table 3. Recognition Rates (%) with Stand Deviations (%) on the FERET-2 Face Database

Samples	3	4	5	6	7	8
PCA	47.28±2.68	51.88±1.53	52.56±2.96	55.54±1.62	62.19±2.67	64.69±4.42
LDA	52.21±2.30	62.17±2.96	63.58±2.41	64.27±2.34	67.99±3.29	72.85±4.00
NLDA	70.93±1.90	74.48±1.82	76.69±2.49	79.65±2.05	79.89±1.69	78.59±4.14
$MODA_2$	62.02±2.38	71.11±3.89	77.33±4.04	80.41±4.61	81.38±7.92	87.91±4.02
$SNLDA_{or}$	56.95±2.01	70.23±2.11	78.60±3.68	83.48±2.60	**86.89 ± 2.13**	88.76±3.21
$SNLDA_{op}$	**72.64 ± 3.54**	**75.83 ± 5.21**	**85.90 ± 3.36**	**84.87 ± 2.73**	86.83±3.35	**90.75 ± 4.38**

4 Conclusion

In this paper, we propose a spectral clustering based null space linear discriminant analysis algorithm. A main contribution is that we generalize the NLDA algorithm into multiple clusters through the combination of the spectral clustering and the proposed unified null-space technique. Considering SSS problem and the properties of null-space, meanwhile, two further refinements on the definition of covariance and the null-space

are proposed. The experimental results on face databases and simulated database show that the proposed SNLDA approach can help classifiers to obtain higher recognition rate than the mentioned traditional discriminant analysis approaches.

Recently, M. L. Zhu and A. M. Martinez have introduced the spectral clustering method for LDA [9] and get the appropriate cluster number by considering the angle information of the eigenvectors of the covariance matrix. We will try to combine the clustering process with the oriented discriminant analysis method and give a comparison with this method in the further researches.

Acknowledgement

This work is partially sponsored by NSFC (60635030), NSFC (60505002), and the State Key Laboratory of Rail Traffic Control and Safety (Beijing Jiaotong University), China. Part of the research in this paper uses the Gray Level FERET database of facial images collected under the FERET program.

References

1. L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu, "A New LDA-based Face Recognition System Which Can Solve The Small Sample Size Problem," *Pattern Recognition*, 2000, vol. 33, no. 10, pp. 1713–1726.
2. T. Hastie, R. Tibshirani, and A. Buja, "Flexible discriminant and mixture models," in proceedings of *Neural Networks and Statistics* conference, Edinburgh, 1995. J. Kay and D. Titterington, Eds. Oxford University Press.
3. F. de la Torre, and T. Kanade, "Multimodal oriented discriminant analysis," *Tech. Report CMURI-TR-05-03*, Robotics Institute, Carnegie Mellon University, January 2005.
4. S. X. Yu and J. Shi, "Multiclass Spectral Clustering," *International Conference on Computer Vision*. Nice, France, October, 2003, pp.11–17.
5. L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in Advances in Neural Information Processing Systems 17, LK Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1601–1608.
6. J. Huang, P. C. Yuen, W. S. Chen, J. H. Lai, "Choosing Parameters of Kernel Subspace-LDA for Recognition of Face Images under Pose and Illumination Variations," in Proceedings. Sixth IEEE International Conference on *Automatic Face and Gesture Recognition*. 2004, pp. 327–332.
7. D. B. Graham and N. M. Allinson, "em Characterizing virtual Eigensignatures for General Purpose Face Recognition," In: H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie and T. S. Huang (eds.): Face Recognition: From Theory to Applications. NATO ASI Series F, Computer and Systems Sciences, 1998, 163, pp. 446-456.
8. P. J. Phillips and H. Moon and S. A. Rizvi and P. J. Rauss, "The FERET Evaluation Methodology for Face Recognition Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000, vol. 22, no. 10, pp. 1090-1104.
9. M. L. Zhu and A. M. Martinez, "Subclass discriminant analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, vol. 28, no. 8, pp. 1274–1286.

Appendix

Proof. By definition, the covariance of the n-th cluster is formulated as:

$$\boldsymbol{\Sigma}_n = \sum_{l=1}^{L(n)} (\boldsymbol{x}_n^l - \boldsymbol{\mu}_n)(\boldsymbol{x}_n^l - \boldsymbol{\mu}_n)^T \tag{9}$$

where $\boldsymbol{x}_n^l \in \Re^d$ is the l-th sample in cluster n, $\boldsymbol{\mu}_n$ denotes the mean, $L(n)$ is the sample's number of cluster n. Let $\boldsymbol{A}_n = [\boldsymbol{x}_n^1 - \boldsymbol{\mu}_n, \ldots, \boldsymbol{x}_n^{L(n)} - \boldsymbol{\mu}_n]$, Eq.(9) can be expressed as:

$$\boldsymbol{\Sigma}_n = \boldsymbol{A}_n (\boldsymbol{A}_n)^T \tag{10}$$

if $\boldsymbol{\beta}_k$ is an orthonormal basis which can project $\boldsymbol{\Sigma}$ to its null space, then

$$\begin{aligned} 0 = (\boldsymbol{\beta}_k)^T \boldsymbol{\Sigma} \boldsymbol{\beta}_k = (\boldsymbol{\beta}_k)^T (\sum_{n=1}^{C} \boldsymbol{\Sigma}_n) \boldsymbol{\beta}_k = \sum_{n=1}^{C} (\boldsymbol{\beta}_k)^T \boldsymbol{\Sigma}_n \boldsymbol{\beta}_k \\ = \sum_{n=1}^{C} (\boldsymbol{\beta}_k)^T \boldsymbol{A}_n (\boldsymbol{A}_n)^T \boldsymbol{\beta}_k = \sum_{n=1}^{C} \|(\boldsymbol{A}_n)^T \boldsymbol{\beta}_k\|^2 \end{aligned} \tag{11}$$

where $\boldsymbol{\beta}_k$ is a basis in the null space of $\boldsymbol{\Sigma}$, $k \in 1, \ldots, R_N = d - rank(\boldsymbol{\Sigma})$, $\|\cdot\|$ denotes the Euclidean norm. Obviously, Eq.(11) holds if $(\boldsymbol{A}_n)^T \boldsymbol{\beta}_k = 0$. From this relation, we can see that

$$0 = ((\boldsymbol{A}_n)^T \boldsymbol{\beta}_k)^T (\boldsymbol{A}_n)^T \boldsymbol{\beta}_k = (\boldsymbol{\beta}_k)^T \boldsymbol{A}_n \boldsymbol{A}_n{}^T \boldsymbol{\beta}_k = (\boldsymbol{\beta}_k)^T \boldsymbol{\Sigma}_n \boldsymbol{\beta}_k \tag{12}$$

where k is independent, therefore, we have

$$0 = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{R_N}]^T \boldsymbol{\Sigma}_n [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{R_N}] = \boldsymbol{B}^T \boldsymbol{\Sigma}_n \boldsymbol{B} \tag{13}$$

which proves the proposition.

On a New Class of Framelet Kernels for Support Vector Regression and Regularization Networks

Wei-Feng Zhang[1], Dao-Qing Dai[1,⋆], and Hong Yan[2]

[1] Center for Computer Vision and Department of Mathematics
Sun Yat-Sen (Zhongshan) University, Guangzhou 510275 China
Tel.: (86)(20)8411 0141
stsddq@mail.sysu.edu.cn
[2] Department of Electronic Engineering, City University of Hong Kong
83 Tat Chee Avenue, Kowloon, Hong Kong

Abstract. Kernel-based machine learning techniques, such as support vector machines, regularization networks, have been widely used in pattern analysis. Kernel function plays an important role in the design of such learning machines. The choice of an appropriate kernel is critical in order to obtain good performance. This paper presents a new class of kernel functions derived from framelet. Framelet is a wavelet frame constructed via multiresolution analysis, and has both the merit of frame and wavelet. The usefulness of the new kernels is demonstrated through simulation experiments.

1 Introduction

The goal of regression is to approximate a function from function values, maybe perturbed by noise, evaluated at a finite set of points. Kernel methods, such as support vector machines (SVMs) [4,15,16], regularization networks (RN) [1], have been successfully applied to solve regression problems because of their capacity in handling nonlinear relations and learning form sparse data. Kernel function plays an important role in such methods. However, there are a number of open questions that have not been well solved such as the selection of kernel functions and selection of kernel parameters [3,5].

Recently, some researchers respectively proposed classes of frame-based kernels and demonstrated that they are superior to the well-established kernel functions in the framework of SVM regression (SVR) and RN [8,10,11,17]. The redundant property of frames make them well-suited to deal with noisy samples in a robust way. These kind of kernels are good at approximating multiscale functions. However, those frame-based kernels all depend on knowing the dual frame, which is always difficult to compute for a given frame. The effect of choosing frame elements with different approximation properties has not been considered. The purpose of this paper is to present a new class of kernel functions constructed by means of the framelet theory. A framelet is a multiresolution analysis (MRA)-based tight wavelet frame, its dual frame is itself. They combine the power of

⋆ Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 355–366, 2007.

MRA and the flexibility of redundant representations. We will use them in SVR and RN for function approximation. The usefulness of framelet-based kernels and effect of choosing different framelet elements will be discussed. We start by presenting some basic concepts of SVR and RN in section 2. Then we introduce the framelet theory in sections 3. In section 4 we introduce the construction of framelet kernels. Finally, some results on simulation experiments are presented in section 5, and section 6 concludes the paper.

2 Support Vector Regression and Regularization Networks

For regression problem with input-output pairs $\{(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathbb{R}, i = 1, 2, \ldots, l\}$ it is desired to construct a function f that maps input vectors $\mathbf{x}$ onto labels y. When new input $\mathbf{x}$ is presented the target output y is predicted by the function $f(\mathbf{x})$. Support vector regression and regularization networks are kernel-based techniques for solving regression problems of learning from examples. In fact, they both can be justified in Vapnik's Structural Risk Minimization (SRM) framework [7,13]. Next we briefly review the concepts of SVR and RN.

2.1 SVM for Regression

Consider the regression problem of estimating an unknown function $f : \mathbb{R}^n \to \mathbb{R}$, from the noisy observations

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \qquad i = 1, 2, \ldots, l$$

where the additive measurement errors ε_i are uncorrelated zero-mean Gaussian random variables. Let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$ be a set of training samples. The SV algorithm for regression computes a linear function in the high dimensional feature space $\mathcal{F}$. Thereby this algorithm can compute a nonlinear function by minimizing the following functional:

$$H[f] = \sum_{i=1}^{l} |y_i - f(\mathbf{x}_i)|_\varepsilon + \lambda \|f\|_K^2 \tag{1}$$

where $\|f\|_K^2$ is a norm in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_K$ defined by the kernel function $K(\mathbf{x}, \mathbf{y})$, $\lambda \in \mathbb{R}^+$ is a regularization constant and

$$|x|_\varepsilon = \begin{cases} 0 & \text{if } |x| < \varepsilon, \\ |x| - \varepsilon & \text{otherwise.} \end{cases}$$

is Vapnik's ε-insensitive loss function [15,16]. The parameter ε defines the tube around the regression function within which errors are not penalized.

2.2 Regularization Networks

Regularization theory is a classical way to solve the ill-posed problem of approximating a function from sparse data [2,14]. Classical regularization theory formulates the regression problem as the following variational problem

$$\min_{f\in\mathcal{H}} H[f] = \sum_{i=1}^{l}(y_i - f(\mathbf{x}_i))^2 + \lambda\|f\|_K^2 \tag{2}$$

where the second term is a penalty functional called stabilizer [9].

3 Framelet Theory

Framelet is tight wavelet frame constructed via MRA [6]. Tight wavelet frame is different from orthonormal wavelet in one important respect: it is redundant system but with the same fundamental structure as wavelet system. It has both the merit of frame and wavelet. We briefly present some results on the construction of framelets and the approximation properties of the system.

Suppose that $(V_j)_j$ is a MRA induced by a refinable function φ in $L^2(\mathbb{R}^d)$. Let $\Psi := \{\psi_i, i = 1, \ldots, r\}$ be a finite subset of V_1. Then there exist a set of 2π-periodic measurable functions $\{\tau_i, i = 1, \ldots, r\}$ called wavelet masks such that $\hat{\psi}_i = (\tau_i\hat{\varphi})(\frac{\cdot}{2})$ for every i. There also exists a 2π-periodic measurable function τ_0 called refinable mask such that $\hat{\varphi} = (\tau_0\hat{\varphi})(\frac{\cdot}{2})$. We will simplify the notations by writing $\tau := (\tau_0, \ldots, \tau_r)$ for the combined MRA mask. We define $\psi_{j,k}(\cdot) := 2^{jd/2}\psi(2^j \cdot -k)$ and the dyadic wavelet system

$$X(\Psi) := \{\psi_{j,k} : \psi \in \Psi, j \in \mathbf{Z}, k \in \mathbb{Z}^d\}$$

The following is the fundamental tool to construct framelets:

Definition 1. *Given a combined MRA mask $\tau := (\tau_0, \ldots, \tau_r)$, the fundamental function Θ is defined as*

$$\Theta(\omega) := \sum_{j=0}^{\infty}\sum_{i=1}^{r}|\tau_i(2^j\omega)|^2\prod_{m=0}^{j-1}|\tau_0(2^m\omega)|^2.$$

Proposition 1 (The Oblique Extension Principle (OEP) [6]). *Suppose that there exists a 2π-periodic function Θ that is non-negative, essentially bounded, continuous at the origin with $\Theta(0) = 1$. And for every $\omega \in \{-\pi, \pi\}^d$ and $\nu \in \{-\pi, \pi\}^d$*

$$\Theta(2\omega)\tau_0(\omega)\overline{\tau_0(\omega+\nu)} + \sum_{i=1}^{r}\tau_i(\omega)\overline{\tau_i(\omega+\nu)} = \begin{cases}\Theta(\omega) & \text{if } \nu = 0,\\ 0 & \text{otherwise.}\end{cases}$$

then the wavelet system $X(\Psi)$ defined by τ is a tight frame.

For $\Theta \equiv 1$, proposition 1 reduces to the *Unitary Extension Principle*.

The approximation order of the framelet system is proved to be strongly connected to the number of vanishing moment of the mother wavelet system [6]. Let $X(\Psi)$ be an framelet system, then

Proposition 2. *Assume that the system has vanishing moments of order m_1, and the MRA provides approximation order m_0. Then, the approximation order of $X(\Psi)$ is* $\min\{m_0, 2m_1\}$.

4 Framelet Kernels in Hilbert Space

The choice of the kernel $K(\mathbf{x}, \mathbf{y})$ determines the function space in which the norm $\|f\|_K^2$ in equation (1) and (2) is defined. It also determines the form of the solution. A kernel function computes the inner product of the images of two data points under a nonlinear map Φ

$$K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$$

where Φ defines the feature space. In practice, the kernel K can be defined directly without explicitly defining the map Φ. It is this property that makes the kernel methods so attractive.

Theorem 1. *Let $\Psi := \{\psi_1, \ldots, \psi_r\}$ be the mother wavelets of a framelet system*

$$X(\Psi) := \{\psi_{i,j,k} : \ i = 1, 2, \ldots, r, \ j \in \mathbb{Z}, \ k \in \mathbb{Z}^d\}$$

in $L^2(\mathbb{R}^d)$ where $\psi_{i,j,k}(\cdot) := 2^{jd/2}\psi_i(2^j \cdot -k)$. Then the framelet kernel is defined as

$$K(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{r} \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}^d} \psi_{i,j,k}(\boldsymbol{x}) \psi_{i,j,k}(\boldsymbol{y}) \tag{3}$$

Proof: We prove that framelet kernel (3) is admissible reproducing kernel.

With the kernel K, we can define a function space $\mathcal{H}_K$ to be the set of functions of the form

$$f(\mathbf{x}) = \sum_{i=1}^{r} \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}^d} \alpha_{i,j,k} \psi_{i,j,k}(\mathbf{x})$$

for $\alpha_{i,j,k} = \langle f, \psi_{i,j,k} \rangle \in \mathbb{R}$, and define the scale product in our space to be

$$\left\langle \sum_{i=1}^{r} \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}^d} \alpha_{i,j,k} \psi_{i,j,k}(\mathbf{x}), \sum_{i=1}^{r} \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}^d} \beta_{i,j,k} \psi_{i,j,k}(\mathbf{x}) \right\rangle_{\mathcal{H}_K} = \sum_{i=1}^{r} \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}^d} \alpha_{i,j,k} \beta_{i,j,k}$$

It is easy to check that such an Hilbert space is a RKHS with reproducing kernel given by $K(\mathbf{x}, \mathbf{y})$. In fact, we have

$$\langle f(\mathbf{y}), K(\mathbf{y}, \mathbf{x}) \rangle_{\mathcal{H}_K} = \sum_{i=1}^{r} \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}^d} \alpha_{i,j,k} \psi_{i,j,k}(\mathbf{x}) = f(\mathbf{x})$$

The framelet $\psi_{i,j,k}$ is a frame for the RKHS $\mathcal{H}_K$ which is called the feature space induced by the kernel K. Hence, the framelet-based kernel K and the corresponding space $\mathcal{H}_K$ can be used within the framework of SVR and RN.

Most constructions of framelets based on a spline MRA structure whose refinable function is chosen to be B-splines. In the following, we choose three univariate spline framelets presented in [6] to generate our kernels. They will be used in the experiments in next section.

A. Linear Spline Framelet Kernel K_1

Let ϕ be the B-spline function of order 2 supported on $[0, 2]$ which is a piecewise linear polynomial. Then the refinable mask is $\tau_0(\omega) = (1 + e^{-i\omega})^2/4$. Let

$$\tau_1(\omega) = -\frac{1}{4}(1 - e^{-i\omega})^2 \quad \text{and} \quad \tau_2(\omega) = -\frac{\sqrt{2}}{4}(1 - e^{-2i\omega})$$

be the wavelet masks. The corresponding $\{\psi_1, \psi_2\}$ generates a framelet system with vanishing moments 1 and the approximation order 2. We use $\{\psi_1, \psi_2\}$ to construct framelet kernel K_1

$$K_1(x, y) = \sum_{i=1}^{2} \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} \psi_{i,j,k}(x)\psi_{i,j,k}(y)$$

B. Cubic Spline Framelet Kernel K_2

Let ϕ be the B-spline function of order 4 supported on $[0, 4]$, which is a piecewise cubic polynomial. Then the refinable mask is $\tau_0(\omega) = (1 + e^{-i\omega})^4/16$. Let

$$\tau_1(\omega) = -\frac{1}{4}(1 - e^{-i\omega})^4, \ \tau_2(\omega) = -\frac{1}{4}(1 - e^{-i\omega})^3(1 + e^{-i\omega}),$$
$$\tau_3(\omega) = -\frac{\sqrt{6}}{16}(1 - e^{-i\omega})^2(1 + e^{-i\omega})^2, \ \tau_2(\omega) = -\frac{1}{4}(1 - e^{-i\omega})(1 + e^{-i\omega})^3.$$

The corresponding $\{\psi_1, \psi_2, \psi_3, \psi_4\}$ generates a framelet system with the vanishing moments 1 and the approximation order 2. We use $\{\psi_1, \psi_2, \psi_3, \psi_4\}$ to construct our framelet kernel function K_2

$$K_2(x, y) = \sum_{i=1}^{4} \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} \psi_{i,j,k}(x)\psi_{i,j,k}(y)$$

C. Cubic Spline Framelet Kernel K_3

The third one is also a cubic spline framelet kernel. It is based on the same MRA structure as in K_2 but has different vanishing moments and approximation order. We take

$\tau_1(\omega) = t_1(1 - e^{-i\omega})^4[1 + 8e^{-i\omega} + e^{-2i\omega}]$,
$\tau_2(\omega) = t_2(1 - e^{-i\omega})^4[1 + 8e^{-i\omega} + (7775/4396t - 53854/1099)e^{-2i\omega} + 8e^{-3i\omega} + e^{-4i\omega}]$,
$\tau_3(\omega) = t_3(1 - e^{-i\omega})^4[1 + 8e^{-i\omega} + (21 + t/8)(e^{-2i\omega} + e^{-4i\omega}) + te^{-3i\omega} + 8e^{-5i\omega} + e^{-6i\omega}]$.

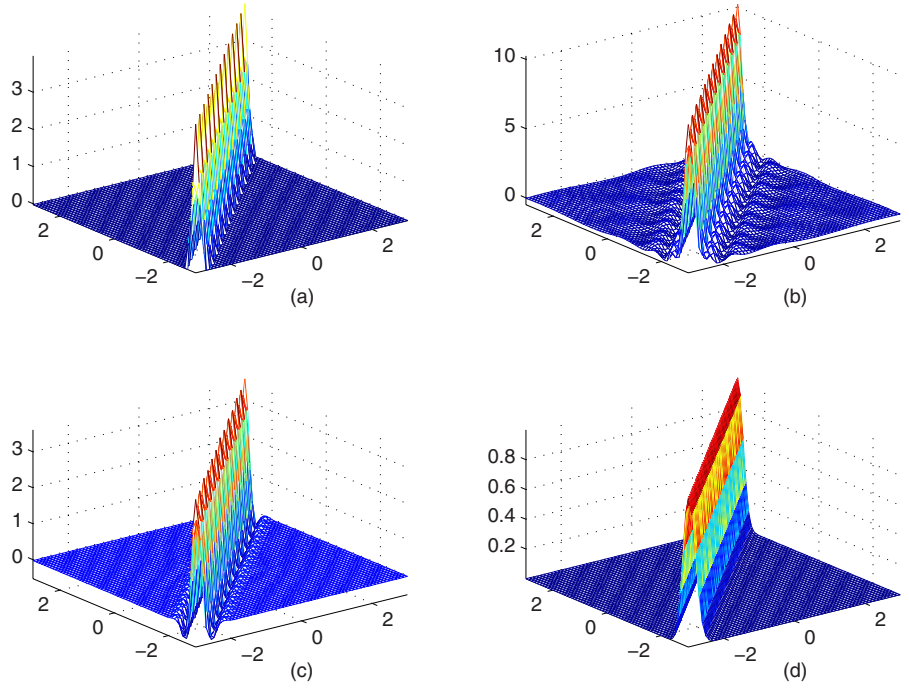


Fig. 1. Kernel functions. (a) Framelet kernel K_1 with $j_{min} = -5$, $j_{max} = 1$. (b) Framelet kernel K_2 with $j_{min} = -5$, $j_{max} = 1$. (c) Framelet kernel K_3 with $j_{min} = -5$, $j_{max} = 1$. (d) Gaussian kernel with $\sigma = 0.2$.

where $t = 317784/7775 + 56\sqrt{16323699891}/2418025, t_1 = \sqrt{11113747578360 - 245493856965t}/62697600, t_2 = \sqrt{1543080 - 32655t}/40320$, $t_3 = \sqrt{32655}/20160$.

Then the corresponding $\{\psi_1, \psi_2, \psi_3\}$ generates another framelet system with vanishing moments 4 and approximation order 4. We use $\{\psi_1, \psi_2, \psi_3\}$ to construct a framelet kernel function K_3

$$K_3(x, y) = \sum_{i=1}^{3} \sum_{j \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} \psi_{i,j,k}(x) \psi_{i,j,k}(y)$$

The sum of infinite terms can be truncated into a sum of finite terms in practical applications. For the three framelet kernels above, only finite terms corresponding to the shift index k are summed because the framelet elements are all compactly supported. We will truncate the infinite index j by defining the minimal and maximal dilations j_{min} and j_{max}. Note that these last two parameters determine the different scales in the kernel function. The kernel functions K_1, K_2 and K_3 with $j_{min} = -5$ and $j_{max} = 1$ are plotted in Figure 1(a),(b) and (c) respectively, (d) is the Gaussian kernel.

5 Simulation Experiments

This section provides two experiments on single-variate function regression problems using SVM regression and regularization networks methods. We illustrate the usefullness of framelet-based kernels K_1, K_2, K_3 by comparing with the classical Gaussian kernel

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

in the simulated regression experiments I and II.

For both SVR and RN, some hyperparameters have to be tuned. The performance of SVR in (1) depends on the hyperparameters such as ε, regularization factor λ. The performance of RN in (2) depends on the choice of regularization factor λ. Different approaches have been developed for solving this model selection problem [3,5]. The idea is to find the parameters that minimize the generalization error of the algorithm at hand. This error can be estimated either via a bound given by theoretical analysis or via testing on some data which has not been used for learning (hold-out testing or cross-validation techniques).

In our experiments, the hyperparameters were optimized from a range of finely sampled values, where the generalization error was estimated by the 10-fold cross-validation. We split the data set into 10 roughly equal-sized parts; for the ith part, we fit the model to the rest parts of the data, and calculate the mean-square error of the fitted model when predicting the ith part of the data; the generalization error is estimated by averaging the 10 mean-square prediction errors.

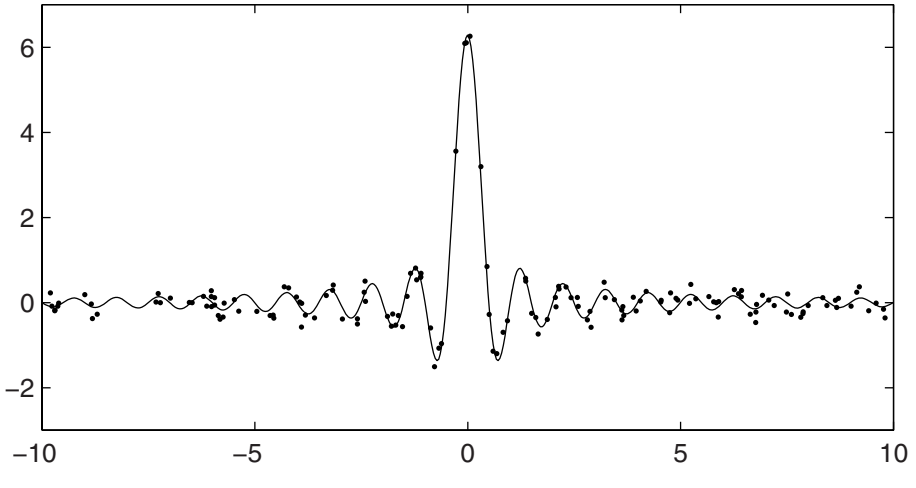


Fig. 2. The true function f_1 and its sample set

Table 1. Generalization error for SVM regression and regularization networks using Framelets and Gaussian kernel with optimal hyperparameters in experiment I

	SVM Regression	Regularization Networks
Framelet Kernel K_1	0.0224	0.0272
Framelet Kernel K_2	0.0182	0.0253
Framelet Kernel K_3	0.0168	0.0221
Gaussian Kernel	0.0203	0.0132

5.1 Experiment I

In this experiment, the sample set $\{(x_i, y_i)\}$ come from the function

$$f_1(x) = x^{-1}\sin(2\pi x)$$

which was commonly used to test SVR [8,16]. The data points $\{x_i\}_{i=1}^{150}$ were obtained from uniform random sampling 150 data points of interval $[-10, 10]$, they were not necessarily equal-spaced. The targets y_i were corrupted by the zero-mean Gaussian noise with variance 0.2. Figure 2 shows the sample set and the true function.

The performance of the framelet kernel strongly depends on the value of the scale parameters j_{min} and j_{max}. We find that if we fix j_{max}, the generalization

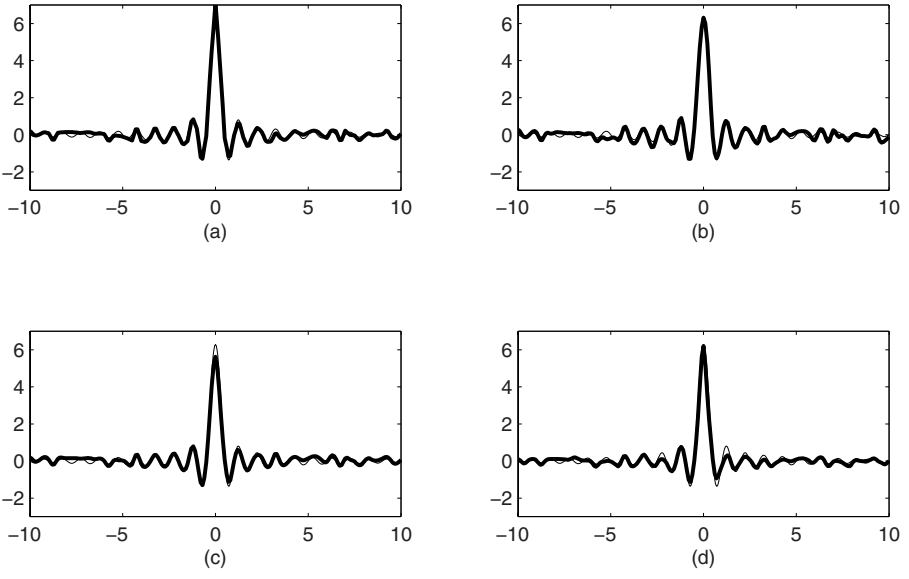


Fig. 3. SVM regression results for experiment I. (a) Framelet kernel K_1 with $j_{min} = -5$, $j_{max} = 1$, $\lambda = 0.4$, $\varepsilon = 0.06$. (b) Framelet kernel K_2 with $j_{min} = -5$, $j_{max} = 1$, $\lambda = 0.2$, $\varepsilon = 0.04$. (c) Framelet kernel K_3 with $j_{min} = -5$, $j_{max} = 1$, $\lambda = 0.6$, $\varepsilon = 0.06$. (d) Gaussian kernel with $\sigma = 0.15$, $\lambda = 0.2$, $\varepsilon = 0.02$.

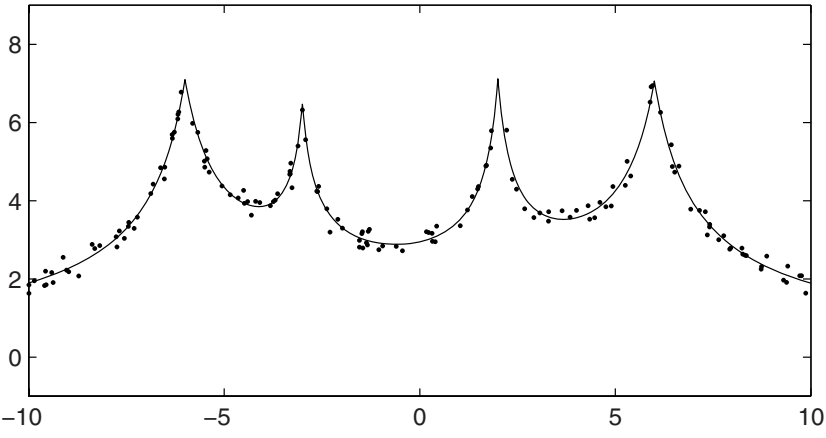


Fig. 4. The true function f_2 and the sample set

Table 2. Generalization error for SVM regression and regularization networks using Framelet and Gaussian kernels with optimal hyperparameters in experiment II

	SVM Regression	Regularization Networks
Framelet Kernel K_1	0.0221	0.0133
Framelet Kernel K_2	0.0343	0.0250
Framelet Kernel K_3	0.0318	0.0238
Gaussian Kernel	0.0589	0.0260

error for the two learning machines decreases monotonously as j_{min} reduced. However, the results improve slightly after the value of -5. So we fix the scale lower bound parameter j_{min} to be -5. The scale upper bound parameter j_{max} is taken from -3 to 5. The parameter σ for Gaussian kernel is selected by using cross validation introduced above. Table 1 lists the generalization error for the two learning machines and the different kernels using the optimal hyperparameter setting. For SVR the optimal parameters are $j_{min} = -5$, $j_{max} = 1$, $\lambda = 0.4$, $\varepsilon = 0.06$ for K_1; $j_{min} = -5$, $j_{max} = 1$, $\lambda = 0.2$, $\varepsilon = 0.04$ for K_2; $j_{min} = -5$, $j_{max} = 1$, $\lambda = 0.6$, $\varepsilon = 0.06$ for K_3; and $\sigma = 0.15$, $\lambda = 0.2$, $\varepsilon = 0.02$ for Gaussian kernel. For RN the optimal parameters are $j_{min} = -5$, $j_{max} = 1$, $\lambda = 0.4$ for K_1; $j_{min} = -5$, $j_{max} = 1$, $\lambda = 1$ for K_2; $j_{min} = -5$, $j_{max} = 1$, $\lambda = 0.6$ for K_3; and $\sigma = 0.35$, $\lambda = 0.1$ for Gaussian kernel. Figure 3 shows the SVR results of the optimal parameters. The slender lines depict the true function and the bold lines represent the approximation functions.

As can be seen, the framelet kernels achieve good performance in SVR. The best performance in SVR is given by the framelet kernel K_3 which has the highest approximation order. In RN, the framelet kernels do not give better performance compared to Gaussian kernel. The best result in RN is achieved by Gaussian kernel and it is better than the best one in SVR. This may be because the sample set came from a smoothly changed function and large sample size

give enough information for approximation, and the SVM algorithm loses its superiority.

5.2 Experiment II

To illustrate the multiscale approximation property of framelet kernels, we use the following function

$$f_2(x) = \frac{6}{|x+6|+1} + \frac{4}{3|x+3|+1} + \frac{5}{2.5|x-2|+1} + \frac{6}{|x-6|+1}$$

which contains multiple scales as shown in Figure 4. The tips of the cones at $x = -6$, -3, 2 and 6 are singular points so that $f_2(x)$ is not differentiable there. Note that the function is composed of four cone-shaped parts which are different from each other of the width and position. A data set of 150 is generated by uniform random sampling from the interval $[-10, 10]$ in which the targets are corrupted by the zero-mean Gaussian noise with variance 0.2. Figure 4 shows the sample set and the true function.

Table 2 shows the generalization error for the two learning machines and the different kernels using the optimal hyperparameter setting. For SVR the optimal parameters are $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.6$, $\varepsilon = 0.04$ for K_1; $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.2$, $\varepsilon = 0.06$ for K_2; $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.6$, $\varepsilon = 0.02$ for K_3; and $\sigma = 0.4$, $\lambda = 0.6$, $\varepsilon = 0.02$ for Gaussian kernel. For RN

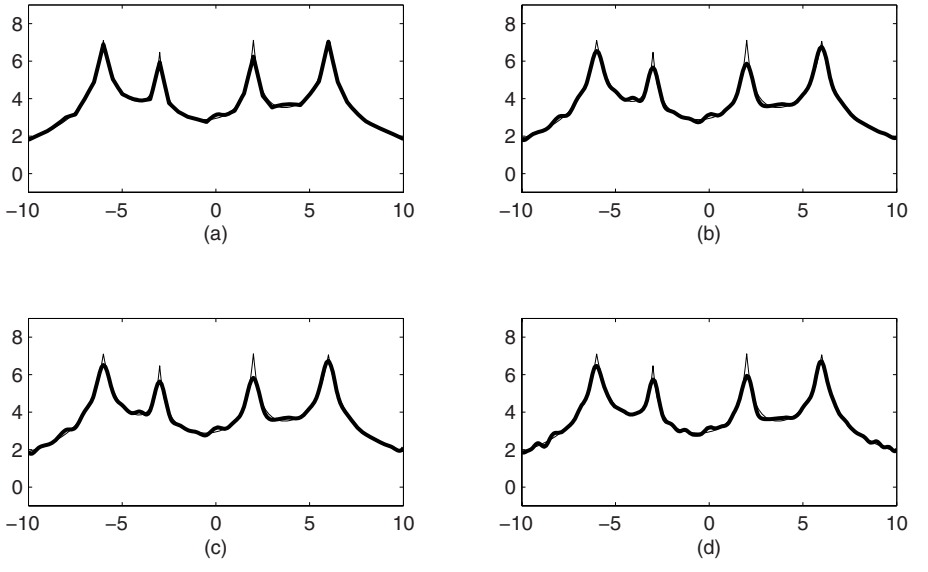


Fig. 5. Regularization Networks results for experiment II. (a) Framelet kernel K_1 with $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.1$. (b) Framelet kernel K_2 with $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.1$. (c) Framelet kernel K_3 with $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.1$. (d) Gaussian kernel with $\sigma = 0.3$, $\lambda = 0.1$.

the optimal parameters are $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.1$ for K_1; $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.1$ for K_2; $j_{min} = -5$, $j_{max} = 0$, $\lambda = 0.1$ for K_3; and $\sigma = 0.3$, $\lambda = 0.1$ for Gaussian kernel. Figure 5 shows the RN results of the optimal parameters. The slender lines depict the true function and the bold lines represent the approximation functions.

The neighborhood of the singular points is the most difficult part to approximate. It is obvious that the performance of the framelet kernels is superior to the Gaussian kernel in that place. In both SVR and RN, Gaussian kernel achieved the poorest performance and framelet kernl K_1 is the best. This suggests that the framelet kernels are good at catching the multiscale structures of the signal.

6 Conclusion

In this paper, we introduced a class of kernel functions based on framelet theory for the learning methods of SVM regression and regularization networks. This kind of kernels inherit the merits of multiscale representation and redundant representation from the framelet system. They are good at approximating functions with multiscale structure and can reduce the influence of noise in data. Specifically, there is sufficient choice of framelet kernels, the tools introduced in section 3 facilitate us greatly to construct framelet kernels with certain approximation properties. Experiments in this article illustrated the superiority of the newly proposed kernels to the classical kernel. More comparison with other kernels and extensions to two dimensions will be considered in our future work.

Acknowledgments

This project is supported in part by NSF of China(60175031, 10231040, 60575004), the Ministry of Education of China(NCET-04-0791), NSF of Guang-Dong (05101817) and the Hong Kong Research Grant Council(project CityU 122506).

References

1. T. Poggio and F. Girosi, Networks for approximation and learning, *Proc. IEEE*, 78(9), 1481-1497, 1990.
2. M. Bertero, T. Poggio and V. Torre. Ill-posed problems in early vision, *Proc. IEEE*, 76, 869-889, 1988.
3. O. Chapelle, V.N. Vapnik, O. Bousquet and S. Mukherjee. Choosing multiple parameters for support vector machines, *Machine Learning*, 46, 131-159, 2002.
4. C. Cortes and V.N. Vapnik. Support vector networks, *Machine Learning*, 20, 1-25, 1995.
5. F. Cuker and S. Smale. Best choices for regularization parameters in learning theory: on the bias-variance problem, *Found. Comput. Math.* 2, 413-428, 2002.
6. I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: MRA-based constructions of wavelet frames, *Appl. Comput. Harmon. Anal.* 124, 44-88, 2003.

7. T. Evgeniou, M. Pontil and T. Poggio. Regularization networks and support vector machines, *Advances in Computational Mathematics*, 13, 1-50, 2000.
8. J. B. Gao, C. J. Harris and S. R. Gunn. On a class of support vector kernels based on frames in function hilbert spaces, *Neural Computation*, 13(9), 1975-1994, 2001.
9. F. Girosi, M. Jones and T. Poggio. Regularization theory and neural networks architectures, *Neural Comput*, 7, 219-269, 1995.
10. K. R. Müller, S. Mika, G. Rätsch, K. Tsuda and B. Schölkopf. An introduction to kernel-based learning algorithms, *IEEE Transactions on Neural Networks*, 12(2), 181-201, 2001.
11. A. Rakotomamonjy, X. Mary and S. Canu. Non-parametric regression with wavelet kernels, *Appl. Stochastic Models Bus. Ind.*, 21, 153-163, 2005.
12. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
13. A. J. Smola, B. Schölkopf and K. R. Müller. The connection between regularization operators and support vector kernels, *Neural Networks*, 11, 637-649, 1998.
14. A. N. Tikhonov and V.Y. Arsenin. *Solutions of Ill-posed Problems* W.H Winston, Washington, DC, 1977.
15. V. N. Vapnik. *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.
16. V. N. Vapnik. *Statistical Learning Theory.* New York: Wiley, 1998.
17. L. Zhang, W. Zhou, and L. Jiao, Wavelet support vector machine, *IEEE T. on System, Man and Cybernetics-Part B: Cybernetics*, Vol. 34, No. 1, 34-39 2004.

A Clustering Algorithm Based on Mechanics

Xianchao Zhang, He Jiang, Xinyue Liu, and Hong Yu

School of Software, Dalian University of Technology, Dalian 116621,China
{xczhang,jianghe,liuxy,hongyu}@dlut.edu.cn

Abstract. Existing clustering algorithms use distance, density or concept as clustering criterion. These criterions can not exactly reflect relationships among multiple objects, so that the clustering qualities are not satisfying. In this paper, a mechanics based clustering algorithm is proposed. The algorithm regards data objects as particles with masses and uses gravitation to depict relationships among data objects. Clustering is executed according to displacements of data objects caused by gravitation, and the result is optimized subjecting to Minimum Potential Energy Principle. The superiority of the algorithm is that the relationships among multiple objects are exactly reflected by gravitation, and the multiple relationships can be converted to the single ones due to force composition, so that the computation can be executed efficiently. Experiments indicate that qualities of the clustering results deduced by this algorithm are better than those of classic algorithms such as CURE and K-Means.

Keywords: Data Mining; Clustering Analysis; Mechanics; Minimum Potential Energy Principle.

1 Introduction

Clustering analysis is the data mining technique to achieve the feature and pattern information of unknown object sets. It is widely used in many applications such as financial data classification, spatial data processing, satellite photo analysis, medical figure auto-detection and so on. Since the 1940's, researchers have proposed a large number of clustering algorithms (CURE[1], CHAMELEON[2], BIRCH[3], COBWEB[4], PDDP[5], K-Means[6,7], CLARANS[8], CLARA[9], WaveCluster[10], STING[11], CLIQUE[12], DBSCAN[13], OPTICS[14], DBCLASD[15]) .

Clustering divides the data set into clusters, making objects similar in the same cluster and dissimilar among different clusters. The quality of clustering algorithms depends heavily on the clustering criteria. Three kinds of criteria are employed in existing algorithms: distance (e.g., CURE、PDDP、K-Means), density (e.g., DBSCAN、OPTICS、DBCLASD) and concept (e.g., COBWEB). Distance only reflects the relationship between two objects, while can't reflect the relationships and interaction among multiple objects. Density is the average distance among objects in a region, which reflects the global characteristic of the whole region and can't reflect the relationship between each pair of objects. Concept is constrained with the data type and only fits for conceptual and categorical data. So current clustering criteria

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 367–378, 2007.

can not quantitatively reflect the relationships among multiple objects, and qualities of the clustering algorithms based on the above criteria are not satisfying in actual applications.

In this paper, a new clustering algorithm based on Mechanics (CABM) is proposed, which uses gravitation to reflect the relationship among data objects. The algorithm regards data objects as particles with mass, and they are linked with flexible poles which twist as trussing structures. The function of the trussing structures is to limit the particles' moving scope so as to keep the original shape of the clusters. The structures deform under gravitation, and when the structures are stable, particles are grouped into clusters according to displacements and the change of the structure's potential energy. The advantage of CABM is that the relationships among multiple objects are reflected as forces. Due to the composition of forces, multiple relationships can be equally converted to single ones and the computation is simplified. Experimental results indicate that the quality of CABM is much better than those of CURE and K-means.

The rest of this paper is organized as follows. In section 2 we introduce some definitions and notations. The CABM algorithm is described in detail in section 3. We evaluate the experimental results upon CURE, K-Means and CABM in section 4. And finally, the paper is concluded in section 5.

2 Preliminaries

Clustering is the problem of grouping data objects based on similarity.

Definition 1. Let $P=\{p_1, p_2, \cdots, p_n\}$ be a set of data objects. **Clustering** is the process of dividing P into sub-sets $C=\{C_1, C_2, \cdots, C_k\}$, such that $\forall i, j \in \{1,2,\cdots,k\}$, $\bigcup_{i=1}^{k} C_i = P$ $C_i \bigcap C_j = \phi$, $C_i \neq \phi, C_j \neq \phi$, where C_i is called the ith **cluster** of P.

The objective of clustering is to achieve high similarity among objects inside a same cluster as well as high dissimilarity among different clusters. Sergio M.Savaresi, etc. proposed a method to evaluate the quality of clustering algorithms as follows.

Given $w=\{w_1, w_2, ..., w_k\}$, $\forall i, j \in \{1,2,...,k\}$, where w_i is the barycenter of C_i. The inner-cluster similarity is measured by $SC_i = \sum_{p \in C_i} \| p - w_i \| / | C_i |$.

The inter-cluster dissimilarity is measured by the distance between two clusters, i.e., $d_i = \min_j(d_{ij})$, $d_{ij} = \| w_i - w_j \|$.

With the above inner-cluster similarity and inter-cluster dissimilarity measures, the quality of a clustering division is computed as:

$$Q(C_1, C_2, \ldots, C_k) = \sum_{i=1}^{k} | C_i | SC_i / d_i n \quad (1)$$

The smaller Q is, the higher clustering quality is achieved.

In CABM, each object p_i is viewed as a particle with unit mass in an s-dimension data space. The gravitation between each pair of particles is defined as follows:

Definition 2. Let o_i and o_j are two particles with mass m_i and m_j respectively, the Euclid-Distance between o_i and o_j is denoted by l_{ij}, and the gravitation constant is denoted by G_0, then the **gravitation** between o_i and o_j is $\overrightarrow{F_{ij}} = G_0 \mathrm{m}_i \mathrm{m}_j / l_{ij}^2$.

The particles move under the gravitation until equilibrium is reached. If there isn't any constraint on the particles, the particles will contract into a mass. Then the particles leave their original positions completely and the clusters' original shapes are broken. Therefore, in order to reflect the motion tendency and limit the motion scope of particles, linear elasticity poles are added between some pairs of particles to constraint the particles' motion in CABM algorithm. We denote the intersection surface of a pole by A.

Definition 3. The force imposed on the intersection surface of a pole is called a **stress**, denoted by σ. The deformation on a unit length of a pole is called a **strain**, denoted by ε. The ratio of a pole's stress to its strain is called **elasticity modulus**, denoted by $E = \sigma / \varepsilon$.

Assume that a pole with length l_{ij} deforms under the stalk force and the deformation is Δl_{ij}. Then the pole's stress is $\sigma = F / A$ and the strain is $\varepsilon = \Delta l_{ij} / l_{ij}$.

3 Clustering Algorithm Based on Mechanic (CABM)

CABM firstly divides the data set into several regions and elects some delegation nodes for each region. The algorithm then builds a truss on the delegation nodes, computes the displacements of the nodes and clusters the nodes according to the displacements. Finally the algorithm marks the original data objects according to the clustering result of the delegation nodes and the clustering result on the original data set is achieved.

3.1 Pre-processing

The aim of pre-processing is to eliminate noise and reduce the scale of the data set. CABM adopts the same processing method as CURE (see [1]):

Firstly, divide the data set P as $P = \{R_1, R_2, \cdots, R_t\}$, subjecting to $\forall i, j \in \{1, 2, \cdots, t\}$, $R_i \cap R_j = \phi$ and $\bigcup_{i=1}^{t} R_i = P$, each R_i is called a **candidate cluster**. It is proved that the number of clusters of a data set with *n* objects is less than $\sqrt{n}$ [16], thus in this paper, $t = \sqrt{n}$.

Let $w' = \{w'_1, w'_2, \cdots, w'_t\}, \forall i \in \{1,2,\cdots,t\}$, where w'_i denotes the barycenter of candidate cluster R_i, whose mass is denoted by m'_i, and its coordinate in the s-dimension data space is $(X'_{i1}, X'_{i2}, \cdots, X'_{is})$, then $\forall j \in \{1,2,\cdots,s\}$, $k \in \{1,2,\cdots,n\}$,

$$X'_{ij} = \frac{\sum_{k=1}^{n} {}_{p_k \in R_i} x_{kj}}{|R_i|}, \quad m'_i = |R_i|. \tag{2}$$

Secondly, select π particles in the candidate cluster and contract the particles towards the barycenter with a given contracted factor α [1], the resulting virtual particles are the delegation nodes of the candidate cluster. Let $i \in \{1,2,\cdots,t\}$, and $V_i = \{v_{i1}, v_{i2}, \cdots, v_{i\pi}\}$ is the delegation node set of the candidate cluster R_i. Suppose that $j \in \{1,2,\cdots,\pi\}, v_{ij}$ is the jth delegation node of R_i whose coordinate is $(X_{ij1}, X_{ij2}, \cdots, X_{ijs})$, and the mass of V_{ij} is m_{ij}, then,

$$m_{ij} = \frac{\| v_{ij} - w'_i \|}{\sum_{a=1}^{\pi} \| v_{i\alpha} - w'_i \|} m'_i \tag{3}$$

.

Definition 4. $\forall k \in \{1,2,\cdots,t\}$, $\zeta_k = \frac{1}{\binom{|R_k|}{2}} \sum_{i=1}^{\pi} \sum_{j=i+1}^{\pi} \| v_{ik} - v_{jk} \|$ is the **effective radius** of R_k.

Definition 5. Let ζ_{cr} be a pre-defined threshold, for $i \in \{1,2,\cdots,t\}$, if $\zeta_k > \zeta_{cr}$, then the candidate cluster R_i is an **isolated region**, and if $p_j \in R_i$, then p_j is an **outlier**.

ζ_{cr} is chosen by experience. All the candidate clusters except **isolated regions** are the study objects of CABM algorithm.

3.2 Building the Trussing Structure

The quantity of the gravitation depends on the distance between a pair of delegation nodes. The delegation nodes can be classified as adjacent nodes and non-adjacent nodes.

Definition 6. Let v_{ij}, v_{ik} be two delegation nodes in the ith candidate cluster, if $\| v_{ij} - v_{ik} \| \leq \zeta_i$, then v_{ij}, v_{ik} are **adjacent nodes**, otherwise, they are **non-adjacent nodes**.

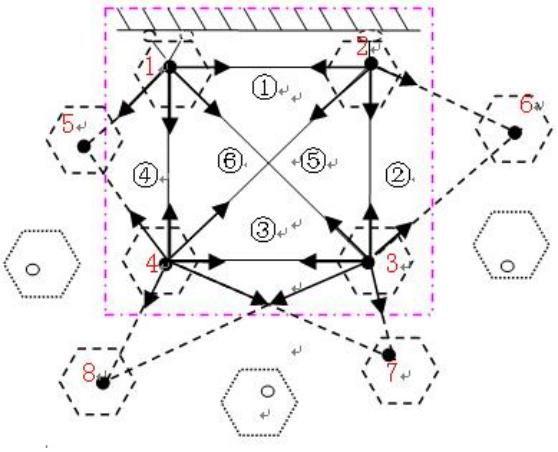


Fig. 1. Diagram of building the trussing structure

In CABM, the adjacent nodes in a candidate cluster are connected with poles. The poles twist with each other. Fig. 1 shows the distribution of some delegation nodes. In the figure, the solid circles represent delegation nodes, and the hollow circles represent outliers. Four delegation nodes inside the frame are in the same candidate cluster.

For every trussing structure, consider the gravitation between the every pair of nodes inside as well as outside the candidate cluster. According to the principle of composition of forces and the orthogonal decomposition method, compose the force computed each time and decompose to the force belong the each direction $x_1, x_2, \cdots, x_s$, denoted by $\overrightarrow{F^i} = \left[\overrightarrow{F_{x_1}^i}, \overrightarrow{F_{x_2}^i}, \cdots, \overrightarrow{F_{x_s}^i}\right]^{\mathrm{T}}$, see Fig. 1. Now the trussing structure is built and the forces are computed.

3.3 The Analysis and Computation of the Trussing Structure

Poles of a trussing structure will deform under external forces. Let the intersection angle of the pole and the positive direction of the axis of the global coordinates $x_1, x_2, \cdots, x_s$ is $\beta_1, \beta_2, \cdots, \beta_s$, then the unit stiffness matrix in the global coordinates is $\Omega^e = \theta^{-1}\overline{\Omega^e}\theta$, in which θ is transformation of coordinates matrix as follows:

$$\theta = f(\beta_1, \beta_2, \cdots, \beta_s) \ . \tag{4}$$

For the six nodes in Fig. 1, we have the unit stiffness matrices $\Omega^{(1)}, \Omega^{(2)}, \Omega^{(3)}, \Omega^{(4)}, \Omega^{(5)}$ and $\Omega^{(6)}$. So the total stiffness matrix of the trussing structure shown in Fig. 1 is:

$$\Omega^e = \begin{bmatrix} \Omega_{11}^{(1)} + \Omega_{11}^{(4)} + \Omega_{11}^{(5)} & \Omega_{12}^{(1)} & \Omega_{13}^{(5)} & \Omega_{14}^{(4)} \\ \Omega_{21}^{(1)} & \Omega_{22}^{(1)} + \Omega_{22}^{(2)} + \Omega_{22}^{(6)} & \Omega_{23}^{(2)} & \Omega_{24}^{(6)} \\ \Omega_{31}^{(5)} & \Omega_{32}^{(2)} & \Omega_{33}^{(2)} + \Omega_{33}^{(3)} + \Omega_{33}^{(5)} & \Omega_{34}^{(3)} \\ \Omega_{41}^{(4)} & \Omega_{42}^{(6)} & \Omega_{43}^{(3)} & \Omega_{44}^{(3)} + \Omega_{44}^{(4)} + \Omega_{44}^{(6)} \end{bmatrix} \tag{5}$$

According Hooke's law $\overrightarrow{F^e} = \Omega^e \overrightarrow{\Delta^e}$, we have:

$$\begin{Bmatrix} \overrightarrow{F^1} \\ \overrightarrow{F^2} \\ \overrightarrow{F^3} \\ \overrightarrow{F^4} \end{Bmatrix} = \begin{bmatrix} \Omega_{11}^{(1)} + \Omega_{11}^{(4)} + \Omega_{11}^{(5)} & \Omega_{12}^{(1)} & \Omega_{13}^{(5)} & \Omega_{14}^{(4)} \\ \Omega_{21}^{(1)} & \Omega_{22}^{(1)} + \Omega_{22}^{(2)} + \Omega_{22}^{(6)} & \Omega_{23}^{(2)} & \Omega_{24}^{(6)} \\ \Omega_{31}^{(5)} & \Omega_{32}^{(2)} & \Omega_{33}^{(2)} + \Omega_{33}^{(3)} + \Omega_{33}^{(5)} & \Omega_{34}^{(3)} \\ \Omega_{41}^{(4)} & \Omega_{42}^{(6)} & \Omega_{43}^{(3)} & \Omega_{44}^{(3)} + \Omega_{44}^{(4)} + \Omega_{44}^{(6)} \end{bmatrix} \begin{Bmatrix} \overrightarrow{\Delta^1} \\ \overrightarrow{\Delta^2} \\ \overrightarrow{\Delta^3} \\ \overrightarrow{\Delta^4} \end{Bmatrix} \tag{6}$$

Where $\overrightarrow{\Delta^i}$ is the displacement of node i and can be computed following the formula.

In Fig. 2, the arrows indicate the displacements of nodes.

Because of displacements, the candidate cluster to which each delegation node belongs should be redefined. Now we give some related geometric definitions.

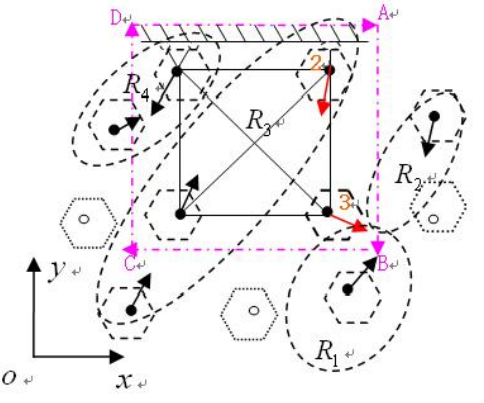


Fig. 2. Displacements of node in the trussing structure under forces

Definition 7. The **boundary** of the candidate clusters consists of planes which divide the whole region. A closed region enclosed by the planes is called a **boundary polyhedron**. Providing that the positive direction of an edge on every plane of a boundary polyhedron is clockwise, the direct edge which makes up the boundary polyhedron is called **boundary vector**.

Let $\overrightarrow{vv'}$ represent the displacement of node v, let u be the node closest to v in the boundary polyhedron. The projection of $\overrightarrow{vv'}$ on the plane to which u belongs is

denoted by $\overrightarrow{v''v'''}$. If the plane with u is a concave polygon and η is a concave point, then the two boundary vectors on the counter-clockwise direction linked with η are noted as $\vec{e}_1, \vec{e}_2$, otherwise the two boundary vectors are noted as $\vec{e}_1, \vec{e}_2$ on the clockwise direction. $\vec{n}$ denotes the normal of the plane.

Principle 1. Point v belongs to the original candidate cluster after the deformation if and only if $(\vec{e}_1 \times \overrightarrow{v''v'''}) \bullet \vec{n} \le 0$ and $(\overrightarrow{v''v'''} \times \vec{e}_2) \bullet \vec{n} \ge 0$.

Principle 1 prescribes whether a delegation node should belong to the original cluster it is located. The principle is based on geometrical principles, we omit its proof here to save space. Delegation nodes not belonging to their original clusters need to be relocated into other clusters, we show how to do this in the next sub-section.

3.4 Delegation Node Relocation

When a delegation node does not belong to the original candidate clusters, it needs to be relocated into a new cluster. At this time, the stability of the structure should be considered.

Theorem 1. An elastic pole or a structure is in the stable state if and only if its potential energy under the deformation is the minimum[17,18].

Theorem 1 indicates that the deformation of a structure tends to make the potential energy minimum.

Rule 1. (**Node Relocation Rule**) Delegation nodes should be located into candidate clusters so that the potential energy of the structure is the minimum.

Move a delegation node along the direction of displacement vector to the corresponding new candidate cluster. The motion is limited by the boundaries of the adjacent candidate clusters. An example is shown in Fig. 2. As Fig. 2 shows, v_3 is judged not to belong to R_3. The displacement vector of v_3 points to R_1 and R_2, v_3 should be relocated into the one which makes the potential energy minimum.

CABM iteratively executes the rule until the sum of the potential energy of all the trussing structures converges. At this time, there isn't any node motion among the candidate clusters, and the clustering result is stable.

3.5 Original Data Object Labeling

The above process is the clustering process of the delegation nodes. In order to cluster the original data objects, the algorithm labels each original data object with the index of the cluster to which its corresponding delegation node belongs. Now every original data object has a unique index except outliers. Objects with the same index are regarded as to be in a same cluster, in this way, the task of clustering is accomplished. See algorithm 1 for the whole process of CABM.

Algorithm 1. CABM

Algorithm: CABM
Input: the set of data points and related parameters
Output: clusters
Begin:

(1) Data processing: Divide the set of data objects into t candidate clusters. Select π virtual points in each candidate cluster and contract the points towards the centre by the given contracted factor α to get the delegation node. Judge the isolated regions and outliers according to given parameter ζ_{cr}.
(2) Constructing truss: Link the adjacent points and get the truss. Compute the gravitation between particles and represent the gravitation on a particle with an equivalent external composition of force. The external force acts on the points of the truss. Compute the displacement of every particle.
(3) Iteratively executes the node relocation rule until the sum of the potential energy of all the trussing structures converges.
(4) Label the original datum: On the basis of the clustering of the delegation nodes, label the original data objects in the candidate clusters with the index of the cluster to which the corresponding delegation nodes belong.

End

3.6 Performance Analysis

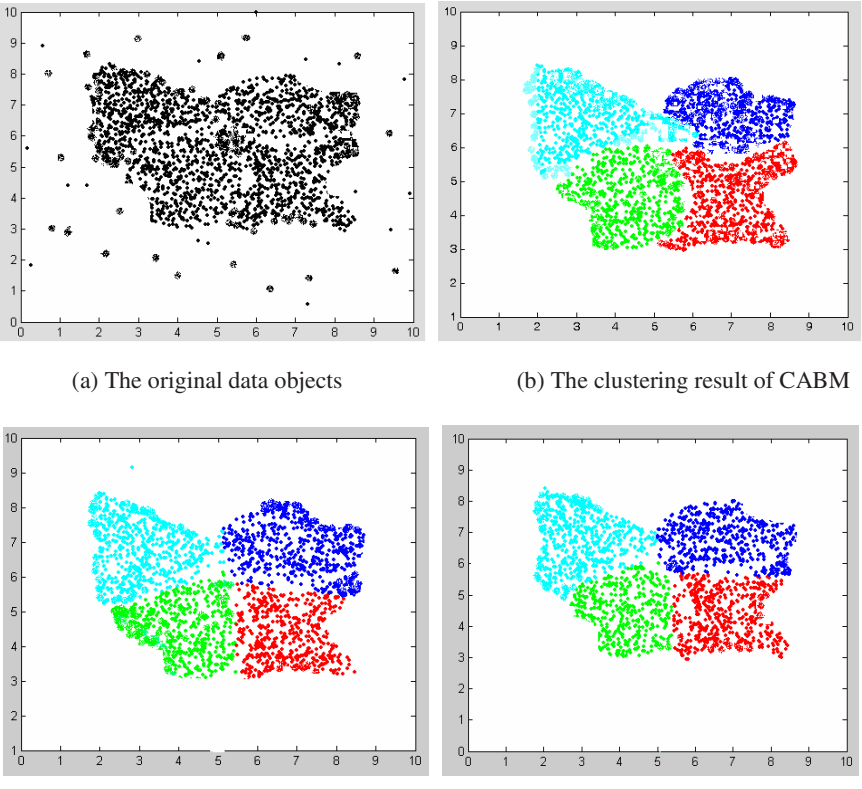
The time cost by CABM is mainly the time to compute the displacement of every delegation node and the potential energy of every truss. A cluster makes up a truss in CABM. Each cluster has $n/k\zeta_{cr}$ delegation nodes at most. Thus the time of one iteration is $O\left(k*(n/k\zeta_{cr})^3\right)$. When the clustering result is stable, the total time is $O\left(k*(n/k\zeta_{cr})^3*t\right)$, in which t is the iterative time.

4 Experimental Evaluation

We compare the performances of CABM, K-Means and CURE through two series of experiments in this session. The experiments were conducted on Windows 2000. We use engineering mechanics software Ansys 7.0 to compute the displacements of points and the potential energy of the trusses.

4.1 Clustering Results on the Same Data Sets

The first series of experiments was to compare the different clustering results of the three clustering algorithms on the same data set with $n=2000$ points on the square [0,10]×[0,10] as Fig. 3(a) shows.



(a) The original data objects

(b) The clustering result of CABM

(c) The clustering result of K-Means

(d) The clustering result of CURE

Fig. 3. The clustering results of CABM, K-Means and CURE with $n = 2000$ data objects

The clustering results of CABM, K-Means and CURE are shown in Fig. 3(b), Fig. 3(c) and Fig. 3(d) respectively. The parameters in CABM were given as $t=45, \alpha=0.11, \zeta_{cr}=1.41$. The nodes with the same color belong to one cluster in Fig. 3(b). Through the process of CABM, the number of candidate clusters decreases from 45 to 4. Fig. 3(c) shows the clustering result of K-Means with $k = 4$. Fig. 3(d) shows the result of CURE. It is clearly that the ability to explore arbitrary shape of Fig. 3(c) and Fig. 3(d) is not as good as Fig. 3(b). The four sharp angles in the right and down region of Fig. 3(b) are submerged in Fig. 3(c) and Fig. 3(d), indicating that K-Means and CURE have weak abilities than CABM to explore concave regions.

Fig. 4 represents the relationship between the number of nodes moving among the clusters and the iterative times. As the figure shows, there are almost no nodes moving among the clusters after the $11th$ iteration of CABM. While CURE and K-Means tends converge after 13 and 19 iterations. It is clear that CABM converges faster than CURE and K-Means. It follows that CABM is more efficient than CURE and K-Means.

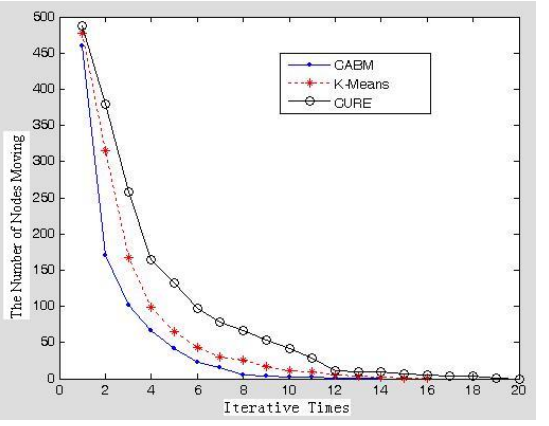


Fig. 4. The relationship between iterative times and the number of nodes moving among the clusters

4.2 Clustering Results on Different Data Sets

The series of experiments is to compare the clustering results of CABM, K-Means and CURE on different data sets. We selected seven different data sets with $n=1000,2000,\cdots,7000$. We use Q introduced in section 2 to evaluate the quality of the clustering results. The quality of the clustering results of the three algorithms on seven different data sets ranges from 1.0100 to 1.0300. Q monotonously increases as the number of the data points increases. On every data set, $Q_{\text{CABM}} < \min\{Q_{CURE}, Q_{\text{K-Means}}\}$. This indicates that the qualities of the clustering results of CABM are better than those of CURE and K-Means.

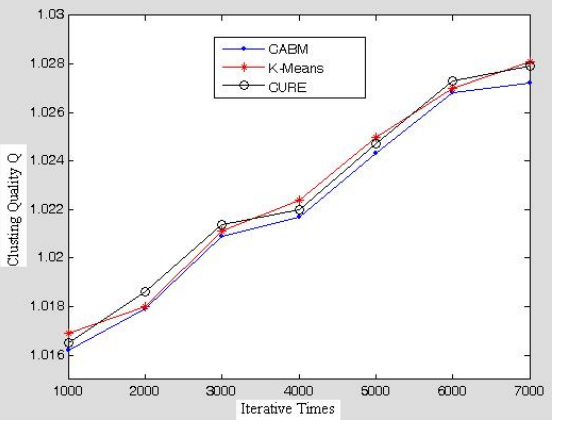


Fig. 5. Relationship between the number of data objects and the qualities of the clustering of the three algorithms

5 Conclusion

The clustering algorithm based on mechanics proposed in this paper considers the clustering problem in the view of the force and energy and transforms the theoretical problem into physical entity model. The clustering result is optimized with the theories of engineering mechanic. Experiments indicate the effectiveness of the algorithm. Future works includes studying the optimization of the model and the algorithm and trying other mechanical models.

Acknowledgements

We would like to acknowledge that the work was supported by a research grant from Chinese National Science Foundations under grant No:60503003.

References

1. GUHA, S., RASTOGI, R.,SHIM, K. CURE: An efficient clustering algorithm for large databases. In Proceedings of the ACM SIGMOD Conference (1998)73-84
2. KARYPIS, G., HAN, E.-H.,KUMAR, V. CHAMELEON:A hierarchical clustering algorithm using dynamic modeling, COMPUTER (1999)32:68-75
3. ZHANG, T., RAMAKRISHNAN, R.,LIVNY, M. BIRCH: an efficient data clustering method for very large databases. In Proceedings of the ACM SIGMOD Conference (1996)103-114
4. FISHER, D.: Knowledge acquisition via incremental conceptual clustering. Machine Learning (1987)2:139-172
5. SAVARESI, S., BOLEY, D: On performance of bisecting k-means and PDDP. In Proceedings of the 1st SIAM ICDM (2001)
6. HARTIGAN, J.: Clustering Algorithms. John Wiley & Sons (1975)
7. HARTIGAN, J., WONG, M.: A k-means clustering algorithm. Applied Statistics (1979)28:100-108
8. NG, R.,HAN, J.: Efficient and effective clustering methods for spatial data mining. In Proceedings of the 20th Conference on VLDB (1994)144-155
9. Kaufman L., Rousseeuw P.J.: Finding groups in data: An introduction to cluster analysis. New York: John Wiley & Sons (1990)
10. SHEIKHOLESLAMI, G., CHATTERJEE, S.,ZHANG, A. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In Proceedings of the 24th Conference on VLDB (1998)428-439
11. WANG, W., YANG, J., MUNTZ, R. STING: a statistical information grid approach to spatial data mining. In Proceedings of the 23rd Conference on VLDB (1997)186-195
12. AGRAWAL, R., GEHRKE, J., GUNOPULOS, D.,RAGHAVAN, P.: Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the ACM SIGMOD Conference (1998)94-105
13. ESTER, M., KRIEGEL, H-P., SANDER, J.,XU, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd ACM SIGKDD (1996)226-231

14. ANKERST, M., BREUNIG, M., KRIEGEL, H.-P.,SANDER, J. OPTICS: Ordering points to identify clustering structure. In Proceedings of the ACM SIGMOD Conference (1999) 49-60
15. XU, X., ESTER, M., KRIEGEL, H.-P.,SANDER, J.: A distribution-based clustering algorithm for mining large spatial datasets. In Proceedings of the 14th ICDE (1998) 324-331
16. Yu Jian, Cheng Qian-sheng: The searching range with the best clustering number of the fuzzy clustering methods. China Science (2002)32(2):274-280
17. Long Yu-qiu, Bao Shi-hua: Structural mechanics I,II. Higher education publishing house (2000)
18. Sun Xun-fang, Fang Xiao-shu, Guan Lai-tai: Mechanics of material (First volume and Secodn volume). Higher education publishing house (2000)

DLDA/QR: A Robust Direct LDA Algorithm for Face Recognition and Its Theoretical Foundation

Yu-Jie Zheng[1], Zhi-Bo Guo[1], Jian Yang[2], Xiao-Jun Wu[3], and Jing-Yu Yang[1,*]

[1] Department of Computer Science, Nanjing University of Science and Technology, Nanjing 210094, P.R. China
yjzheng13@yahoo.com.cn,yangjy@mail.njust.edu.cn,
zhibo_guo@163.com
[2] Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong
csjyang@comp.polyu.edu.hk
[3] School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang 212003, P.R. China
wu_xiaojun@yahoo.com.cn

Abstract. Feature extraction is one of the hot topics in face recognition. However, many face extraction methods will suffer from the "small sample size" problem, such as Linear Discriminant Analysis (LDA). Direct Linear Discriminant Analysis (DLDA) is an effective method to address this problem. But conventional DLDA algorithm is often computationally expensive and not scalable. In this paper, DLDA is analyzed from a new viewpoint via QR decomposition and an efficient and robust method named DLDA/QR algorithm is proposed. The proposed algorithm achieves high efficiency by introducing the QR decomposition on a small-size matrix, while keeping competitive classification accuracy. Experimental results on ORL face database demonstrate the effectiveness of the proposed method.

1 Introduction

Face Recognition (FR) has a wide range of applications, such as military, commercial and law enforcement et al. Within the past two decades, numerous FR [1-3],[6-16] algorithms have been proposed. Among these FR methods, the most popular methods are appearance-based approaches.

Of the appearance-based FR methods, those utilizing Linear Discriminant Analysis (LDA) [4-16] techniques have shown promising results. Conventional LDA [4,5] algorithm aims to find an optimal transformation by minimizing the within-class scatter matrix and maximizing the between class scatter matrix simultaneously. The optimal transformation is readily computed by applying the eigen-decomposition to the scatter matrices. But an intrinsic limitation of conventional LDA is that its objective function

* This work was supported by NSF of China (60632050, 60472060, 60473039, 60503026 and 60572034).

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 379–387, 2007.

requires the within-class scatter matrix nonsingular. For many real applications, the within-class scatter matrix is often singular since the dimension of sample exceeds the number of sample and conventional LDA based methods suffer from the so-called "small sample size" [6-8] problem.

In the last decades, numerous methods have been proposed to solve this problem. Tian et al [9] used the pseudoinverse method by replacing inverse of within-class scatter matrix with its pseudoinverse. The perturbation method is used in [10], where a small perturbation matrix is added to within-class scatter matrix in order to make it nonsingular. Cheng et al [11] proposed the Rank Decomposition method based on successive eigen-decomposition of the total scatter matrix and the between-class scatter matrix. However, the above methods are typically computationally expensive since the scatter matrices are very large. Swets and Weng [12] proposed a two stages PCA+LDA method, also known as the Fisherface [6] method, in which PCA is first used for dimension reduction so as to make the within-class scatter matrix nonsingular before the application of LDA. By far, the PCA+LDA method is popular used. However, algorithms based on this solution may discard effective features in PCA step. To prevent this from happening, many extended LDA algorithms with null space conception were proposed. Chen et al [13] and Yang et al [14] developed DLDA algorithm for face recognition, which can effectively solve "small sample size" problem and extract optimal classification features from original samples. But conventional DLDA [13-15] algorithm is often computationally expensive and not scalable.

In this paper, first of all, we briefly recall the DLDA algorithm. Then we perform an in-depth analysis on DLDA algorithm and proposed a DLDA/QR algorithm. The utilization of the QR [16-17] decomposition on the small-size matrix is one of key steps. Thus we can implement the second stage of DLDA in a low dimensional space. Hence the proposed method is efficient and robust. Moreover, the theoretical foundation of the proposed method is revealed.

2 Outline of Direct LDA

Throughout the paper, C denotes the number of classes, m is the dimension, N is the number of samples in each class, μ_i is the centroid of the ith class, and μ is the holistic centroid of the whole data set. S_b, S_w and S_t represent between-class scatter matrix, within-class scatter matrix and total class scatter matrix, respectively.

DLDA [13-15] algorithm was proposed by Chen and Yang, which attempts to avoid the shortcomings existing in conventional solution to the "small sample size" problem. The basic idea behind the DLDA algorithm is that the null space of S_w may contain effective discriminant information if the projection of S_b is not zero in that direction, and that no effective information will be lost if the null space of S_b is discarded. For

example, assuming that N_b and N_w represent the null space of S_b and S_w, respectively, the complement spaces of N_b and N_w can be written as $N_b^{'} = R^n - N_b$ and $N_w^{'} = R^n - N_w$. Therefore, the optimal discriminant subspace extracted by the DLDA algorithm is the intersection space $N_b^{'} \cap N_w$.

The difference between Chen's method and Yang's method is that Yang's method first diagonalizes S_b to find $N_b^{'}$, while Chen's method first diagonalizes S_w to find N_w. Although there is no significant difference between the two approaches, it may be intractable to calculate N_w when the size of S_w is large, which is the case in most FR application. Therefore, we adopted Yang's method as the derivation of the proposed algorithm.

3 DLDA/QR Algorithm for Dimension Reduction and Feature Extraction

In this section, we will present the DLDA/QR algorithm. Based on the analysis of DLDA algorithm, the proposed algorithm can be realized through two stages. The first stage is the maximum separability among different classes obtained by QR-decomposition. The second stage contains LDA algorithms that involve the concern of within-class distance.

The first stage aims to solve the following optimization problem:

$$G = \arg\max_{G^T G = I} trace\left(G^T S_b G\right) \tag{1}$$

From Eq.(1), we can find this optimization problem only gives the concern on maximizing between-class scatter matrix. The solution can be obtained by solving the eigenproblem on S_b similar to PCA algorithm. However, the solution to Eq.(1) can also be obtained through the QR-decomposition on the matrix H_b as follows, where

$$H_b = [\sqrt{N}(\mu_1 - \mu) \cdots \sqrt{N}(\mu_C - \mu)] \tag{2}$$

and satisfies

$$S_b = H_b H_b^T \tag{3}$$

Let

$$H_b = (Q_1, Q_2)\begin{pmatrix} R \\ 0 \end{pmatrix} \tag{4}$$

be the QR decomposition of H_b, where $Q_1 \in X^{m\times t}$, $Q_2 \in X^{m\times(m-t)}$, $R \in X^{t\times L}$, and $t = rank(S_b)$. It is easy to verify that $H_b = Q_1 R$ is a full-rank decomposition of H_b. Then $G = Q_1 W$, for any orthogonal matrix $W \in R^{t\times t}$, solves the optimization problem in Eq.(1). Note the rank t of the matrix H_b, is bounded above by $C-1$. In practice, the C centroids in the data set are usually linearly independent. In this case, the reduced dimension $t = C-1 = rank(S_b)$.

The second stage of DLDA/QR algorithm will concern the within-class distance. In this stage, the optimization problem is exactly the same one as in classical DLDA, but with matrixes of much smaller size, hence can be solved efficiently and stably. After we have obtained the matrix Q_1, we assume that $\tilde{S}_t = Q_1 S_t Q_1$ and $\tilde{S}_b = Q_1 S_b Q_1$. In this case, it is easy to verify that both $\tilde{S}_t$ and $\tilde{S}_b$ are $t\times t$ matrices. In addition, it should be noticed that the matrix $\tilde{S}_b$ is nonsingular.

In this stage, we should find a matrix that simultaneously diagonalizes both $\tilde{S}_b$ and $\tilde{S}_t$:

$$V^T \tilde{S}_t V = \Lambda,\ V^T \tilde{S}_b V = I \tag{5}$$

Where Λ is a diagonal matrix whose diagonal elements are sorted in increasing order and I is a unitary matrix.

In simultaneous diagonalization mentioned above, first, we diagonalize the symmetric matrix $\tilde{S}_b$. Since the dimension of the matrix $\tilde{S}_b$ is $t\times t$, generally $t << m$, it is easy to diagonalize the matrix.

Assume that there exists the matrix U such that

$$U^T \tilde{S}_b U = \Lambda_b \tag{6}$$

where $U^T U = I$ and Λ_b is a diagonal matrix whose diagonal elements are sorted in decreasing order.

Let

$$Z = U\Lambda_b^{-1/2} \tag{7}$$

Then we can obtain

$$\left(U\Lambda_b^{-1/2}\right)^T \tilde{S}_b U\Lambda_b^{-1/2} = I \Rightarrow Z^T \tilde{S}_b Z = I \tag{8}$$

Next, let

$$\hat{S}_t = Z^T \tilde{S}_t Z \tag{9}$$

In a similar way, we can diagonalize the matrix $\hat{S}_t$.

Assume that there exists the matrix Y such that

$$Y^T \hat{S}_t Y = \Lambda_t \tag{10}$$

where $Y^T Y = I$ and Λ_t is a diagonal matrix whose diagonal elements are sorted in increasing order.

Ordinarily, we select s ($s \leq t$) eigenvectors corresponding to the first s smallest eigenvalues. Assume that s eigenvectors constitute the matrix $P = (y_1, \cdots, y_i, \cdots, y_s)$, where y_i is the ith column of the matrix Y. Then we obtain

$$P^T \hat{S}_t P = \Lambda_s \tag{11}$$

Let

$$V_s = ZP \tag{12}$$

Then, we can obtain

$$V_s^T \tilde{S}_t V = P^T Z^T \tilde{S}_t ZP = P^T \hat{S}_t P = \Lambda_s \tag{13}$$

And

$$V_s^T \tilde{S}_b V = P^T Z^T \tilde{S}_b ZP = P^T IP = I \tag{14}$$

Therefore, we obtain the matrix $V_s = ZP$ simultaneously diagonalizes both $\tilde{S}_b$ and $\tilde{S}_t$. Certainly, $V = ZY$ also satisfies this condition.

Thus, we can obtain the following transform matrix:

$$E = Q_1 ZP \Lambda_s^{-1/2} \tag{15}$$

where E is a $m \times s$ matrix.

To a test image x_{test}, the feature of this test image is found by

$$\Omega_{test} = E^T x_{test} \tag{16}$$

which can be used to classify.

Furthermore, the aforementioned simultaneous diagonalization can be further simplified by the following theorem [4].

Theorem 1. We can diagonalize two symmetric matrices $\widetilde{S}_t$ and $\widetilde{S}_b$ as $V^T \widetilde{S}_t V = \Lambda$, $V^T \widetilde{S}_b V = I$, where V and Λ are the eigenvector and eigenvalue matrix of $\widetilde{S}_b^{-1} \widetilde{S}_t$ and satisfy $\widetilde{S}_b^{-1} \widetilde{S}_t V = \Lambda V$.

Assume that the diagonal elements of the matrix Λ are sorted in increasing order. Accordingly, the final transform matrix is $E = Q_1 V_s \Lambda_s^{-1/2}$, where V_s is a $t \times s$ matrix that consists of the first s column vectors of V and Λ_s is a $s \times s$ matrix that is obtained by the matrix Λ.

Based on the above discussion, the proposed DLDA/QR algorithm is described as follows:

***Step* 1:** Obtain H_b, S_b and S_t, and calculate the QR decomposition of H_b. Let $\widetilde{S}_t = Q_1 S_t Q_1$ and $\widetilde{S}_b = Q_1 S_b Q_1$.

***Step* 2:** Calculate the eigenvector matrix and the eigenvalue matrix of $\left(\widetilde{S}_b\right)^{-1} \widetilde{S}_t$, denote by V and Λ. Assume the first s ($s \le t$) eigenvectors corresponding to the first s smallest eigenvalues that the s eigenvectors and eigenvalues form the matrix V_s and Λ_s, respectively. Hence we can obtain the final transform matrix $E = Q_1 V_s \Lambda_s^{-1/2}$.

***Step* 3:** Project samples into subspace according to Eq.(16) and classify.

4 Experimental Results

To demonstrate the effectiveness of our method, experiments were done on the ORL face database (http://www.uk.research.att.com/facedatabase.html). Fig.1 depicts some images from the ORL face database.

Fig. 1. Some samples from the ORL face database

Table 1. Recognition rates (%) on the ORL face database with Euclidean distance (mean and standard deviation)

# Training sample / class (ϑ)	4	5	6
Fisherface	89.85 ± 1.88	92.90 ± 1.37	93.13 ± 1.02
DLDA	92.58 ± 1.33	95.00 ± 0.75	95.10 ± 0.78
DLDA/QR	92.71 ± 1.26	96.00 ± 0.47	96.56 ± 1.11

Table 2. Recognition rates (%) on the ORL face database with cosine distance (mean and standard deviation)

# Training sample / class (ϑ)	4	5	6
Fisherface	91.62 ± 0.64	93.25 ± 1.55	93.31 ± 1.56
DLDA	92.62 ± 1.02	94.10 ± 1.58	94.75 ± 1.54
DLDA/QR	93.05 ± 0.83	95.50 ± 1.29	96.31 ± 0.62

In our experiments, the training and testing set are selected randomly from each subject. In each round, the training samples are selected randomly from the gallery and the remaining samples are used for testing. This procedure was repeated 10 times by randomly choosing different training and testing sets. The number of training samples per subject, ϑ, increases from 4 to 6 and the number of final discriminant vectors is 39 (i.e. $C-1$). After feature extraction, a nearest neighbor classifier with different distance metrics is employed for classification. Two distance metrics: Euclidean distance metric and cosine distance are used in our experiments.

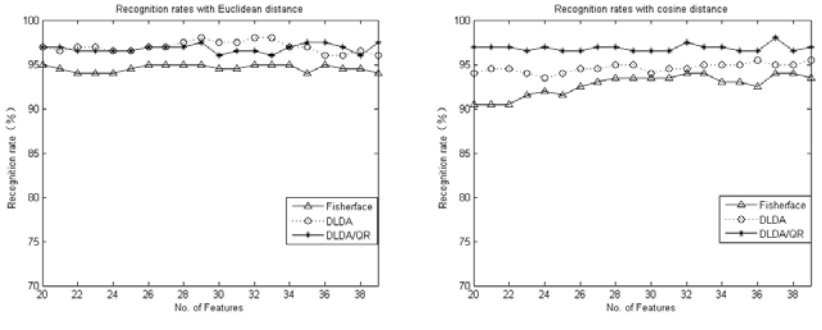


Fig. 2. Comparison of recognition rates with different features numbers ($\vartheta = 5$). Top: recognition rate with Euclidean metric. Bottom: recognition rate with cosine metric.

For each distance metric, mean and standard deviation of Fisherface, DLDA and DLDA/QR are listed in Table 1 and Table 2. From these results, we can conclude that the performance of the DLDA/QR algorithm is as well as (even slightly better than) that of conventional DLDA algorithm and superior to that of Fisherface algorithm. Then, recognition rates with different features numbers are shown in Fig.2. In this figure, as the features numbers varying from 20 to 39 with number of training samples per subject equals to 5 and number of classes equals to 40, the recognition rates are depicted. From this figure, we can conclude that the DLDA/QR algorithm is robust and stable, especially with cosine distance metric. Results with these experiments demonstrate that, with the QR decomposition, the DLDA algorithm becomes efficiency and scalability. Furthermore, after QR decomposition, the second stage of the proposed algorithm can implement in a low dimension space, which avoids handling large matrices and improves the stability of the computation.

5 Conclusions

In this paper, we proposed an extension of direct linear discriminant analysis algorithm, namely, DLDA/QR algorithm, which is highly efficient and scalable. The proposed method does not require the whole data matrix in main memory. This is desirable for large data sets. In addition, our theoretical analysis indicates that the computational complexity of the DLDA/QR algorithm is linear in the number of the data items in the training data set as the number of classes and the number of dimensions. It is the QR decomposition that contributes to the efficiency and scalability of the DLDA/QR algorithm, which is not only shown by our theoretical analysis, but also strongly supported by our experimental results.

Our experiments on face database have shown that the accuracy achieved by the DLDA/QR algorithm is competitive with the ones achieved by DLDA and Fisherface algorithm. With efficiency and scalability, DLDA/QR algorithm is promising in real-time application involving extremely high-dimensional data.

References

1. Samal A and Iyengar P A. (1992) Automatic recognition and analysis of human faces and facial expressions: a survey. Pattern Recognition 25(1): 65-77
2. W. Zhao, R. Chellappa, and J. Phillips (1999) Subspace linear discriminant analysis for face recognition. Technical Report, CS-TR4009, Univ. of Maryland
3. W.Zhao, R.Chellappa, P.J.Phillips, A.Rosenfeld. (2003) Face recognition: A literature survey. ACM Computing Surveys. 35(4):395-458
4. K.Fukunaga (1990) Introduction to statistical pattern recognition. Academic Press, Boston, 2nd edition
5. R. Fisher (1936) The use of multiple measures in taxonomic problems. Ann. Eugenics. 7:179-188
6. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriengman (1997) Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. Pattern Analysis and Machine Intelligence. 19 (7):711-720

7. J. Yang, A.F. Frangi, J.Y. Yang, D. Zhang, Z. Jin (2005) KPCA plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature extraction and Recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 27(2):230-244
8. J. Yang, J.Y. Yang (2003) Why can LDA be performed in PCA transformed space?. Pattern Recognition 36:563-566
9. Q.Tian, M.Barbero, Z.Gu, S.Lee (1986) Image classification by the foley-sammon transform". Opt. Eng. 25(7):834-840
10. Z.Q.Hong, J.Y.Yang (1991) Optimal discriminant plane for a small number of samples and design method of classifier on the plane. Pattern Recognition 24(4):317-324
11. Y.Q. Cheng, Y.M.Zhuang, J.Y.Yang (1992) Optimal fisher discriminant analysis using the rank decomposition. Pattern Recognition 25(1):101-111.
12. Swets and J. Weng (1996) Using discriminant eigenfeatures for image retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence 18(8):831-836
13. L.F.Chen, H.Y.M.Liao, J.C.Lin, M.T.Ko, and G.J.Yu. A New LDA-based Face Recognition System Which Can Solve the Small Sample Size Problem, Pattern Recognition, 2000, 33(10), pp.1713-1726.
14. J. Yang, H. Yu, W. Kunz (2000) An Efficient LDA Algorithm for Face Recognition. International Conference on Automation, Robotics, and Computer Vision (ICARCV'2000), Singapore, December
15. Yujie Zheng, Jingyu Yang, Jian Yang, Xiaojun Wu (2006) Effective classification image space which can solve small sample size problem. Proc. Of. the 18th Int. Conf. on Pattern Recognition (ICPR'06), vol.3, pp.861-864
16. J.P.Ye, Q.Li (2005) A Two-Stage Linear Discriminant Analysis via QR-Decomposition. IEEE Trans. Pattern Anal. Machine Intell. 2(6): 929-941
17. G.H.Golub and C.F.Van Loan (1996) Matrix Computations, third ed. Baltimore, M.D.: The Johns Hopkins Univ. Press

gPrune: A Constraint Pushing Framework for Graph Pattern Mining

Feida Zhu†, Xifeng Yan†, Jiawei Han†, and Philip S. Yu‡

† Computer Science, UIUC
{feidazhu,xyan,hanj}@cs.uiuc.edu
‡ IBM T. J. Watson Research Center
psyu@us.ibm.com

Abstract. In graph mining applications, there has been an increasingly strong urge for imposing user-specified constraints on the mining results. However, unlike most traditional itemset constraints, structural constraints, such as density and diameter of a graph, are very hard to be pushed deep into the mining process.

In this paper, we give the first comprehensive study on the pruning properties of both traditional and structural constraints aiming to reduce not only the pattern search space but the data search space as well. A new general framework, called **gPrune**, is proposed to incorporate all the constraints in such a way that they recursively reinforce each other through the entire mining process. A new concept, *Pattern-inseparable Data-antimonotonicity*, is proposed to handle the structural constraints unique in the context of graph, which, combined with known pruning properties, provides a comprehensive and unified classification framework for structural constraints. The exploration of these antimonotonicities in the context of graph pattern mining is a significant extension to the known classification of constraints, and deepens our understanding of the pruning properties of structural graph constraints.

1 Introduction

Graphs are widely used to model complicated structures in many scientific and commercial applications. Frequent graphs, those occurring frequently in a collection of graphs, are especially useful in characterizing graph sets, detecting network motifs [2], discriminating different groups of graphs [3], classifying and clustering graphs [4,5,6], and building graph indices [7]. For example, Huan *et al.* [5] successfully applied the frequent graph mining technique to extract coherent structures and used them to identify the family to which a protein belongs. Yan *et al.* [7] chose discriminative patterns from frequent graphs and applied them as indexing features to achieve fast graph search.

Unfortunately, general-purpose graph mining algorithms cannot fully meet users' demands for mining patterns with their own constraints. For example, in computational biology, a highly connected subgraph could represent a set of genes within the same functional module [8]. In chem-informatics, scientists are often interested in frequent graphs that contain a specific functional fragment, *e.g.*, a benzine ring. In all these applications, it is critical for users to have control on certain properties of the mining results for them to be meaningful. However, previous studies have left open the problem

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 388–400, 2007.

of pushing sophisticated structural constraints to expedite the mining process. This gap between user demand and the capability of current known mining strategies calls for a constraint-based mining framework that incorporates these structural constraints.

Related Work. A number of efficient algorithms for frequent graph mining are available in data mining community, *e.g.*, AGM [14], FSG [15], the path-join algorithm [16], gSpan[17], MoFa[3], FFSM [18], SPIN[19] and Gaston [20]. Few of them considered the necessary changes of the mining framework if structural constraints are present. Constraint-based frequent pattern mining has been studied in the context of association rule mining by Ng *et al.* [9], which identifies three important classes of constraints: *monotonicity, antimonotonicity* and *succinctness* and develops efficient constraint-based frequent itemset mining algorithms for a single constraint. Ng *et al.* also pointed out the importance of exploratory mining of constrained association rules so that a user can be involved in the mining process. Other complicated constraints, such as gradients [26], block constraints [27], constraints on sequences [28], and connectivity constraints [23], are proposed for different applications. Pei *et al.* discovers another class of constraint *convertible constraints* and its pushing methods. Constrained pattern mining for graphs has been looked into by Wang *et al.*, [29], although only constraints with monotonicity/antimonotonicity/succinctness are discussed. Bucila *et al.* [10] introduced a DualMiner framework that simultaneously takes advantage of both monotonicity and antimonotonicity to increase mining efficiency. The general framework of these mining methods is to push constraints deep in order to prune pattern search space. Although this is effective in many cases, the greatest power of constraint-based frequent pattern mining is achieved only when considering together the reduction on both the pattern search space and the data search space. Bonchi *et al.*have taken successful steps in this direction by proposing ExAnte, [11,12,13], a pre-processor to achieve data reduction in constrained itemset mining. ExAnte overcame the difficulty of combining the pruning power of both anti-monotone and monotone constraints, the latter of which had been considered hard to exploit without compromising the anti-monotone constraints. Boulicaut and De Raedt [1] have explored constraint-based mining as a step towards inductive databases.

Our Contributions. In this paper we show that the data reduction technique can in fact be extended beyond the preprocessing stage as in ExAnte, and pushed deeper into the mining algorithm such that the data search space is shrunk recursively each time it is projected for a pattern newly grown, through the entire mining process. More importantly, our study of graph constraints shows that for sophisticated constraints in data sets whose structures are more complicated than itemsets, data space pruning could be effective only when the structural relationship between the embedded pattern and the data is taken into account. This new constraint property, which we term as *Pattern-inseparable D-antimonotonicity*, is unique in the context of graphs and, to our best knowledge, has not been explored before in literature. It distinguishes itself from other pruning properties in that most sophisticated structural constraints, *e.g.*, diameter, density, and connectivity, exhibit neither antimonotonicity nor monotonicity. Without exploiting Pattern-inseparable D-antimonotonicity, current mining algorithms would have to enumerate all frequent graphs in the first place and then check constraints on them

one by one. The paper makes the following contributions: First, we present the first systematic study of the pruning properties for complicated structural constraints in graph mining which achieves pruning on both pattern and data spaces. The full spectrum of pruning power is covered by (1) extending the known antimonotonicities for itemsets to easier cases and (2) discovering novel pattern-separable and pattern-inseparable D-antimonotonicities to handle structural constraints when pattern embeddings have to be considered. Secondly, a general mining framework is proposed that incorporates these pruning properties in graph pattern mining. In particular, data space pruning is coupled tightly with other constraint-based pruning such that data reduction is exploited throughout the entire mining process. Thirdly, discussion is given on mining strategy selection when a trade-off has to be made between the naive enumerating-and-checking approach and our pruning-property-driven approach.

2 Preliminaries

As a convention, the *vertex set* of a graph P is denoted by $V(P)$ and the *edge set* by $E(P)$. For two graphs P and P', P is a subgraph of P' if there exists a subgraph isomorphism from P to P', denoted by $P \subseteq P'$. P' is called a supergraph of P. In graph mining, a pattern is itself a graph, and will also be denoted as P. Given a set of graphs $D = \{G_1, G_2, \ldots, G_n\}$, for any pattern P, the *support database of* P is denoted as D_P, and is defined as $D_P = \{G_i | P \subseteq G_i, 1 \leq i \leq n\}$. D_P is also referred to as the *data search space* of P, or *data space* for short. A graph pattern P is *frequent* if and only if $\frac{|D_P|}{|D|} \geq \sigma$ for a support threshold σ.

A constraint C is a boolean predicate on the pattern space U. Define $f_C : U \rightarrow \{0, 1\}$ as the corresponding boolean function of C such that $f_C(P) = 1, P \in U$ if and only if P satisfies the constraint C. For example, let C be the constraint $Max\_Degree(P) \geq 10$ for a graph pattern P. Then $f_C(P) = 1$ if and only if the maximum degree of all the vertices of P is greater than 10. We formulate the constraint-based frequent graph pattern mining problem as the following:

Definition 1. (Constraint-based Frequent Graph Pattern Mining) *Given a set of graphs* $D = \{G_1, G_2, \ldots G_n\}$*, a support threshold* σ*, and a constraint* C*, constraint-based frequent graph pattern mining is to find all* P *such that* $\frac{|D_P|}{|D|} \geq \sigma$ *and* $f_C(P) = 1$.

Here are some graph constraints used in this paper: (1) The *density ratio* of a graph P, denoted as $Density\_Ratio(P)$, is defined as $Density\_Ratio(P) = \frac{|E(P)|}{|V(P)|(|V(P)|-1)/2}$. (2) The *Density* of a graph P is defined as $Density(P) = \frac{|E(P)|}{|V(P)|}$. (3) The *Diameter* of a graph P is the maximum length of the shortest path between any two vertices of P. (4) $EdgeConnectivity(P)$($VertexConnectivity(P)$) is the minimum number of edges(vertices) whose deletion from P disconnects P.

3 Pattern Mining Framework

gPrune can be applied to both Apriori-based model and the pattern-growth model. In this paper, we take the pattern-growth model as an example to illustrate the pruning optimizations. Nevertheless, the techniques proposed here can also be applied to

Apriori-based methods. The pattern-growth graph mining approach is composed of two stages (1) pattern seed generation (2) pattern growth. The mining process is conducted by iterating these two stages until all frequent patterns are found.

Pattern Seed Generation: We use gSpan[17] to enumerate all the seeds with size of increasing order. One pattern seed is generated every time and proceeds to the pattern growth stage. A pattern seed could be a vertex, an edge, or a small structure.

Pattern Growth: As outlined in Algorithm 1, PatternGrowth keeps a set S of pattern seeds and a set F of frequent patterns already mined. Each iteration of $PatternGrowth$ might generate new seeds (added to S), and identify new frequent patterns (added to F). Line 1 initializes the data structures. Initially, S contains only the pattern seed. Line 2 to 11 grow every pattern seed in S until S is exhausted. For each pattern seed Q, which is taken from the set S in Line 3, Q is checked through its data search space and augmented incrementally by adding a new edge or vertex (Lines 4 and 5). Each augmented pattern is checked for pattern pruning in Line 6 and dropped whenever it satisfies the pattern pruning requirements. All the augmented patterns that survive the checking are recorded in S_t. Then in Line 8, for each surviving pattern, we construct its own support data space from that of Q's. Line 9 checks data pruning for each G in the support space. Since each thus augmented pattern is a frequent pattern, we add them to F in Line 10. Finally, these patterns are added to S, so that they will be used to grow new patterns. When S is exhausted, a new pattern seed will be generated until it is clear that all frequent patterns are discovered. The algorithm input: A frequent pattern seed P, graph database $D = \{G_1, G_2, \ldots, G_n\}$ and the existing frequent pattern set F. The algorithm output: New frequent pattern set F.

Algorithm 1. PatternGrowth

1: $S \leftarrow \{P\}; F \leftarrow F \bigcup \{P\}; S_t \leftarrow \emptyset$
2: **while** $S \neq \emptyset$;
3: $Q \leftarrow pop(S)$;
4: **for each** graph $G \in D_Q$
5: Augment Q and save new patterns in S_t;
6: Check pattern pruning on each $P \in S_t$;
7: **for each** augmented pattern $Q' \in S_t$
8: Construct support data space $D_{Q'}$ for Q';
9: Check data pruning on $D_{Q'}$;
10: $F \leftarrow F \bigcup S_t$;
11: $S \leftarrow S \bigcup S_t$;
12: **return** F;

PatternGrowth checks for pattern pruning in Line 6 and data pruning in Line 9. Pattern pruning is performed whenever a new augmented pattern is generated. This means any unpromising pattern will be pruned before constructing its data search space. Notice that data pruning is performed whenever infrequent edges are dropped after a

new data search space is constructed and offers chance to drop new target graphs. As such, the search space for a pattern keeps shrinking as the pattern grows.

4 Pruning Properties

A pruning property is a property of the constraint that helps prune either the pattern search space or the data search space. Pruning properties which enable us to prune patterns are called *P-antimonotonicity*, and those that enable us to prune data are called *D-antimonotonicity*.

4.1 Pruning Patterns

(1) Strong P-antimonotonicity

Definition 2. *A constraint C is **strong P-antimonotone** if $f_C(P') = 1 \rightarrow f_C(P) = 1$ for all $P \subseteq P'$.*

Strong P-antimonotonicity is simply the antimonotone property which has been known long since [9]. We call it strong P-antimonotonicity only to distinguish it from the other P-antimonotonicity introduced below. An example of strong P-antimonotone constraint for graph is acyclicity.

(2) Weak P-antimonotonicity

Constraints like "$Density\_Ratio(G) \geq 0.1$" is not strong P-antimonotone. Growing a graph G could make $Density\_Ratio(G)$ go either up or down. However, they have *weak P-antimonotonicity*, which is based on the following intuition. If a constraint C is not strong P-antimonotone, then there must exist a pattern P violating C and a supergraph of P, say Q, that satisfies C. In this case, we cannot prune graph P even if P violates C because Q might be missed if Q can only be grown out of P. However, if we can guarantee that Q can always be grown from some other subgraph P' such that P' satisfies C, we can then safely prune P.

Definition 3. *A constraint C is **weak P-antimonotone** if for a graph P' where $|V(P')| \geq k$ for some constant k, $f_C(P') = 1 \rightarrow f_C(P) = 1$ for some $P \subset P'$, such that $|V(P)| = |V(P')| - 1$.*

k is the size of the minimum instance to satisfy the constraint. When mining for weak P-antimonotone constraints, since we are sure that, for any constraint-satisfying pattern Q, there is a chain of substructures such that $g_1 \subset g_2 \subset ... \subset g_n = Q$ and g_i satisfies the constraint for all $1 \leq i \leq n$, we can drop a current pattern P if it violates the constraint, even if some supergraph of P might satisfy the constraint. Weak P-antimonotonicity allows us to prune patterns without compromising the completeness of the mining result. A similar property on itemsets, "loose antimonotonicity", has been discussed by Bonchi *et al.*in [13]. Notice that if a constraint is strong P-antimonotone, it is automatically weak P-antimonotone; but not vice versa. Also note that we can have similar definition of weak P-animonotonicity with the chain of substructure decreasing in number of edges.

We us the graph density ratio example to illustrate the pruning. The proof of the following theorem is omitted due to space limit.

Theorem 1. *Given a graph G, if $Density\_Ratio(G) > \delta$, then there exists a sequence of subgraphs $g_3, g_4, \ldots, g_n = G$, $|V(g_i)| = i$ $(3 \leq i \leq n)$ such that $g_3 \subset g_4 \subset \ldots g_n$ and $Density\_Ratio(g_i) > \delta$.*

Theorem 1 shows a densely connected graph can always be grown from a smaller densely connected graph with one vertex less. As shown in this example of graph density ratio, even for constraints that are not strong P-antimonotone, there is still pruning power to tap if weak P-antimonotonicity is available.

4.2 Pruning Data

(1) Pattern-separable D-antimonotonicity

Definition 4. *A constraint C is **pattern-separable D-antimonotone** if for a pattern P and a graph $G \in D_P$, $f_C(G) = 0 \rightarrow f_C(P') = 0$ for all $P \subseteq P' \subseteq G$.*

For constraints with pattern-separable D-antimonotonicity, the exact embeddings of the pattern are irrelevant. Therefore, we only need to check the constraint on the entire graphs in the pattern's data search space, and safely drop a graph if it fails the constraint.

Consider the constraint *"the number of edges in a pattern is greater than* 10*"*. The observation is that every time a new data search space is constructed for the current pattern P, we can scan the graphs in the support space and prune those with less than 11 edges.

It is important to recognize that this data reduction technique can be applied repeatedly in the entire mining process, instead of applying in an initial scan of the database as a preprocessing procedure. It is true that we will not benefit much if this data pruning is effective only once for the original data set, *i.e.*, if any graph surviving the initial scanning will always survive in the later pruning. The key is that ***in our framework, data pruning is checked on every graph in the data search space each time the space is updated for the current pattern. As such, a graph surviving the initial scan could still be pruned later.*** This is because when updating the search space for the current pattern P, edges which were frequent at last step could now become infrequent, and are thus dropped. This would potentially change each graph in the data search space, and offer chance to find new graphs with less than 11 edges which become eligible for pruning *only* at this step. Other examples of pattern-separable D-antimonotonic constraints include *path/feature containment*, e.g., *pattern contains three benzol rings*.

(2) Pattern-inseparable D-antimonotonicity

Unfortunately, many constraints in practice are not pattern-separable D-antimonotone. $VertexConnectivity(P) > 10$ is a case in point. The exact embedding of the pattern is critical in deciding whether it is safe to drop a graph in the data search space. These constraints are thus pattern-inseparable. In these cases, if we "*put the pattern P back to G*", *i.e.*, considering P together with G, we may still be able to prune the data search space.

Definition 5. *A constraint C is **pattern-inseparable D-antimonotone** if for a pattern P and a graph $G \in D_P$, there exists a measure function $M : \{P\} \times \{G\} \rightarrow \{0, 1\}$ such that $M(P, G) = 0 \rightarrow f_C(P') = 0$ for all $P \subseteq P' \subseteq G$.*

Constraint	strong P-antimonotone	weak P-antimonotone	pattern-separable D-antimonotone	pattern-inseparable D-antimonotone
$Min\_Degree(G) \geq \delta$	No	No	No	Yes
$Min\_Degree(G) \leq \delta$	No	Yes	No	Yes
$Max\_Degree(G) \geq \delta$	No	No	Yes	Yes
$Max\_Degree(G) \leq \delta$	Yes	Yes	No	Yes
$Density\_Ratio(G) \geq \delta$	No	Yes	No	Yes
$Density\_Ratio(G) \leq \delta$	No	Yes	No	Yes
$Density(G) \geq \delta$	No	No	No	Yes
$Density(G) \leq \delta$	No	Yes	No	Yes
$Size(G) \geq \delta$	No	Yes	Yes	Yes
$Size(G) \leq \delta$	Yes	Yes	No	Yes
$Diameter(G) \geq \delta$	No	Yes	No	Yes
$Diameter(G) \leq \delta$	No	No	No	Yes
$EdgeConnectivity(G) \geq \delta$	No	No	No	Yes
$EdgeConnectivity(G) \leq \delta$	No	Yes	No	Yes
G contains P (e.g., P is a benzol ring)	No	Yes	Yes	Yes
G does not contain P (e.g., P is a benzol ring)	Yes	Yes	No	Yes

Fig. 1. A General Picture of Pruning Properties of Graph Constraints

The idea of using pattern-inseparable D-antimonotone constraints to prune data is the following. After embedding the current pattern P into each $G \in D_P$, we compute by a measure function, for all supergraphs P' such that $P \subset P' \subset G$, an upper/lower bound of the graph property to be computed in the constraint. This bound serves as a necessary condition for the existence of a constraint-satisfying supergragh P'. We discard G if this necessary condition is violated. For example, suppose the constraint is $VertexConnectivity(P) > 10$. If after embedding P in G, we find that the maximum vertex connectivity of all the supergraphs of P is smaller than 10, then no future pattern growing out of P in G will ever satisfy the constraint. As such G can be safely dropped. The measure function used to compute the bounds depends on the particular constraint. For some constraints, the computational cost might be prohibitively high and such a computation will not be performed. Another cost issue associated with pruning based on pattern-inseparable D-antimonotonicity is the maintenance of the pattern growth tree to track pattern embeddings. The Mining algorithm has to make a choice based on the cost of the pruning and the potential benefit. More discussion on the trade-off in these cases is given in Section 5. We use the vertex connectivity as an example to show how to perform data pruning. The time cost is linear in the pattern size for this constraint.

Let $Neighbor(P)$ be the set of vertices adjacent to pattern P. For the vertex connectivity constraint, the following lemma gives a necessary condition for the existence of a P' such that $VertexConnectivity(P') \geq \delta$.

Lemma 1. *If $|Neighbor(P)| < \delta$, then there exists no P' such that $P \subset P' \subset G$ and $VertexConnectivity(P') > \delta$.*

Therefore, for each pair of pattern P and $G \in D_P$, the measure function $M(P, G)$ could first embed P in G, and then identify $Neighbor(P)$. If $|Neighbor(P)|$ is smaller than 10, returns 0. This pruning check is computationally cheap and only takes time linear in $|V(G - P)|$.

We summarize our study on the most useful constraints for graphs in Figure 1. Proofs are omitted due to space limit.

5 Mining Strategy Selection

The checking steps for pruning patterns and data are both associated with a computational cost. Alternatively, one can first mine all frequent patterns by a known mining algorithm, gSpan [17] for example, then check constraints on every frequent pattern output, and discard those that do not satisfy the constraint. We call this method the enumerate-and-check approach in the following discussion. Which approach is better depends on the total mining cost in each case. The best strategy therefore is to estimate the cost and potential benefit for each approach at every pruning step, and adopt the one that would give better expected efficiency.

The growth of a pattern forms a partial order $\prec$ defined by subgraph containment, *i.e.*, $P \prec Q$ if and only if $P \subseteq Q$. The partial order can be represented by a pattern tree model in which a node P is an ancestor of a node Q if and only if $P \prec Q$.

Each internal node represents a frequent pattern, which is associated with its own data search space $\{T_i\}$. The execution of a pattern mining algorithm can be viewed as growing such a pattern tree. Every initial pattern seed is the root of a new tree. Every time it augments a frequent pattern P, it generates all P's children in the tree. As such, each leaf corresponds to an infrequent pattern, or, in our mining model, a pattern that does not satisfy the constraints, since it is not further grown. Accordingly, the running time of a pattern mining algorithm can be bounded by the total number of nodes it generates in such a tree. This total sum is composed of two parts: (1) the set of all internal nodes, which corresponds to all the frequent patterns and is denoted as F; and (2) the set of all leaves, denoted by L, which corresponds to the infrequent patterns or constraint-violating patterns.

Let's look at the running time of the enumerate-and-check approach. Let the minimum support threshold be σ, *i.e.*a frequent pattern has to appear in at least $\sigma|D|$ graphs, where D is the entire graph database. Let $T_c(P)$ be the cost to check a constraint C on a graph P. The running time of the enumerate-and-check approach can be lower-bounded as follows:

1. Internal Nodes
   If an augmented pattern P is frequent, at least $\sigma|D|$ time has to be spent in the frequency checking and data search space construction. Hence, the construction time for such a node P is $|D_P| + T_c(P) \geq \sigma|D| + T_c(P)$.
2. Leaf Nodes
   If P is infrequent, at least $\sigma|D|$ time has to be spent in frequency checking. Since frequency checking is limited to support data space of P's parent node, denoted as $Parent(P)$, the construction time for P is $\geq min(\sigma|D|,|D_{Parent(P)}|) = \sigma|D|$.

Then the total cost, T_P, for mining from an initial pattern seed P by the enumerate-and-check approach is lower-bounded as $T_P \geq \sum_{P_i \in F_P}(\sigma|D| + T_c(P_i)) + \sum_{P_i \in L} \sigma|D|$ $= \sigma|D||F_P| + \sum_{P_i \in F_P} T_c(P_i) + \sigma|D||L| \geq 2\sigma|D||F_P| + \sum_{P_i \in F_P} T_c(P_i)$, where F_P is the set of frequent patterns grown from P. Essentially, the time cost of the enumerate-and-check approach is proportional to $|F_P|$. To bound $|F_P|$ means to bound the number of frequent patterns that would be generated from a pattern P. It is very hard to analytically give an accurate estimation of this heavily-data-dependent quantity. Our empirical studies show that in general, $|F_P|$ is very large for small patterns. An upper-bound of

$|F_P|$ can also be proved to be $\Theta(2^{|G|}), G \in D_P$. The proof is omitted due to space constraint.

Now we show how we should choose between the enumerate-and-check approach and our mining framework in both pattern pruning and data pruning cases.

1. **Pruning Patterns:** If we can prune a frequent pattern P after checking constraints on it, then the entire subtree rooted at P in the pattern tree model will not be grown. The time we would save is T_P. The extra time spent for constraint checking is $T_c(P)$. If it turns out that we can not prune it after the checking, we will grow it as in the enumerate-and-check approach. The expected cost of using our mining model is $T_{prune} = T_c(P)+(1-p)\cdot T_P$ where p is the probability that P fails the constraint checking. $T_c(P)$ depends on the algorithm used for the constraint checking, while p will be estimated empirically. The cost of the enumerate-and-check approach is T_P. As such, pattern pruning should be performed when $T_c(P) \leq p \cdot T_P$.

2. **Pruning Data:** If we can prune a graph G from the data search space of P after data pruning checking, G will be pruned from the data search spaces of all nodes in the subtree rooted at P. Therefore the time we save is lower-bounded by T_P. Let $T_d(P, G)$ be the time cost to check data pruning for a pattern P and a graph $G \in D_P$. Let q be the probability that G can be discarded after checking data pruning. Then for a graph $G \in D_P$, using data pruning by our model takes expected time $T_{prune} = T_d(P, G) + (1 - q)T_P$, while the enumerate-and-check approach would cost time T_P. The probability q can be obtained by applying sampling technique on D_P. We would perform data pruning checking for G if $T_d(P, G) < q \cdot T_P$. Otherwise, we shall just leave G in the search space.

6 Experimental Evaluation

In this section, we are going to demonstrate the pruning power provided by the the new antimonotonicities introduced in our framework, *i.e.*, weak pattern-antimonotonicity and data-antimonotonicity. Among all of structural constraints described in Figure 1, minimum density ratio and minimum degree are selected as representatives. All of our experiments are performed on a 3.2GHZ, 1GB-memory, Intel PC running Windows XP.

We explored a series of synthetic datasets and two real datasets. The synthetic data generator[1] is provided by Yan *et al.*[23], which includes a set of parameters that allow a user to test the performance under different conditions. There are a set of parameters for users to specify: the number of target graphs(N), the number of objects (O), the number of seed graphs (S), the average size of seed graphs (I), the average number of seed graphs in each target graph (T), the average density of seed graphs (D), and the average density of noise edges in target graphs.

The detailed description about this synthetic data generator is referred to [23]. For a dataset which has 60 relational graphs of 1,000 distinct objects, 20 seed graphs (each seed graph has 10 vertices and an average density 0.5), 10 seed graphs per relational graph, and 20 noise edges per object ($0.01 \times 1,000 \times 2$), we represent it as N60O1kS20T10I10 D0.5d0.01.

[1] It will produce a distinctive label for each node in a graph.

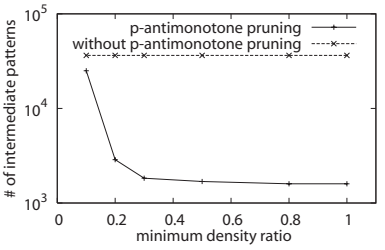


Fig. 2. Weak P-Antimonotonicity

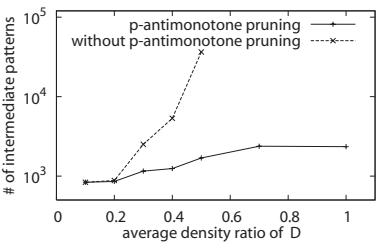


Fig. 3. Weak P-Antimonotonicity

The first experiment is about the minimum density ratio constraint. As proved in Section 4, minimum density ratio has weak pattern-antimonotonicity property. For each graph whose density ratio is greater than δ ($0 < \delta \leq 1.0$), we can find a subgraph with one vertex less whose density is greater than δ. That means we can stop growing any frequent pattern with one more vertex if its density is less than δ.

Figure 2 shows the pruning performance with various minimum density ratios. The data set used here is N60O1kS20T10I10 D0.5d0.01. The Y axis depicts the intermediate frequent patterns that are accessed during the mining process. The fewer the intermediate patterns, the better the performance, given the cost of checking the pattern's density ratio is negligible. The two curves show the performance comparison between methods with and without weak P-antimonotone pruning. As the figure shows, with the integration of the minimum density ratio constraint, we only need to examine much fewer frequent patterns, which proves the effectiveness of weak pattern-antimonotonicity. In the next experiment, we fix the density ratio threshold at 0.5 and change the average density ratio of seed graphs (D) in the above synthetic dataset. The denser the seed graphs, the more the dense subgraph patterns. It could take longer to find these patterns. Figure 3 depicts the performance comparison between methods with or without weak P-antimonotone pruning. We found that when D is greater than 0.6, the program without P-antimonotone pruning cannot finish in hours, while the one with P-antimonotone pruning can finish in 200 seconds.

Besides the weak P-antimonotone pruning, we also examined pattern inseparable D-antimonotonicity to pruning the data search space for the density ratio constraint. Given a frequent subgraph P and a graph G in the database ($P \subseteq G$), we need a measure to quickly check the maximum density ratio for each graph Q, where $P \subseteq Q \subseteq G$. For this purpose, we developed an algorithm for fast maximum density ratio checking. Let P' be the image of P in G. Our algorithm has three steps: (1) transform G to G' by merging all of the nodes in P'. (2) apply Goldberg's maximum density ratio subgraph finding algorithm to find a maximum density ratio subgraph in G' (time complexity $O(n^3 logn)$, where $n = |V(G')|$) [24]. (3) for graph G, calculate a maximum density ratio subgraph that contains P'; if this density ratio is below the density ratio threshold, we can safely drop G from the data search space of P(*i.e.*, G does not contain any subgraph Q that contains P and whose density ratio is greater than the threshold). For each discovered subgraph, we perform this checking to prune the data search space as much as possible. Although this checking is much faster than enumerating all subgraphs in G, we find it runs slower than a method without pruning, due to the high computational

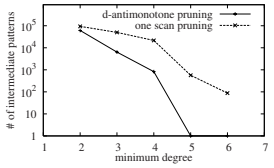


Fig. 4. P-Inseparable D-Antimonotonicity's effect on reducing # of intermediate patterns

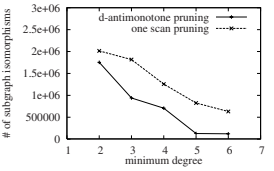


Fig. 5. P-Inseparable D-Antimonotonicity's effect on reducing # of subgraph isomorphism testings

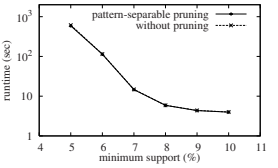


Fig. 6. P-Separable D-Antimonotonicity's effect on runtime

cost. That is, the cost model discussed in Section 5 does not favor this approach. Through this exercise, it was learned that, in order to deploy D-antimonotone pruning, the corresponding measure function in Defintion 5 has to be fast enough.

We applied the above frequent dense subgraph mining algorithm to the real data that consists of 32 microarray expression sets measuring yeast genome-wide expression profiles under different types of perturbations, e.g., cell cycle, amino acid starvation, heat shock, and osmotic pressure. Each dataset includes the expression values of 6661 yeast genes over multiple conditions. We model each dataset as a relational graph, where nodes represent genes, and we connect two genes with an edge if they have high correlation in their expression profiles. The patterns mined by our methods exhibit strong biological meanings. The mining result was published in our previous work [23].

Although we did not find a good D-antimonotonicity for the density ratio constraint, D-antimonotonicity is still applicable for other constraints, e.g., the minimum degree constraint. Neither pattern antimonotonicity nor weak pattern antimonotonicity is available for the minimum degree constraint. Thus, we develop a pruning technique using pattern-inseparable data antimonotonicity, which checks the minimum degree of a pattern embedded in each graph. If the degree is below threshold δ, we drop the corresponding graph from the data search space. The dropping will also decrease the frequency of each pattern and its superpatterns, which may make them infrequent as a result.

Figure 4 shows the comparison of pruning performance between data-antimonotonicity and a one-scan pruning method that drops vertices with less than δ edges before running PatternGrowth. When the minimum degree constraint is weak, *e.g.*, minimum degree threshold is low, these two methods have similar performance. However, when the constraint becomes strong, the pruning based on data-antimonotonicity performs much better.

Figure 5 shows the number of subgraph isomorphisms performed for these two algorithms. It is clear that, using data antimonotonicity, a lot of graphs are pruned in the early stage so that the number of subgraph isomorphisms done in the later stage can be significantly reduced. We now check one constraint with pattern separable D-antimonotonicity — the minimum size constraint. The minimum size constraint on frequent itemset mining and sequential pattern mining has been explored before, e.g., SLPMiner developed by Seno and Karypis [25]. Suppose our task is to find frequent graph patterns with

minimum length δ. One approach is to check the graphs in the data search space of each discovered pattern P and prune the graphs that are not going to generate patterns whose size is no less than δ. We developed several heuristics and applied our algorithm to mine the AIDS antiviral screen compound dataset from Developmental Theroapeutics Program in NCI/NIH [30]. The dataset contains 423 chemical compounds that are proved active to HIV virus.

Figure 6 shows the runtime of the two algorithms with and without pattern separable D-antimonotone pruning, with different support thresholds. The size constraint is set in a way such that less than 10 largest patterns are output. It is a surprise that for this dataset, pattern separable D-antimonotone pruning is not effective at all. Closer examination of this dataset reveals that most of the graphs can not be pruned because the sizes of frequent patterns are relatively small in comparison with the graphs in the database. This once again demonstrates, as also shown in the density ratio constraint, that the effectiveness of the integration of a constraint with the mining process is affected by many factors, e.g., the dataset and the pruning cost.

7 Conclusions

In this paper, we investigated the problem of incorporating sophisticated structural constraints in mining frequent graph patterns over a collection of graphs. We studied the nature of search space pruning for both patterns and data, and discovered novel antimonotonicities that can significantly boost pruning power for graph mining in each case: (1) weak pattern-antimonotonicity for patterns; (2) pattern-separable and pattern-inseparable data-antimonotonicities for data. We showed how these properties can be exploited to prune potentially enormous search space. An analysis of the trade-off between the enumerating-and-checking approach and the antimonotonicity-based approach was also given in this study.

References

1. Boulicaut,J.,De Raedt, L.: Inductive Databases and Constraint-Based Mining. (ECML'02) Tutorial.
2. Koyuturk, M., Grama, A., Szpankowski, W.: An efficient algorithm for detecting frequent subgraphs in biological networks. (ISMB'04). 200–207
3. Borgelt, C., Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules. (ICDM'02), 211–218
4. Deshpande, M., Kuramochi, M., Karypis, G.: Frequent sub-structure-based approaches for classifying chemical compounds. (ICDM'03). 35–42
5. Huan, J., Wang, W., Bandyopadhyay, D., Snoeyink, J., Prins, J., Tropsha, A.: Mining spatial motifs from protein structure graphs. (RECOMB '04), 308–315
6. Deshpande, M., Kuramochi, M., Wale, N., Karypis, G.: Frequent substructure-based approaches for classifying chemical compounds. IEEE TKDE **17**(8) (2005) 1036–1050
7. Yan, X., Yu, P.S., Han, J.: Graph indexing: A frequent structure-based approach.(SIGMOD'04), 335–346
8. Butte, A., Tamayo, P., Slonim, D., Golub, T., Kohane, I.: Discovering functional relationships between rna expression and chemotherapeutic susceptibility. In: Proc. of the National Academy of Science. Volume 97. (2000) 12182–12186

9. Ng, R., Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. (SIGMOD'98), 13–24
10. Bucila, C., Gehrke, J., Kifer, D., White, W.: DualMiner: A dual-pruning algorithm for itemsets with constraints. Data Mining and Knowledge Discovery **7** (2003) 241–272
11. Bonchi, F., Giannotti, F., Mazzanti, A., Pedreschi, D.: Exante: Anticipated data reduction in constrained pattern mining. (PKDD'03)
12. Bonchi, F., Giannotti, F., Mazzanti, A., Pedreschi, D.: Exante: A preprocessing method for frequent-pattern mining. In: IEEE Intelligent Systems 20(3). (2005) 25–31
13. Bonchi, F., Lucchese, C.: Pushing tougher constraints in frequent pattern mining. (PAKDD'04), 114 – 124
14. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. (PKDD'00), 13–23
15. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. (ICDM'01), 313–320
16. Vanetik, N., Gudes, E., Shimony, S.E.: Computing frequent graph patterns from semistructured data. (ICDM'02), 458–465
17. Yan, X., Han, J.: gSpan: Graph-based substructure pattern mining. (ICDM'02), 721–724
18. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraph in the presence of isomorphism. (ICDM'03), 549–552
19. Prins, J., Yang, J., Huan, J., Wang, W.: Spin: Mining maximal frequent subgraphs from graph databases. (KDD'04), 581–586
20. Nijssen, S., Kok, J.: A quickstart in frequent structure mining can make a difference. (KDD'04), 647–652
21. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. (VLDB'94), 487–499
22. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. (SIGMOD'00), 1–12
23. Yan, X., Zhou, X.J., Han, J.: Mining closed relational graphs with connectivity constraints. (KDD'05), 324–333
24. Goldberg, A.: Finding a maximum density subgraph. Berkeley Tech Report, CSD-84-171
25. Seno, M., Karypis, G.: Slpminer: An algorithm for finding frequent sequential patterns using length decreasing support constraint. (ICDM'02), 418–425
26. Dong, G., Han, J., Lam, J., Pei, J., Wang, K., Zou, W.: Mining constrained gradients in multi-dimensional databases. IEEE TKDE **16** (2004) 922–938
27. Gade, K., Wang, J., Karypis, G.: Efficient closed pattern mining in the presence of tough block constraints. (KDD'04), 138 – 147
28. Zaki, M.: Generating non-redundant association rules.(KDD'00), 34–43
29. Wang, C., Zhu, Y., Wu, T., Wang, W., Shi, B.: Constraint-based graph mining in large database. (In: APWeb 2005) 133 – 144
30. Yan, X., Han, J.: CloseGraph: Mining Closed Frequent Graph Patterns (KDD'03)286-295

Modeling Anticipatory Event Transitions

Ridzwan Aminuddin, Ridzwan Suri, Kuiyu Chang, Zaki Zainudin, Qi He, and Ee-Peng Lim

School of Computer Engineering
Nanyang Technological University
Singapore 639798
{muha0005,ridz0001,muha003,aseplim}@ntu.edu.sg,
{kuiyu.chang,qihe}@pmail.ntu.edu.sg

Abstract. Anticipatory Event Detection (AED) attempts to monitor user anticipated events, called Anticipatory Events (AE) that have yet to occur. Central to AED is the Event Transition Graph (ETG), which defines the pre and post states of a user specified AE. A classification model can be trained on documents in the pre and post states to learn to detect an AE. However, this simplistic classification model does not make use of discriminatory keywords between the two states. We propose a simple but effective feature selection method to identify important bursty features that highly discriminate between the pre and post states of an AE. Bursty features are first computed using Kleinberg's Algorithm, then various combination of features in both states are selected. Experimental results show that bursty features can significantly improve the accuracy of AED.

1 Introduction

The value of providing accurate and timely information, especially news, has exponentially increased with the prevalence of the Internet. For example, push-based news alert systems like Google News Alerts notify subscribers when user pre-specified events (called Anticipatory Event or AE) take place.

As alerts are pushed into handheld devices 24/7, it is extremely important[1] for the alerts to be delivered accurately and precisely. Unfortunately, current alert systems are not smart enough to figure out if a news document containing the user defined words satisfy the AE. In fact, to ensure uncompromisable accuracy, some portals like Yahoo rely on a human operator to approve system triggered news alerts, whereas others like Google use a completely automated approach, resulting in many false alarms[2].

AED systems based on classifying sentences/documents into pre/post AE states have been previously proposed[2,3]. The idea is to train a classifier using the pre/post documents of historical events with similar characteristics to the AE. For example, to create an AED system to detect an AE such as "US invades Iran", it can be trained using available documents from the pre/post states of the historical events "US invades Afghanistan" and "US invades Iraq". The crux

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 401–408, 2007.

of the AED system is thus the 2-state Event Transition Graph (ETG), where documents are assigned to either the pre or post states of the AE.

However, the trained model is a black box that does not open up for interpretation. Further, the element of time is not considered if we use standard Information Retrieval (IR) techniques to represent the documents from each of the two states. We therefore propose a novel approach to select representative features from the pre and post states based on the burstiness of words before and after the AE transition in the ETG, respectively.

2 Related Work

He et al.[3,2] originally proposed the AED concept as a new area under Topic Detection and Tracking (TDT). They also defined the general AED framework which includes an Event Transition Graph (ETG). Our work focus on deriving better features to represent the two states of an ETG, thereby improving the model.

Tax et al.[4] explained the application of support vector classifiers in two-class classification problems. The paper discussed the strengths as well as some of the critical limitations of the support vector classifier. Amongst these were the problems of small sample size as well as peaking effect. It proved that using as many features as possible does not necessarily improve the overall classification accuracy. Instead, there exists an optimal number of features beyond which performance of the classifier degrades. Our work, hopes to approximate this approach by limiting the features selected for classification in hope of improving the classification accuracy.

Kleinberg[5] formulated a finite state automaton to identify and define bursty words in a document stream. Bursty words are useful for extracting the structure of the document stream and thus attach a formal meaning to the stream. This is possible because bursty words add an additional time dimension to the set of documents. Such a model can be applied in many ways, one of which is to identify scientific research topics and trends as illustrated in [6]. Noting its usefulness, we applied this algorithm in our feature selection approach.

3 Model

3.1 The Anticipatory Event Detection Model

An overview of the AED framework is given in Figure 1. The ETG involves the formation of the individual events and the AE transition from one event to the next. Our primary goal is thus to accurately model each event in the ETG. An AE could be any event with a predecessor in the ETG. Once the ETG is well defined, say with a list of keywords, it could then be used to classify and trigger an AE.

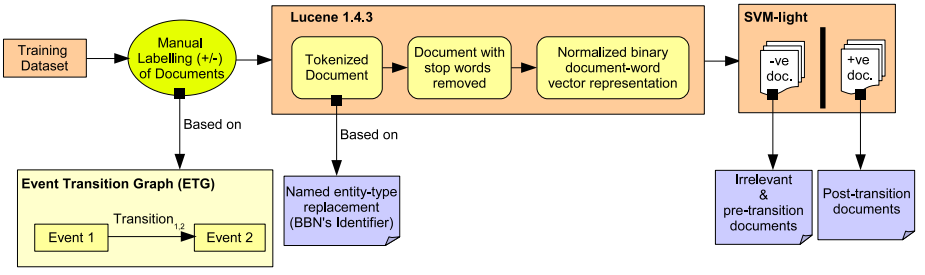


Fig. 1. Overview of the AED Model

3.2 Kleinberg's Burst Model

Kleinberg's burst model [5] finds bursty words from a document stream. A word is considered bursty if its document frequency (DF) exceeds a pre-defined threshold over a given time period. The idea thus is to extract a set of bursty words related to the two states of an AE. The bursty words would provide additional disciminatory information between the two states. We adopt the batch processing formulation of Kleinberg's 2-state automaton model as defined by Ketan Mane [6]. Figure 2 shows an overview of the model with two states q_0 and q_1, where the transition cost from the lower to the higher state is defined along with the cost for remaining in each state. Note that there is no cost for going from state q_1 back to q_0.

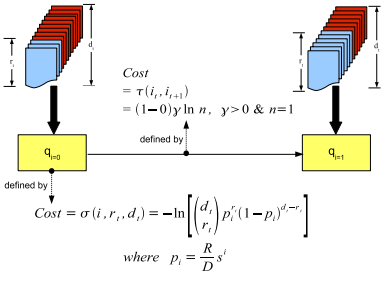


Fig. 2. Overview of Kleinberg's State Automaton

The objective of Kleinberg's model is to find for each word a state sequence with minimum cost. A word is considered bursty if its state sequence includes at least one bursty state q_1. Each burst has an associated weight, which is simply the reduction in cost between states q_1 and q_0 over its bursty interval. Thus, if the aggregated bursty weight value of a word is large, it implies that there are many periods in time when the word DF is extremely high.

3.3 Our Feature Selection Approach

We applied Kleinberg's algorithm to word DF of the set of pre and post transition documents separately as shown in Figure 3. In particular,

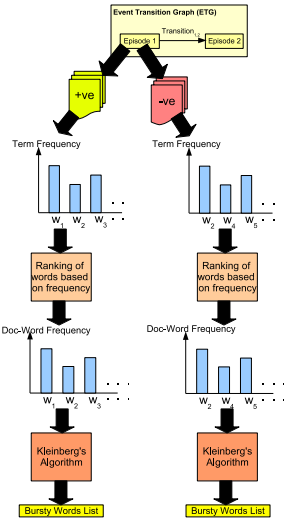


Fig. 3. Overview of Our Feature Selection Method

1. Select a set of documents corresponding to an AE.
2. Manually label each document as negative (pre) or positive (post).
3. For every word in a given document set, its DF is plotted and ranked.
4. The set of top DF words from each set are then fed into Kleinberg's algorithm where each word's state sequence and burstiness weight is computed. This state sequence characterizes the burstiness period of the word in time whereas the weight quantifies its overall burstiness.
5. If a word's ouput state sequence has at least one transition to state q_1, it will be considered bursty.
6. Top bursty words from each of the pre and post document streams are then collected to form two bursty sets: Positive (P) and Negative (N), respectively.

We tried different ways of selecting the bursty words for classification:

1. **Union** ($P \cup N$): Union of all bursty words found in both sets.
2. **Union+** ($P \cup N + P \cap N$): Same as Union, except that weights of common bursty words will be boosted by a factor of two.
3. **Discriminatory** ($P \cup N - P \cap N$): Union of unique words from both sets.
4. **Baseline**: Set of all (bursty and non-bursty) words.

Our approach aims to combine the best of static and temporal information from the document stream. The word document frequency ranking returns a list of highest document frequency terms within documents in an AE state. On the other hand, Kleinberg's model aims to select the words that encounter a sudden and extended burst in document frequency. The former selects the most frequent words and the latter extract words among which that becomes more relevant in some time periods.

3.4 SVM Classifier

Representing the documents from the pre/post event states using each of the 4 feature spaces, we train 4 SVM classifiers[7]. The SVM attempts to create a hyperplane that separates the pre and post states of the AE.

4 Experiments

4.1 Dataset

Six topics from the TDT3[8] corpus of news articles were selected and manually filtered to find a suitable event for modelling a state transition, i.e., from a 'negative' event state to a 'positive' event state. In simple terms, this means that we manually split each topic dataset into a two-state event-transition model.

Listed in Figure 4 are the six manually selected datasets. Negative and positive events were defined for each dataset, and the corresponding documents were identified and appropriately tagged. These datasets consist of only relevant documents with no off-topic items. Figure 5 shows the well balanced characteristics of each dataset.

Document vectors were created for all the documents within each of the test datasets. This was achieved by running the dataset documents through the Lucene [9] software that enabled indexing, removal of common stopwords, and representation of the dataset items into a document vector format. These were then utilized to obtain the top 500 high frequency terms and high document-word frequency terms for each dataset.

These high frequency words were then fed into Kleinberg's two state model, which will then output the burstiest words amongst the set of 500 high frequency words. It should be noted that Kleinberg's model does not take into account the term frequency of the word but instead considers the word document frequency to calculate the transition costs from state to state.

The SVM classifier is then used to train and test the datasets using the document vectors created using each of the 4 feature space representation.

4.2 Overlap of Bursty Features

In our experiments, bursty words are independently selected over the collections of negative and positive AE documents. The intention is simply to identify possible terms within the event states that would best represent each AE.

Dataset	'negative' pre-state	'positive' post-state
Dataset 1 : Hurricane Mitch	Hurricane Mitch has hit Central and South America wreaking devastation, causing death and disease.	Hurricane Mitch finally hits North America, Florida, after threatening to do so for some time.
Dataset 2 : NBA Strike	Players from the NBA go on strike after non-agreement and dispute with its owners. Plans are then made for talks.	Closed-door discussion sessions finally commence after a 2 week break following Thanksgiving.
Dataset 3 : New York Yankees win The World Series	New York Yankees are 3-0 games up and need one more game to win the championship	New York Yankees win the 1998 World Series after claiming game 4 at San Diego
Dataset 4 : Bill Clinton peace talks in Israel	US President Bill Clinton facing impeachment and is under tremendous pressure back in the states. At the same time, peace talks in the middle east is breaking down	US President Bill Clinton makes the trip to Israel to mediate peace talks.
Dataset 5 : US Budget	The US federal government rush to prepare a federal budget for fiscal year 1999.	An announcement was made Thursday afternoon that the White House and congressional negotiators have reached a budget deal.
Dataset 6 : US House Representatives Election	Newt Gingrich steps down as Republican House Representative which opens up the nominations for a new representative.	A new House Representative is elected and immediately assumes the role.

Fig. 4. Description of the 6 TDT3 topics

	# Training Documents	*# Features*	*# Bursty Features*			
		All words	**All bursty**	*negative*	*overlap*	*positive*
1 hurricane	*287*	*7195*	**100**	*37*	*16*	*47*
2 nba	*172*	*7360*	**180**	*51*	*35*	*94*
3 worldseries	*156*	*7719*	**143**	*30*	*40*	*73*
4 clinton	*202*	*7064*	**209**	*70*	*50*	*89*
5 budget	*167*	*5904*	**147**	*49*	*30*	*68*
6 election	*148*	*7703*	**263**	*71*	*53*	*139*

Fig. 5. Details of the 6 TDT3 topics

Figure 6 illustrates the overlap of the top 15 bursty words between the two sets. We see that the words selected are quite representative of the fired AE, e.g. 'deal' and 'money' were some of the discriminating words for the passed budget, while 'relief', 'aid', 'million', 'food' confirms the arrival of Hurricane Mitch.

4.3 AED Results

For each topic, five-fold cross validation was done to obtain the average SVM test results, which include the overall accuracy, and for each class of data the precision, recall, and F-measure. Results are summarized in Figure 7. Clearly, the 3 bursty feature selections consistently meet or better the full feature space,

1 hurricane		
Negative	**Shared**	**Positive**
central winds miles rain	*mitch hurricane people honduras nicaragua*	*america million aid relief countries food*

2 nba		
Negative	**Shared**	**Positive**
labour first more games union	*nba players season basketball national league lockout association*	*time david*

3 worldseries		
Negative	**Shared**	**Positive**
game one two diego san padres three	*world yankees new*	*series team york baseball season*

4 clinton		
Negative	**Shared**	**Positive**
Israel palestinian bank west agreement minister wye netenyahu	*president clinton week*	*us friday days troops*

5 budget		
Negative	**Shared**	**Positive**
white leaders democrat	*congress budget president house republicans clinton spending bill*	*money deal social*

6 election		
Negative	**Shared**	**Positive**
election new majority people democrats	*republican livingston republicans newt gingrich speaker house*	*congress impeachment president*

Fig. 6. Top 15 Bursty Features for Each Dataset

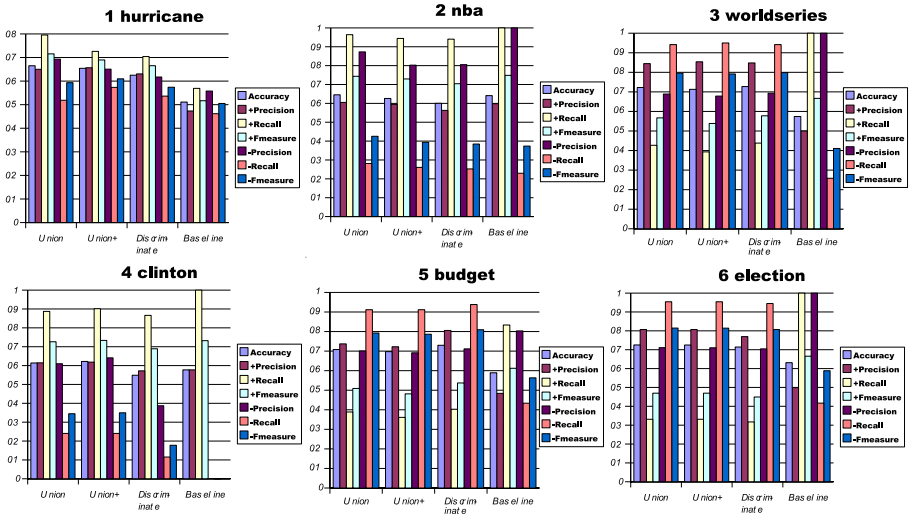


Fig. 7. Experimental Results

with the Union+ strategy leading the pack in every topic. Results of the Discriminatory selection seems more varied, probably due to SVM needing the removed common words to determine an optimal hyperplane.

Note that some of the 100% baseline precision and recall values are due to SVM classifying the majority or all of the data as positive, which skews the F-Measure for the positive target class. This can be seen by the corresponding lower negative class metric values. Moreover, accuracy should be considered since the classes are well-balanced.

5 Conclusion and Future Work

Our feature selection approach effectively enables the accurate classification of documents to their appropriate event states for AED. Equal or better classification accuracies were obtained by using less than 4% bursty features from the whole corpus, a great savings in time/complexity. This opens up further research in AED. Specifically, the formation of a more complex ETG including the transitions from one event to the next will be our focus research area in the near future. Lastly, it is noted that our simple yet effective feature selection approach can be applied to a myriad of applications involving text streams, such as chat messages.

References

1. Chua, K., Ong, W.S., He, Q., Chang, K., Kek, A.: Intelligent portal for event-triggered sms alerts. In: In Proceedings of the 2005 IEE Mobility Conference. (2005)
2. He, Q., Chang, K., Lim, E.P.: A model for anticipatory event detection. In: 25th International Conference on Conceptual Modeling (ER 2006), LNCS 4215. (2006) 168–181
3. He, Q., Chang, K., Lim, E.P.: Anticipatory event detection via sentence classification. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC 2006). (2006)
4. Tax, D., de Ridder, D., Duin, R.P.: Support vector classifiers: a first look. In: In Proceedings of the 3rd Annual Conference of the Advanced School of Computing and Imaging, ASCI, Delft. (1997)
5. Kleinberg, J.: Bursty and hierarchical structure in streams. In: ACM SIGKDD. (2002)
6. Mane, K., Borner, K.: Mapping topics and topic bursts in pnas. In: In Proceedings of the National Academy Of Science (PNAS) in the years 1982 - 2001. (2004)
7. Joachims, T.: (Svm-light, http://svmlight.joachims.org)
8. Detection, T., 3, T.D.: (http://projects.ldc.upenn.edu/tdt3)
9. Lucene: (Apache lucene 1.4.3, http://lucene.apache.org)

A Modified Relationship Based Clustering Framework for Density Based Clustering and Outlier Filtering on High Dimensional Datasets

Turgay Tugay Bilgin[1] and A. Yilmaz Camurcu[2]

[1] Department of Computer Engineering , Maltepe University
Maltepe, Istanbul, Turkey
ttbilgin@maltepe.edu.tr
[2] Department of Electronics and Computer Education, Marmara University
Kadikoy, Istanbul, Turkey
camurcu@marmara.edu.tr

Abstract. In this study, we propose a modified version of relationship based clustering framework dealing with density based clustering and outlier detection in high dimensional datasets. Originally, relationship based clustering framework is based on METIS. Therefore, it has some drawbacks such as no outlier detection and difficulty of determining the number of clusters. We propose two improvements over the framework. First, we introduce a new space which consists of tiny partitions created by METIS, hence we call it micro-partition space. Second, we used DBSCAN for clustering micro-partition space. The visualization of the results are carried out by CLUSION. Our experiments have shown that, our proposed framework produces promising results on high dimensional datasets.

1 Introduction

One of the important problems of Data mining (DM) community is mining high dimensional datasets. As dimensionality increases, the performance of the clustering algorithms sharply decreases.

In this paper we introduce a new high dimensional density based clustering and visualization framework based on Strehl & Ghosh's relationship based clustering framework [1]. Their framework has two fundamental parts named CLUSION (CLUSter visualizatION toolkit) and OPOSSUM (Optimal Partitioning of Sparse Similarities Using METIS). CLUSION is a similarity matrix based visualization technique and OPOSSUM is a balanced partitioning system which uses a graph based partitioning tool called METIS [2]. OPOSSUM produces either sample or value balanced clusters. Since METIS is a partitioning system, there is no outlier detection or filtering on OPOSSUM/CLUSION framework. We have modified Strehl & Ghosh's framework to deal with density based clustering and outlier filtering. Our framework uses DBSCAN to filter outliers and an intermediate space called micro-partition space as an input space for the DBSCAN. Micro-partition space is created by using METIS.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 409–416, 2007.

2 Related Work

Graph based partitioning methods perform best on very high dimensional datasets. Currently, the most popular programs for graph partitioning are CHACO and METIS [2,3].

Density-based clustering algorithms group neighboring data objects into clusters based density. DBSCAN (Density-Based Spatial Clustering of Applications with Noise)[4] is a typical representative of this group of algorithms.

There are many visualization techniques for high dimensional datasets. Keim and Kriegel [5] grouped visual data exploration techniques for multivariate, multidimensional data into six classes. These techniques become useless on the datasets that have dimensions above some hundreds.

To overcome the drawbacks of dimensionality, matrix based visualization techniques [6] can be used on very high dimensional datasets. In matrix based visualization techniques, similarity in each cell is represented using a shade to indicate the similarity value: greater similarity is represented by dark shading, while lesser similarity by light shading. CLUSION, a matrix based visualization technique, is used in both Strehl and Ghosh's and our framework as explained in the following sections.

3 Relationship-Based Clustering Approach

Strehl A. and Ghosh J. proposed a different approach in [1] very high dimensional data mining. In their framework the focus was on the similarity space rather than the feature space. Most standart algorithms spend little attention on the similarity space. The key difference between relationship-based clustering and regular clustering is the focus on the similarity space $\mathbf{S}$ instead of working directly in the feature domain $\mathbf{F}$. Once similarity space is computed, a modified clustering algorithm which can operate on the similarity space, is used to partition the similarity space. The resulting space is reordered, so that the points within the same cluster put on adjacent positions. The final step is the visualization of the similarity matrix and visually inspecting the clusters.

3.1 Relationship-Based Clustering Framework

A brief overview of the general relationship-based framework is shown in figure 1. χ is a collection of n data source. Extracting features from pure data source yields $\mathbf{X}$ feature space. In most cases, some data preprocessing is applied to the data source to obtain feature space. Similarities are computed, using e.g. euclidean, cosine, jaccard based similarity Ψ yielding the $n \times n$ similarity matrix $\mathbf{S}$. Once the similarity matrix is computed, further clustering algorithms run on similarity space. Clustering algorithm Φ yields cluster labels λ.

3.2 OPOSSUM/CLUSION System and Problems

Relationship-based clustering framework employs METIS for clustering. It can operate on similarity space. Strehl and Ghosh call METIS based balanced

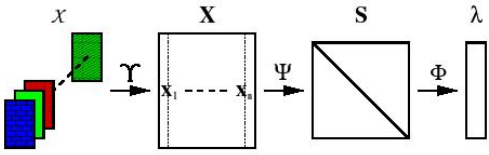


Fig. 1. Relationship-based clustering framework [1]

partitioning visualization system as OPOSSUM [1]. OPOSSUM differs from other graph-based clustering techniques by sample or value balanced clusters and visualization driven heuristics find an appropriate k.

CLUSION is a matrix based visualization tool which is used for visualizing the results. It looks at the output of the clustering routine (λ index), reorders the data points so that points with the same cluster label are contiguous, and then visualizes the resulting permuted similarity matrix, $\mathbf{S}'$. Mathematical background of the system can be found on [1].

The computing performance and the quality of the clusters produced by OPOSSUM/CLUSION framework is quiet impressive. However it is not perfect. It has two major drawbacks:

a) *Determining the number of clusters :* Since partitioning based METIS is used for clustering the similarity space, finding the 'right' number of clusters k for a dataset is a difficult and often ill-posed problem, even for the same dataset, there can be several answers depending on the scale or granularity one is interested in.
b) *No outlier filtering :* Partitioning based clustering algorithms generally suffer from outliers. As it is denoted in previous sections, OPOSSUM system produces either value or sample balanced clusters. On this kind of systems, outliers can reduce the quality and the validity of the clusters depending on the resolution and distribution of the dataset. Outliers are filtered before clustering process in some applications.

4 Our Framework for Density Based Partitioning and Outlier Filtering

The architecture of our framework is shown in Figure 2. It consists of the following three major improvements:

a) An intermediate space is introduced. We call it 'micro-partition space' and it is denoted by **M** in figure 2. We use METIS for ordering the similarity space and creating micro-partitions. This process is represented by μ in figure 2. METIS ensures that, neighboring points are put in adjacent positions in similarity matrix. See section 4.1 for details of creating micro-partition space.
b) Using DBSCAN for density based clustering of micro-partition space. This part is represented by Φ in figure 3. Since the dimensionality of micro-partition space is less than the similarity space, DBSCAN shows better performance on micro-partition space rather than the original similarity space.

c) Easier determination of input parameters. METIS is dependent on k, on the other hand, DBSCAN depends on ϵ-neighborhood radius (Eps) and minimum number points within the ϵ-neighborhood (MinPts). Determining Eps and MinPts are easier than the determination of k. There is a simple but effective heuristics to determine DBSCAN parameters in [4].

DBSCAN produces λ index from micro-partition space. We use CLUSION for visualizing the micro-partition space which is reordered by λ index.

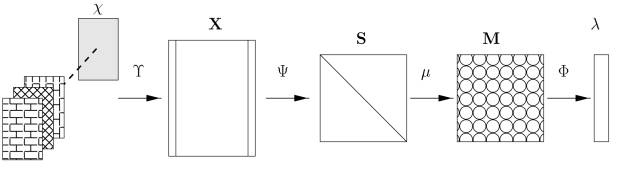


Fig. 2. Our framework for density based partitioning and outlier filtering

4.1 Micropartition Space

When DBSCAN is directly applied to the similarity space, it can not properly order it. In other words, although it can deal with the outliers, DBSCAN can not put the neighboring points in adjacent positions on the similarity graph. This causes poorly distinguishable CLUSION graphs. For human eye it is not easy to find out the clusters from the graphs. Hence, we used METIS for properly ordering the similarity space. METIS needs the number of partitions k as an input parameter. Since we do not know how many partitions exists, we have used METIS with a predetermined value of k. Micro-partition space contains the partitions produced by METIS. DBSCAN algorithm treats micro-partitions as input data, in other words DBSCAN works on micro-partition space.

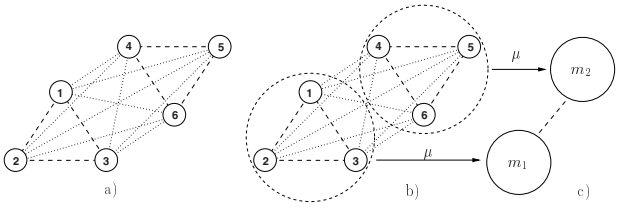


Fig. 3. a) Points on similarity space before METIS, b) METIS creates micro-partitions c) μ function creates **M** space

Let p be the number of samples within a micro-partition, then the relation between k and p should be:

$$n >> p > 1 \tag{1}$$

$$k = \frac{n}{p} \tag{2}$$

where n is the number of samples. In figure 3, schematic representation of creating micro-partition space for $p = 3$ is shown. Let x_j be a sample in dataset with $j \in \{1, ..., n\}$ and six samples be placed as shown in figure 3.a. If similarities $s(1,2), s(1,3), s(2,3)$ for x_1, x_2, x_3 are sufficiently small, METIS will put them into the same partition. x_4, x_5, x_6 will be another partition accordingly (figure 3.b). From now on, micro-partitions in **S** are treated as samples of **M** in such way that

$$\mu(x_1, x_2, x_3) \Rightarrow m_1 \tag{3}$$

$$\mu(x_4, x_5, x_6) \Rightarrow m_2 \tag{4}$$

$$\mu(x_{n-2}, x_{n-1}, x_n) \Rightarrow m_k \tag{5}$$

μ function takes METIS partitions as input, produces a single representative point no matter how many samples exist in micro-partition. μ function chooses the representative point within micro-partition randomly. If similarities $s(1,2)$, $s(1,3)$, $s(2,3)$ are sufficiently small, any of these points can be chosen as repsentative with a very small amount of error.

As can be seen from figure 4, the error is

$$E(m_i, m_j) = |s(x_{act_{m_i}}, x_{act_{m_j}}) - s(m_i, m_j)| \tag{6}$$

$$s(m_i, m_j) = s(x_{rep_{m_i}}, x_{rep_{m_j}}) \tag{7}$$

Where $i, j \in \{1, ..., k\}$ and x_{act} are actual point, x_{rep} is the representative point. As the micro-partition size increased, the error will also increase. This is why we choose p as small as possible. We experienced that $p = 3$ or $p = 4$ values generally yield good results.

4.2 Density Based Clustering of Micropartition Space Using DBSCAN and Outlier Filtering on CLUSION Graphs

DBSCAN operates on micro-partition space which is $k \times k$ dimensional according to equation 2. Due to the decrease in dimensionality, DBSCAN performs better on **M** space than original on **S** space. We use DBSCAN for outlier filtering and density based clustering of M space. Since **M** space is ordered by METIS, this reduces the neighborhood search time in DBSCAN.

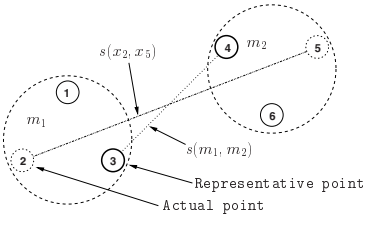


Fig. 4. Demonstration of error in micro-partition space

λ indexes are used to reorder **S** and **M** spaces. λ_M denotes index produced by METIS where $\lambda_M \in \{1, ..., k\}$, and λ_D denote index produced by DBSCAN where $\lambda_D \in \{-1, 1, ..., k\}$. METIS is a partitioning system, therefore there is no outlier remarking capability of λ_M index. However, DBSCAN is a density based clustering algorithm and it has the capability of remarking outliers. λ_D index remarks outliers with the value of -1. When samples from **M** space are reordered according to the λ_D index, outliers will appear in front of the array. As a result, outliers will be placed on the upper left corner of the CLUSION graph (cross patterned areas in figure 5.b and 5.c).

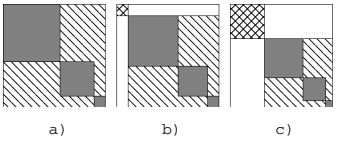


Fig. 5. a) Reordering by λ_M index (no outliers) b) Reordering by λ_D index (less outliers) c) Reordering by λ_D index (more outliers)

Figure 5.a shows the schematic results of reordering by λ_M index. There are a few outliers in figure 5.b which show results of reordering by λ_D index. Figure 5.c has more outliers. Note that, gray areas on the graph are placed for illustrative purposes, therefore the real life graphs may differ from the ones in figure 5.

5 Experiments

We evaluated our proposed framework on two different real world datasets. The first dataset consists of 9636 terms from 2225 complete news articles from the BBC News web site, corresponding to stories in five topical areas (business, entertainment, politics, sport, tech) from 2004-2005 [8]. The second one (COIL 2000) consists of 86 attributes (dimensions) of the 5822 customers and includes product usage data and socio-demographic data derived from zip area codes [7].

In the first experiment we initially used OPOSSUM to cluster BBC news articles dataset. Figure 6 shows the result of OPOSSUM. The optimal k for sample based clustering is found at $k = 3$ after numerous trial and error. There is no way to find out and filter outliers on the graph. Figure 7 shows CLUSION graph of our framework for p=5. Increasing p dramatically reduces the computing time. $p = 3, p = 4$ and $p = 5$ values are optimal for most cases. Consequently, complex determination routines for p values are not necessary.

DBSCAN is used to cluster M space with $\varepsilon = 0.066$ and MinPts=5 parameters. MinPts value is chosen to be equal to p and then the simple heuristics mentioned in [4] is used to determine ε approximately. The heuristics yield $\varepsilon = 0.066$. As it can be seen from the figure 7, the graph is windowed by one horizontal and one vertical line. The outliers are placed on upper left window and the lower right window shows filtered clusters. It is clearly seen that, the filtered area contains almost equal three clusters. One should perform numerous trials to find

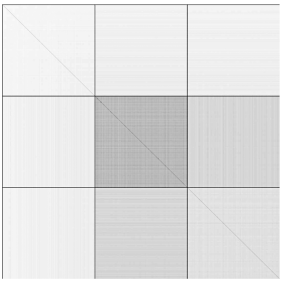


Fig. 6. CLUSION graph of OPOSSUM on BBC news articles dataset for $k = 3$

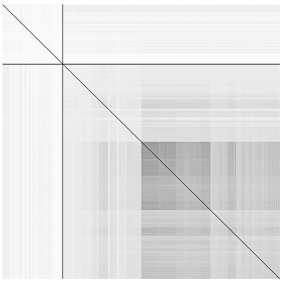


Fig. 7. CLUSION graph of our framework on BBC dataset with p=5, $\varepsilon = 0.066$, MinPts=5

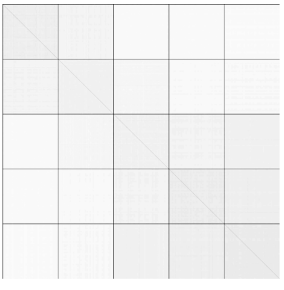


Fig. 8. CLUSION graph of OPOSSUM for $k = 5$ on COIL 2000 dataset

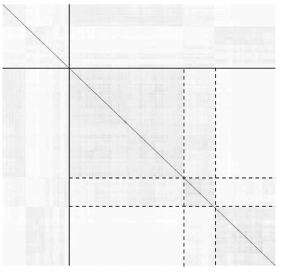


Fig. 9. CLUSION graph of our framework on COIL 2000 dataset with p=5, $\varepsilon = 0.1$, MinPts=5 (dotted lines show cluster borders)

out correct number of clusters using OPOSSUM, indeed the resulting clusters will contain outliers. On the other hand, our framework filtered the outliers and found three clusters just running it only once or a few times.

The second experiment carried out on COIL 2000 dataset. The results of OPOSSUM for k=5 is shown in figure 8. Due to the fuzziness on the CLUSION graphs, none of them produced well distinguishable cluster structure. Therefore we could not discover the exact value of k. We applied our framework for $p = 5$. If we look at the lower right window of the figure 9, we can see two distinct clusters. One bigger cluster is placed on the upper left and one smaller cluster placed on the lower right across the main diagonal. When we carefully examine the intersection of two clusters, we can see a small cluster with very close relationship with the both clusters. To see clearly, the borders of the clusters are marked as dotted lines in figure 9. Therefore we can say that, our framework has succesfully found three clusters on COIL 2000 dataset.

6 Conclusion

This paper proposes a new framework for density based clustering of high dimensional datasets and getting better interpretations for clustering results by filtering the outliers from the main perspective of the CLUSION graph. This is achieved by using a new space called micro-partition space and modifying DBSCAN algorithm to operate on micro-partition space. Our proposed framework is based on the relationship based clustering approach of Strehl and Ghosh. Our experiment shows that our improvements and modifications help us better clustering and visualization of the high dimensional data. In future work, we will replace METIS with another faster partitioning tool for performance improvements and better integration with DBSCAN.

Acknowledgement. The research for this article was supported by grant number FEN DKR-270306-0055 from Marmara University Scientific Research Projects Committee (BAPKO).

References

1. Strehl, A., Ghosh, J.: Relationship-based clustering and visualization for high dimensional data mining. INFORMS Journal on Computing,(2003), 208-230.
2. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal of Scientic Computing, 20(1), (1998), 359-392.
3. Karypis, G., Kumar, V.: A parallel algorithm for multilevel graph-partitioning and sparse matrix ordering. Journal of Parallel and Distributed Computing, 48(1), (1998), 71-95.
4. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. 2nd International Conference on KDD, (1996), 226-231.
5. Keim, D.A., Kriegel, H.P.: Visualization Techniques for Mining Large Databases: A Comparison, IEEE Trans. Knowledge and Data Eng., vol. 8, no. 6, Dec. (1996), 923-936.
6. Gale, N., Halperin, W., Costanzo, C.: Unclassed matrix shading and optimal ordering in hierarchical cluster analysis. Journal of Classication, 1:75-92, (1984).
7. The Insurance Company Benchmark (COIL 2000), The UCI KDD Archive, (http://www.ics.uci.edu/ kdd/databases/tic/tic.html, February 2006).
8. BBC news articles dataset from Trinity College Computer Science Department (https://www.cs.tcd.ie/Derek.Greene/research/, February 2006).

A Region-Based Skin Color Detection Algorithm

Faliang Chang, Zhiqiang Ma, and Wei Tian

School of Control Science and Engineering, Shandong University, P. R. China
flchang@sdu.edu.cn, zqma@mail.sdu.edu.cn

Abstract. In this paper, a new region-based algorithm for detecting skin color in static images is described. We choose the single Gaussian skin color model in the normalized r-g space after analyzing the distributions of skin color in six different 2-D chrominance spaces. Images are first segmented into patches using a improved fuzzy C-means algorithm, in which the local characteristic is adopted to constrain fuzzy functions, and a simple method for initializing clustering centriods is adopted. Then, the percentage of skin color pixels in each patch can be obtained. According to corresponding percentages, patches are classified as skin color regions or not.

Keywords: skin color detection, fuzzy C-means clustering, color image segmentation.

1 Introduction

Skin color detection has played an important role in many applications such as face recognition, gesture recognition, human-computer interaction, tracking faces, and filtering pornographic images in web pages [1,2,3]. However, it is not easy to detect skin color accurately, because images are taken with different camera hardwares and under confusing illumination. Moreover, there are many other objects whose color are similar to skin and which are easily confused with skin. Finally, different human races present different skin tones.

Plenty of detecting strategies and skin color models have been proposed to detect skin color in images. These approaches can be classified into two categories: pixel-based methods and region-based methods [4]. Pixel-based methods have long history. Kovac et al. defined explicitly skin color cluster boundaries through a series of decision rules in the RGB color space [2]. Jones et al. constructed skin and non-skin color histogram models and derived a skin pixel classifier through the standard likelihood ratio approach [5]. As for parametric skin color distribution models, a single Gaussian or a mixture of Gaussians probability density function was used to model the distribution of skin color in a 2-D chrominance space [3]. In contrast to pixel-based methods, region-based detection methods use spatial arrangement information of skin color pixels to improve detection rates. Considering that skin patches especially human faces are nearly elliptic, Kruppa et al. [1] and Yang et al. [6] refined detection algorithms with this shape information.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 417–424, 2007.

In this study, we propose a new region-based approach to detect skin color without assuming that skin color pixels merge in ellipses. In section 2, we first choose the chrominance space in which the single Gaussian model fits the distribution of training skin color samples best. Then, a few decision rules are defined to delete some background color, and nearly 95% of skin pixels are preserved. Fuzzy C-means (FCM) algorithm is employed to segment remainder pixels in Section 3. In particular, the local characteristic [7] is adopted to constrain fuzzy functions in order to decrease influences of illumination. Besides that, the algorithm initializes the maximum number of classes and cluster centroids adaptively in terms of histogram properties of images. In Section 4, how to classify skin color patches and experiments are introduced. Section 5 summarizes our study.

2 Skin Color Distribution and Raw Detection Rule

2.1 Skin Color Distribution in Different Color Spaces

A color space efficiently separating chrominance from luminance in the original color image helps to improve a robustness to changes in illumination conditions. This can be achieved by reducing one dimension from a color space through a transformation from the 3-D RGB color space into a 2-D chrominance space [3]. Different algorithms use different spaces. The algorithm presented in [2] was realized in the Cb-Cr chrominance space. Perceptually uniform spaces CIE-Luv and CIE-Lab were also adopted in [6,4].

We collect 200 real-world images randomly from Internet and sample nearly 1.2 million skin color pixels as the training database. The races include Caucasian and Asian. The cumulative distributions of all training skin pixels are mapped with 256×256 bins in six different chrominance spaces: normalized r-g, T-S, H-S, CIE-ab, I-Q, Cb-Cr (Fig. 1). Comparing the six histograms, we draw a conclusion that the distribution of skin color in the normalized r-g space is a bit more compatible than that in other spaces and fits the single Gaussian model best. The single Gaussian model is given by the following expression

$$f(\boldsymbol{x}|skin) = \frac{1}{2\pi|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] . \tag{1}$$

Parameters $\boldsymbol{\mu}$ and Σ are mean color vector and covariance matrix respectively, which can be trained from training data using Maximum Likelihood Estimation. $f(\boldsymbol{x}|skin)$ is considered as the measure of the likelihood of color $\boldsymbol{x}$ to skin color. So a pixel will be labeled as a skin color pixel if its $f(\boldsymbol{x}|skin)$ is greater than a threshold. As the first part of Eq. 1 is a constant once Σ is given, Eq. 1 reduces to

$$\theta(\boldsymbol{x}) = \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] . \tag{2}$$

Then, the classification rule is replaced by

$$\theta(\boldsymbol{x}) \geq \theta_s, \quad \theta_s \in [0,1] . \tag{3}$$

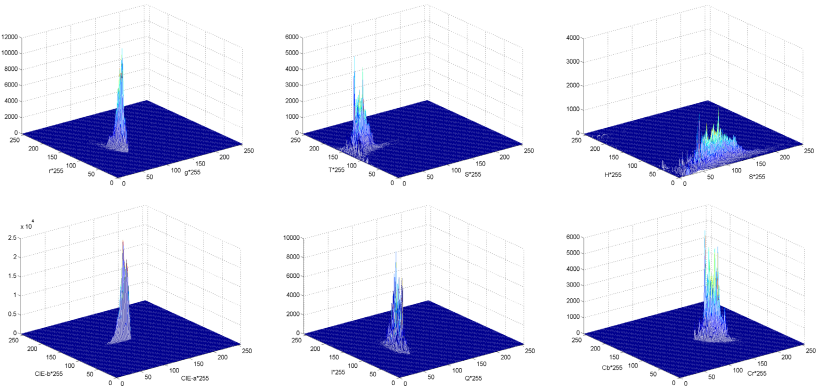


Fig. 1. Cumulative histograms of the training skin color pixels in different chrominance spaces: normalized r-g, T-S, H-S, CIE-ab, I-Q, Cb-Cr

2.2 Raw Detection Rule

Using confining detection rules to describe the skin cluster is very attractive because of its simplicity and low computing consumption. However, detection rules alone are not satisfied. But, as a pre-processing step, they can eliminate some non-skin pixels without losing many skin pixels, which helps to diminish computing cost. This also means that complexities of backgrounds are reduced.

After analyzing the skin color histogram in the normalized r-g chrominance space, we make some detection rules. Projected into the r-g plane, the distribution of skin color forms a approximate ellipse. So a external rectangle of the ellipse is employed to bound the skin color class. It should keep more than 95% of skin color pixels in original images. Meanwhile, we adopt and change some rules of [2]. Lastly, our detection rules are the following

$$\begin{aligned} &\text{pixel } \boldsymbol{x} \text{ will be preserved if} \\ &R > 80 \quad \text{AND} \quad G > 40 \quad \text{AND} \quad B > 20 \quad \text{AND} \\ &80 < r < 155 \quad \text{AND} \quad 60 < g < 105 \,, \end{aligned} \tag{4}$$

where r and g are normalized by 255.

3 Image Segmentation

3.1 Fuzzy C-Means Clustering Algorithm

When the FCM algorithm is applied to cluster data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ into c classes, it is derived by minimizing the following cost function

$$J_m = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d^2(\boldsymbol{x}_j, \boldsymbol{v}_i) \,, \tag{5}$$

where u_{ij} is the membership function of $\boldsymbol{x}_j$ to the ith class, subjecting to the constrains [9]

$$\begin{cases} u_{ij} \in [0,1], \quad i = 1,2,\ldots,c \text{ and } j = 1,2\ldots,n \\ \sum_{i=1}^{c} u_{ij} = 1, \quad j = 1,2,\ldots,n \\ 0 < \sum_{j=1}^{n} u_{ij} < n, \quad i = 1,2,\ldots,c \end{cases} \tag{6}$$

and

$$d^2(\boldsymbol{x}_j, \boldsymbol{v}_i) = (\boldsymbol{x}_j - \boldsymbol{v}_i)^T A(\boldsymbol{x}_j - \boldsymbol{v}_i)\ . \tag{7}$$

A is a $p \times p$ positive definite matrix; p is the dimension of $\boldsymbol{x}_j$; $\boldsymbol{v}_i$ is the centroid of the ith class [8]. If A is replaced by the identity matrix I, d will be the Euclidean distance between $\boldsymbol{x}_j$ and $\boldsymbol{v}_i$. m is the fuzziness index [8], $m \in [1.5, 2.5]$ is probably the best choice and the midpoint $m = 2$ is often the preferred choice [10].

The solution of FCM algorithm is a iterative process. The execution process is expressed as follows [8]:

1. Initialize cluster centroids $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c)$.
2. Fuzzy membership functions is computed by

$$u_{ij} = \frac{\left(\frac{1}{d^2(\boldsymbol{x}_j, \boldsymbol{v}_i)}\right)^{\frac{1}{m-1}}}{\sum_{i=1}^{c} \left(\frac{1}{d^2(\boldsymbol{x}_j, \boldsymbol{v}_i)}\right)^{\frac{1}{m-1}}}, \quad j = 1,2,\ldots,n\ . \tag{8}$$

3. Update centroids $\boldsymbol{V}$ by

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \boldsymbol{x}_j}{\sum_{j=1}^{n} u_{ij}^m}, \quad i = 1,2,\ldots,c\ . \tag{9}$$

4. Repeat until the value of J_m is no longer decreasing.

If $d(\boldsymbol{x}_i, \boldsymbol{v}_j) = 0$, the membership function u_{ij} cannot be computed by Eq. 8. It is then defined as

$$\begin{cases} u_{ij} = 0, & \text{if } d(\boldsymbol{x}_j, \boldsymbol{v}_i) \neq 0 \\ u_{ij} = 1, & \text{if } d(\boldsymbol{x}_j, \boldsymbol{v}_i) = 0 \end{cases}\ . \tag{10}$$

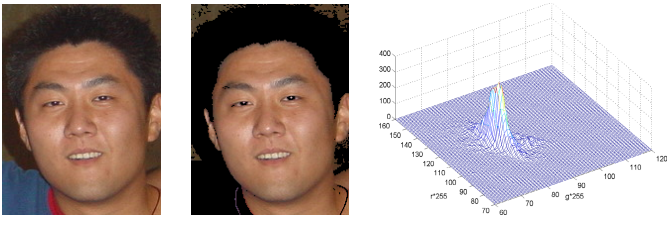
Considering that the number of classes cannot be known previously for each image, we adopt Xie-Beni cluster validity criteria S function [8] to measure the average compactness and separation of clusters and to decide the optimal number of classes. The Xie-Beni criteria is described by

$$S = \frac{J_m}{n \min_{i,k} \|\boldsymbol{v}_i - \boldsymbol{v}_k\|^2}, \quad i,k = 1,2\ldots,c\ , \tag{11}$$

where $c = 2,3,\ldots,c_{max}$, and the value of c corresponding to the minimal S is the optimal number of classes.

3.2 Improved Fuzzy C-Means Algorithm

In this study, a approximate method, valid and fast, is used to initialize cluster centroids. The histogram of a pre-processed image shapes several mountains. An example can be seen in Fig. 2(c). As color vectors corresponding to peaks of mountains are in the interior of samples and distributions of samples near peaks are tight, these color vectors are initialized as initial centroids, and the number of local peaks is used to set the upper limit of the number of classes c_{max} in the cluster validity algorithm.



(a) (b) (c)

Fig. 2. An example of pre-processed image histograms. (a) The original image; (b) the pre-processed image; (c) the histogram of the pre-precessed image.

In addition, we propose to use the local characteristic [7] to restrain illumination effect in segmentation. The local characteristic is the conditional probability of pixel $\boldsymbol{x}_j$ classified to the ith class. It is defined by [7]

$$f(i_j|\eta_j) = \frac{e^{\beta\delta_j(i)}}{\sum_{i=1}^{c} e^{\beta\delta_j(i)}}, \quad j = 1, 2, \ldots, n\ . \tag{12}$$

Here, i_j represents that pixel $\boldsymbol{x}_j$ is classified into the ith class; η_j represents categories of neighborhoods (usually 8-neighbors) of pixel $\boldsymbol{x}_j$; $\delta_j(i)$ is the number of pixels in class i of neighborhoods. The influence of β on clustering is illustrated in [7]. Let $\beta = 1.5$ in our algorithm and it seems to work well in experiments. Furthermore, $\boldsymbol{x}_j$ belongs to the class i^* according to the following rule

$$i^* = \underset{i}{\operatorname{argmax}}\, u_{ij}, \quad i = 1, 2, \ldots, c \text{ and } j = 1, 2, \ldots, n\ . \tag{13}$$

Therefore, the steps of FCM becomes

1. Initialize cluster centroids $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c)$ using the approach above.
2. Classify $\boldsymbol{x}_j$ by $u_{ij}^{(p-1)}$ using the classifying rule Eq. 13, where p is the iteration index. Then, compute fuzzy membership functions using

$$u_{ij}^{(p)} = \frac{\left(\frac{1}{d^2(\boldsymbol{x}_j, \boldsymbol{v}_i)}\right)^{\frac{1}{m-1}} \times f(i_j|\eta_j)}{\sum_{i=1}^{c}\left[\left(\frac{1}{d^2(\boldsymbol{x}_j, \boldsymbol{v}_i)}\right)^{\frac{1}{m-1}} \times f(i_j|\eta_j)\right]}, \quad j = 1, 2, \ldots, n\ . \tag{14}$$

3. Update centroids $\boldsymbol{V}$ using

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{n} u_{ij}^{(p)m} \boldsymbol{x}_j}{\sum_{j=1}^{n} u_{ij}^{(p)m}}, \quad i = 1, 2, \ldots, c\,. \tag{15}$$

4. Repeat until the value of J_m is no longer decreasing.

A comparison of the performance of the original FCM algorithm and the improved FCM algorithm is shown in Fig. 3. In order to distinguish conveniently, different gray values are used to denote different patches in Fig.3(c) and Fig.3(d).

(a) (b) (c) (d)

Fig. 3. An example of image segmentation. (a) The original image; (b) the result after pre-processing; (c) the result of the original FCM; (d) the result of the improved FCM.

4 Experiments

Skin patches will be decided with the following method. Initially, label each patch of segmented images, and the same class which do not connect each other in space will be labeled as different patches. Next, compute the ratio of skin pixels in each patch. A patch will be considered as a skin patch, if its ratio exceeds the threshold τ. This condition is defined as

$$PATCH_i \text{ is classified as a skin patch if} \quad \frac{S_i}{P_i} \geq \tau\,, \tag{16}$$

where S_i and P_i are the number of skin pixels and total pixels in $PATCH_i$ respectively.

Another 160 pictures downloaded from Internet form our test database. These pictures are also chose randomly. Different values of θ_s and τ will definitely lead to distinct detection results. We first draw a ROC curve shown in Fig. 4 corresponding to θ_s and $\tau = \frac{1}{3}$. Table 1 lists the ratios of correct detection and false detection corresponding to different τ and $\theta_s = 0.15, 0.2$.

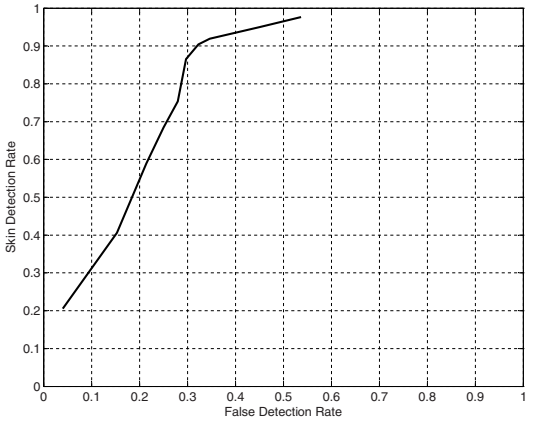


Fig. 4. ROC curve corresponding to the threshold θ_s and $\tau = \frac{1}{3}$

Table 1. Ratios of correct detection and false detection with $\tau = \frac{1}{4}, \frac{1}{3}, \frac{1}{2}$ and $\theta_s = 0.15, 0.2$

	Correct Detection Ratios		False Detection Ratios	
	$\theta_s = 0.15$	$\theta_s = 0.2$	$\theta_s = 0.15$	$\theta_s = 0.2$
$\tau = \frac{1}{4}$	92.3%	91.7%	38.1%	32.5%
$\tau = \frac{1}{3}$	90.4%	86.5%	32.3%	29.7%
$\tau = \frac{1}{2}$	82.8%	75.2%	25.5%	23.9%

(a) (b) (c) (a) (b) (c)

Fig. 5. Examples of experimental results. (a) Original images; (b) segmentation results; (c) detection results of the proposed algorithm.

Fig. 5 are some examples of our algorithm performance with $\theta_s = 0.15$ and $\tau = \frac{1}{3}$. The images in Fig. 5(b) are segmentation results, and different gray values label different patches. Although the edges of skin patches in results (Fig. 5(c)) are not detected accurately, the results are acceptable.

5 Conclusions

The paper presents a new region-based skin detection algorithm. Images are segmented into patches using fuzzy C-means algorithm improved by employing the local characteristic which helps to overcome influences of illumination. Meanwhile, in order to decrease computing cost, local peak values of histograms are initialized as cluster centroids. Lastly, a skin patch is decided by checking its ratio of skin pixels to total pixels in the patch.

Acknowledgments

The authors would like to acknowledge support from Shandong Provincial Natural Science Foundation under contract No.Z2005G03.

References

1. Kruppa, H., Bauer, M.A., Schiele, B.: Skin Patch Detection in Real-World Images. Annual Symposium for Pattern Recognition of the DAGM 2002, Lecture Notes in Computer Science, vol. 2449. Springer-Verlag (2002) 109–116
2. Kovac, J., Peer, P., Solina, F.: Human Skin Colour Clustering for Face Detection. EUROCON 2003. Computer as a Tool. The IEEE Region 8, Vol. 2. (2003) 144–148
3. Terrillon, J.-C., Shirazi, M.N., Fukamachi, H., Akamatsu, S.: Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic Detection of Human Faces in Color Images. Proceedings of Fourth IEEE International Conference on Automatic Face and Gesture Recognition. (2000) 54–61
4. Vezhnevets, V., Sazonov, V., Andreeva, A.: A Survey on Pixel-Based Skin Color Detection Techniques. Proceedings of 13th International Conference of Computer Graphics and Visualization Graphicon–2003. (2003) 85–92
5. Jones, M.J., Rehg, J.M.: Statistical Color Models with Application to Skin Detection. International Journal of Computer Vision, vol. 46(1). (2002) 81-96
6. Yang, M.-H., Ahuja, N.: Detecting Human Faces in Color Images. International Conference on Image Processing (ICIP), vol. 1. (1998) 127–130
7. Zhang, J., Modestino, J.W., Langan, D.A.: Maximum-Likelihood Parameter Estimation for Unsupervised Stochastic Model-Based Image Segmentation. IEEE Trans. Image Processing, vol. 3(4). (1994) 404–420
8. Xie, X.L., Beni, G.: A Validity Measure for Fuzzy Clustering. IEEE Trans. Pattern Anal. Machine Intell., vol. 13(8). (1991) 841–847
9. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. 2nd edition. Academic Press. (2003)
10. Pal, N.R., Bezdek, J.C.: On Cluster Validity for Fuzzy c-Means Model. IEEE Trans. Fuzzy Systems, vol. 3(3). (1995) 370–379

Supportive Utility of Irrelevant Features in Data Preprocessing

Sam Chao, Yiping Li, and Mingchui Dong

Faculty of Science and Technology, Unversity of Macau
Av. Padre Tomás Pereira S.J., Taipa, Macao
{lidiasc,ypli,dmc}@umac.mo

Abstract. Many classification algorithms degrade their learning performance while irrelevant features are introduced. Feature selection is a process to choose an optimal subset of features and removes irrelevant ones. But many feature selection algorithms focus on filtering out the irrelevant attributes regarding the learned task only, not considering their hidden supportive information to other attributes: whether they are really irrelevant or potentially relevant? Since in medical domain, an irrelevant symptom is treated as the one providing neither explicit information nor supportive information for disease diagnosis. Therefore, the traditional feature selection methods may be unsuitable for handling such critical problem. In this paper, we propose a new method that selecting not only the relevant features, but also targeting at the latent useful irrelevant attributes by measuring their supportive importance to other attributes. The empirical results demonstrate a comparison of performance of various classification algorithms on twelve real-life datasets from UCI repository.

Keywords: supportive relevance, latent correlation, data preprocessing, feature selection, data mining.

1 Introduction

The objective of a classification problem is to accurately and efficiently map an input instance to an output class label, according to a set of labeled instances. While many aspects affect the classification performance, among all, data is a prominent one. More data no longer means more discriminative power; contrarily, they may increase the complexity and uncertainty to the learning algorithms, thus burden with heavy computational cost. On the other hand, less data may be either over-fitting or cause the learning algorithms unable to learn meaningful results. In order to learn efficiently, one of the data preprocessing algorithms – feature selection, which aims to optimize the data to be learned, can be involved to overcome such obstacles.

Various state-of-the-art feature selection algorithms are described in [1], as well as their evaluations and comparisons in [2], [3]. The existing feature selection methods are mainly divided into two categories: filter and wrapper. Filter approach evaluates the selected features independently, does not take the learning algorithm into the

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 425–432, 2007.

evaluation process. The advantage of this approach is its reasonable computational complexity and cost; while wrapper approach involves the learning algorithm as part of the evaluation function, for each subset a classifier is constructed and used for evaluating the goodness of generated subset. The advantage of this approach is its high and reliable classification accuracy. Furthermore, most feature selection methods used sequential forward/backward search [4], [5] to construct the best subset of features by starting with an empty set or a full feature set. Then, the search goes on by adding or deleting one more feature each time to or from the best feature subset, until no more performance improvement. In this paper, we adopted filter approach with sequential forward search in our method.

In the next sections, we describe our novel method LUIFS – Latent Utility of Irrelevant Feature Selection in detail, as well as the feature selection problem under medical domain. The evaluation of the proposed method on some real-life datasets is performed in the section following. Finally, we discuss the limitations of the method and present the directions for our further research.

2 Feature (Attributes) Selection

Feature selection is a process that chooses an optimal subset of features according to a certain criterion [1]. Features can be categorized into: relevant, redundant, and irrelevant. An irrelevant feature does not affect the target concept in any way; while a redundant feature does not add anything new to the target concept and a relevant feature is neither irrelevant nor redundant to the target concept [6].

2.1 Feature Selection Problem

The ordinary feature selection methods focus on selecting relevant attributes and filtering out the irrelevant ones regarding the class attribute (learned task) only. This may sometimes lose the significant supportive information hidden in the irrelevant features. For instance, a forward selection method recursively adds a feature x_i to the current optimal feature subset *OptimalA*, among those that have not been selected yet in feature set *A*, until a stop criterion is met. In each step, the feature x_i that makes evaluation measure *W* be greater is added to the subset *OptimalA*. Starting with *OptimalA* ={}, the forward step is illustrated in equation (1).

$$\text{OptimalA} := \text{OptimalA} \cup \{\text{A} \setminus \text{OptimalA} \mid \text{W}(\text{OptimalA} \cup \{\text{x}_\text{i}\}) \text{ is maximum}\} . \quad (1)$$

The main disadvantage of the above formula is that it is impossible to have in consideration certain basic interactions among features, i.e., if x_1, x_2 are such interacted attributes, that $W(\{x_1, x_2\}) >> W(\{x_1\}), W(\{x_2\})$, neither x_1 and x_2 could be selected, in spite of being very useful [7]. This is because most feature selection methods assume that the attributes are independent rather than interactive, hence their hidden correlations have been ignored. However, an attribute that is completely useless by itself can provide a significant performance improvement when taken with others. Two attributes that are useless by themselves can be useful together [8], [9].

2.2 A Medical Example

In medical domain, a single symptom seems useless regarding diagnosis, may be potentially important by providing supportive information to other symptoms. For example, when learning a medical dataset for diagnosing cardiovascular disease, suppose a dataset contains attributes such as patient *age*, *gender*, *height*, *weight*, *blood pressure*, *pulse*, *ECG result*, *chest pain*, etc., during feature selection process, most often attribute *age* or *height* alone will be treated as the least important attributes and discarded accordingly. However, in fact attribute *age* and *height* together with *weight* may express potential significance: whether a patient is *overweight*? On the other hand, although attribute *blood pressure* may be treated as important regarding classifying a cardiovascular disease, while together with a useless attribute *age*, they may reveal more specific meaning: whether a patient is *hypertensive*? As we know that a person's blood pressure is increasing as his/her age increasing. The standard for diagnosing hypertension is a little bit different from young people (regular is 120-130mmHg/80mmHg) to the old people (regular is 140mmHg/90mmHg) [10]. Obviously, the compound features *overweight* and/or *hypertensive* have more diagnostic power regarding classifying a cardiovascular disease than the individual attributes *weight* and *blood pressure*. It is also proven that a person is *overweight* or *hypertension* may have more probabilities to obtain a cardiovascular disease [11].

According to the above example, sometimes a useless symptom by itself may become indispensable when combined with other symptoms. To overcome such problem, in this paper our new feature selection method LUIFS that focuses on discovering the potential importance for those irrelevant attributes rather than ignoring them. The method takes the attributes' interdependences into consideration, in order to uncover the hidden supportive information possessed by the irrelevant attributes. Since in medical domain, compound symptoms always could reveal more accurate diagnostic results.

3 Latent Supportive of Irrelevant Attribute (LSIA)

Our preprocessing method LUIFS mainly focuses on discovering the potential usefulness of LSIA and recruiting them into the optimal feature subset for final classification. It takes the inter-correlation between irrelevant attributes and other attributes into consideration to measure the latent importance of the irrelevant attributes. As we believe that in medical field an irrelevant attribute is the one that providing neither explicit information nor supportive or implicit information.

In [12], Pazzani proposed a similar method to improve the Bayesian classifier by searching for dependencies among attributes. However, his method has several aspects that are different from ours: (1) it is restricted under the domains on which the naïve Bayesian classifier is significantly less accurate than a decision tree learner; while our method aims to be a preprocessing tool for most learning algorithms; (2) It used wrapper model to construct and evaluate a classifier at each step; while a simpler filter model is used in our method, which minimized the computational complexity and cost; (3) His method created a new compound attribute replacing the original two attributes in the classifier after joining attributes. This may result in a less accurate

classifier, because joined attributes have more values and hence there are fewer examples for each value, as a consequence, joined attributes are less reliable than the individual attributes. Contrarily, our method adds the potential useful irrelevant attributes into the optimal feature subset to assist increasing the importance of the other relevant attributes, instead of joining them.

Our method generates an optimal feature subset in two phases: (1) Relevant Attributes Seeking: for each attribute in a dataset, work out its relevant weight regarding the target class, selects the one whose weight is greater than a pre-defined threshold. For the ones whose weights are smaller than the threshold (irrelevant attributes), carry out the second phase; (2) LSIA Discovery: for each irrelevant attribute, determine its supportive importance by performing a multivariate interdependence measure that combined with another attribute. There are two cases that such an irrelevant attribute becomes a potentially supportive relevant attribute and will be selected into the optimal feature subset. *Case 1*: if a combined attribute is a relevant attribute and already be selected into the optimal feature subset, then the combinatorial weight should be greater than the relevant weight of the combined attribute; *case 2*: if a combined attribute is an irrelevant attribute too whose weight is smaller than the threshold, then the combinatorial weight should be greater than the pre-defined threshold, such that both irrelevant attributes become relevant and will be selected into the optimal feature subset accordingly.

3.1 Relevant Attributes Seeking (RAS)

In this phase, each attribute is calculated its relevant weight respect to the target class. Information gain theory [13] is used as the measurement, which may find out the most informative (important) attribute A relative to a collection of examples S and is defined as:

$$InfoGain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \ . \tag{2}$$

where $Values(A)$ is the set of all distinct values of attribute A; S_v is the subset of S for which attribute A has value v, that is $S_v = \{s \in S \mid A(s) = v\}$. And $Entropy(S)$ is:

$$Entropy(S) = -\sum_{i \in C} p(S_i) log(p(S_i)) \ . \tag{3}$$

where $p(S_i)$ is the proportion of S belonging to class i.

Attributes are first sorted in descending order, from the most important one (with the highest information gain) to the least useful one. Meanwhile, a threshold ϖ is introduced to distinguish the weightiness of an attribute. The value of a threshold either too high or too low may cause the attributes insufficient or surplus. Therefore it is defined as a mean value excluding the ones with maximum and minimum information gain, in order to eliminate as much bias as possible.

An attribute A will be selected into the optimal feature subset if $InfoGain(S, A) > \varpi$; otherwise, it will be filtered out and fed into the second phase as the input. In

addition, if an attribute *A* is a numeric attribute, it is discretized first by using the method of [14] to avoid the bias in information gain algorithm that favors attributes with many values. This RAS phase requires only linear time in the number of the given features *N*, i.e. $O(N)$.

3.2 LSIA Discovery

This phase is the key spirit of LUIFS, since its objective is to uncover the usefulness of the latent or supportive relevant attributes. It is targeted at the irrelevant attributes that filtered out from RAS phase, looking for their latent utilities in supporting other attributes. To determine whether an irrelevant attribute is potentially important or not, we measure the interdependence weight between it and another attribute regarding the class attribute. We use relief theory [15], [16], which is a feature weighting algorithm based on distance measure for estimating the quality of attributes, such that it is able to discover the interdependencies between attributes. Our method adopts the combinatorial relief in equation (4), which measures the interdependent weight between a pair of attributes rather than a single attribute regarding the class attribute. It approximates the following probability difference:

$$W[a_i + a_j] = P(\text{different value of } a_i + a_j | \text{nearest instances from different class } c_i) - P(\text{different value of } a_i + a_j | \text{nearest instances from same class } c_j) \quad . \tag{4}$$

where a_i is an irrelevant attribute, whose information gain measure in equation (2) is smaller than the mean threshold ϖ and is filtered out in RAS phase; a_j is a combined attribute either relevant or irrelevant; c_i and c_j are the different values of class attribute *C*. $W[a_i+a_j]$ is the combinatorial interdependent weight between attributes a_i and a_j regarding the class attribute *C*, and *P* is a probability function for the weight of feature pair a_i and a_j. Equation (4) measures the level of hidden supportive importance for an irrelevant attribute to another attribute, hence the higher the weighting, the more information it will provide, such that the better the diagnostic results. According to our hypotheses, an irrelevant attribute a_i may become latent relevant if there exists another attribute a_j, where $a_i \neq a_j$, so that the combinatorial interdependent measure $W[a_i+a_j] > W[a_j]$ if a_j is an explicit relevant attribute and already be selected into the optimal feature subset; or $W[a_i+a_j] > \varpi$ (a pre-defined threshold) if a_j is an irrelevant attribute also.

Unlike the RAS phase, the complexity of this LSIA Discovery phase is no longer simple linear in time. In the worst case, if there is only one important attribute was selected after RAS phase, that is, there are $(N-1)$ irrelevant attributes were ignored and unselected. For each irrelevant attribute $a_i \in$ *UnselectedAttributes*, calculate its interdependent weight with another attribute. Again in the worst case, if a_i could not encounter an attribute that makes it becoming useful, then the process should be repeated for $(N-1)$ times. Whereas the algorithm is symmetric, i.e. $W[a_i+a_j] = W[a_j+a_i]$, so the total times should be in half respect to the number of *UnselectedAttributes*, which equals to $(N-1)/2$. Therefore, the complexity of such phase for irrelevant attributes is $(N-1)*(N-1)/2$ for the worst case, i.e. $O(N^2)$. Nevertheless the data preprocessing is typically done in an off-line manner [17], in

the meantime, the capacity of the hardware components increase while the price of them decrease. Because of these, the execution time of an algorithm becomes less important compared with its final class discriminating performance.

4 Experiments

We have evaluated the effectiveness of LUIFS on twelve real-life datasets from UCI repository [18], the detailed characteristics of each dataset is listed in Table 1. In the experiment, two learning algorithms ID3 [19] and C4.5 [20], [21] are involved as the evaluation algorithms. LUIFS is used as the preprocessing method for them. The last column LRA in Table 1 indicates the number of irrelevant attributes becoming useful, and is additionally selected into the optimal feature subset for learning.

Table 1. Bench-mark datasets from UCI repository

Dataset	Features		Instance Size		Class	LRA
	Numeric	Nominal	Training	Testing		
Cleve	6	7	202	101	2	**2**
Hepatitis	6	13	103	52	2	**7**
Hypothyroid	7	18	2108	1055	2	**8**
Heart	13	0	180	90	2	**2**
Sick-euthyroid	7	18	2108	1055	2	**7**
Auto	15	11	136	69	7	**3**
Breast	10	0	466	233	2	**2**
Diabetes	8	0	512	256	2	**3**
Mushroom	0	22	5416	2708	2	**1**
Parity5+5	0	10	100	1024	2	**4**
Corral	0	6	32	32	2	**2**
Led7	0	7	200	3000	10	**1**

In order to make clear comparison, experiments of learning methods without feature selection (NoFS) and with information gain attribute ranking method (ARFS) are performed, as well as LUIFS. Table 2 summarizes the results in error rates of two algorithms for various methods respectively. As manifested in Table 2, LUIFS does help in increasing the classification accuracy significantly by adding the indicated number of LRA into the optimal feature subset. It improves the performance on 7 and 6 datasets for ID3 and C4.5 learning algorithms respectively. And it maintains the performance as same as ARFS on 4 and 5 datasets for ID3 and C4.5 respectively, nevertheless, the results are still better than the methods with NoFS for most of these datasets. Although LUIFS slightly decreases the performance on one dataset *Heart* among all in compared with ARFS by adding one latent relevant attribute, the results are still better than the methods with NoFS. This may be due to the dataset contains numeric attributes only, which needs to perform an additional discretization prior the LSIA Discovery phase. Such step increases the unexpected uncertainty to the attribute being correlated, hence increases the error rate accordingly.

Table 2. Comparisons for results in error rate (%) of ID3 and C4.5

Dataset	ID3 algorithm			C4.5 algorithm		
	NoFS	ARFS	LUIFS	NoFS	ARFS	LUIFS
Cleve	35.64	23.762	22.772	20	22.2	19.3
Hepatitis	21.154	15.385	13.462	15.6	7.9	7.9
Hypothyroid	0.948	0.190	0.190	1.1	0.4	0.4
Heart	23.333	13.333	18.889	17.8	14.4	15.6
Sick-euthyroid	3.791	0	0	2.5	2.5	2.4
Auto	26.087	18.841	18.841	27.1	21.9	21.8
Breast	5.579	6.438	5.579	5.7	5.6	5.4
Diabetes	Program Error	35.156	32.131	30.9	30.1	30.1
Mushroom	0	0	0	0	0	0
Parity5+5	49.291	50	49.291	50	50	50
Corral	0	12.5	0	9.4	12.5	9.4
Led7	33.467	42.533	32.8	32.6	43.7	32.3

5 Conclusions and Future Research

In this paper, we have proposed a novel data preprocessing method LUIFS focused on the discovery of the potential usefulness of irrelevant attributes. The method can be used before most learning algorithms. The empirical evaluation results presented in the paper indicate significant evidence that LUIFS can improve the final classification performance by adding the discovered latent important attributes into the optimal feature subset for learning process. However, in this work only ARFS method has been implemented to make the comparison. The existing feature selection methods can be involved further in our comparisons. In addition, our method has slow execution time if a dataset contains more than several hundreds of features. Our experiment was performed with ID3 and C4.5 learning algorithms on smaller to medium sized datasets, for further comparisons, we plan to perform the experiments on large datasets with other learning algorithms, such as Naïve Bayes [22], 1R [23], GA [24], or clustering methods, etc. On the other hand, limitations should be resolved to be able to handle large datasets efficiently and effectively.

References

1. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publisher (2000)
2. Molina, L.C., Belanche, L., Nebot, A.: Feature Selection Algorithms: A Survey and Experimental Evaluation. Proceedings of IEEE International Conference on Data Mining, ICDM (2002) 306-313
3. Dash, M., Liu, H.: Feature Selection for Classification. Intelligent Data Analysis, Vol. 1(3) (1997) 131-156

4. Miller, A.J.: Subset Selection in Regression. Chapman and Hall (1990)
5. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice-Hall International (1982)
6. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. In Proceedings of the Eleventh International Conference on Machine learning (1994) 121-129
7. Molina, L.C., Belanche, L., Nebot, A.: Feature Selection Algorithms: A Survey and Experimental Evaluation. Proceedings of IEEE International Conference on Data Mining, ICDM (2002) 306-313
8. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research Vol. 3 (2003) 1157-1182
9. Caruana, R., Sa, V.R.: Benefiting from the Variables that Variable Selection Discards. Journal of Machine Learning Research Vol. 3 (2003) 1245-1264
10. The Hypertensive Research Group of Hear Internal Medicine Department of People's Hospital of Beijing Medical University:Hundred Questions and Answers in Modern Knowledge of Hypertension (1998)
11. Jia, L., Xu, Y.: Guan Xing Bing De Zhen Duan Yu Zhi Liao. Jun Shi Yi Xue Ke Xue Chu Ban She (2001)
12. Pazzani, M.J.: Searching for Dependencies in Bayesian Classifiers. In Proceedings of the Fifth International Workshop on AI and Statistics. Springer-Verlag (1996) 424-429
13. Zhu, X.L.: Fundamentals of Applied Information Theory. Tsinghua University Press (2000)
14. Fayyad, U.M., Irani, K.B.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (1993) 1022–1027
15. Kira, K., Rendell, L.: A Practical Approach to Feature Selection. In Proceedings of International Conference on Machine Learning. Aberdeen, Morgan Kaufmann (1992a) 249-256
16. Kira, K., Rendell, L.: The Feature Selection Problem: Traditional Methods and New Algorithm. In Proceedings of AAAI'92. San Jose, CA.: AAAI Press (1992b)
17. Jain, A., Zongker, D.: Feature Selection: Evaluation, Application and Small Sample Performance. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19 (1997) 153–158
18. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. Department of Information and Computer Science, University of California (1998) [http://www.ics.uci.edu/~mlearn/MLRepository.html]
19. Quinlan, J.R.: Induction of Decision Trees. Machine Learning, Vol. 1 (1986) 81-106
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann (1993)
21. Quinlan, J.R.: Improved Use of Continuous Attributes in C4.5. Journal of Artificial Intelligence Research, Vol. 4 (1996) 77-90
22. Langley, P., Iba, W., Thompsom, K.: An Analysis of Bayesian Classifiers. In Proceedings of the tenth national conference on artificial intelligence. AAAI Press and MIT Press (1992) 223-228
23. Holte, R.C.: Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning, Vol. 11 (1993)
24. Vafaie, H., Imam, I.F.: Feature Selection Methods: Genetic Algorithms vs. Greedy like Search. In Proceedings of International Conference on Fuzzy and Intelligent Control System (1994)

Incremental Mining of Sequential Patterns Using Prefix Tree*

Yue Chen[1,2], Jiankui Guo[1], Yaqin Wang[1,2], Yun Xiong[1], and Yangyong Zhu[1]

[1] Department of Computing and Information Technology, Fudan University, Shanghai, China 200433
[2] Computer Science and Technology School, Soochow University, Suzhou, China
041021067@fudan.edu.cn

Abstract. This paper fist demonstrates that current PrefixSpan-based incremental mining algorithm IncSpan+ which is proposed in PAKDD05 cannot completely mine all sequential patterns. Then a new incremental mining algorithm of sequential patterns using prefix tree is proposed. This algorithm constructs a prefix tree to represent the sequential patterns, and then continuously scans the incremental element set to maintain the tree structure, using width pruning and depth pruning to eliminate the search space. The experiment shows this algorithm has a good performance.

1 Introduction

The goal of Sequential pattern mining is to find frequent subsequences from a sequence database [1]. In many domains, the contents of databases are updated incrementally. In order to get all sequential patterns, the mining algorithm has to be run whenever the database changes, because that some sequences which were not frequent in old database may become frequent in updated database. Obviously, to discover sequential patterns from scratch every time is ineffective. This leads to the study of the incremental mining algorithm of sequential patterns. When new sequences are added into old databases, the incremental mining algorithm minimizes the computational and I/O costs by re-using the information from the previous mining results from old database.

Currently, several incremental mining algorithms of sequential patterns have been proposed. But most of them are priori-like, which would generate huge set of candidate sequences when the sequence database is huge. Chen [2] proposed an incremental mining algorithm based on PrefixSpan in KDD'04, named IncSpan. In PAKDD'05, Son [3] found that the IncSpan had some weakness, that is, it can not find complete sequential patterns. They classified these shortcomings and proposed a new algorithm called IncSpan+, which is an improvement of IncSpan. But we find that IncSpan+ also has the same weakness as in IncSpan, lacking the ability to find complete sequential patterns.

* Supported partially by national natural science foundation of China (No. 60573093), Hi-Tech research and development program of China (863 program, No. 2006AA02Z329), 973 project of China (No. 2005CB321905).

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 433–440, 2007.

In this paper, we first argue that in general, IncSpan+ cannot find complete set of sequential patterns. Therefore, we propose a new incremental mining algorithm of sequential patterns based on prefix tree, called PBIncSpan. PBIncSpan first constructs a prefix tree to represent the sequential patterns based on PrefixSpan, and then continuously scans the incremental element set to maintain the tree structure, using some advanced pruning techniques named width pruning and depth pruning to eliminate the search space.

The rest of this paper is organized as follows. Section 2 introduces the related work. In section 3, we introduce some basic concepts. Section 4 points out the non-completeness of the IncSpan+. PBIncSpan is proposed in section 5 in detail. We evaluate PBIncSpan in section 6. Finally, this paper is concluded in Section 7 .

2 Related Work

Sequential pattern mining was first introduced in [4]. Most of the general algorithms for sequential pattern mining are based on Apriori property. In order to reduce the huge set of candidate sequences, another approach that mining sequential patterns by pattern-growth is proposed by Pei [5], called PrefixSpan. Other algorithms include SPIRIT [6], MEMISP [7]. Accordingly, the general incremental sequential pattern mining algorithms can be divided into two main catalogs. One is priori-based; the other is projection-based. The former includes GSP+ [8], ISM [9], ISE [10] and SPADE[11]. As far as we know, the later only includes IncSpan and IncSpan+. The weakness of priori-based algorithms is that they have to store huge sequential index. PrefixSpan [1] is an efficient algorithm, using a projection-based, sequential pattern-growth approach to mine the sequential patterns. It can avoid generating huge candidate sequences. Hence, IncSpan is proposed for incremental mining over multiple database increments, taking the advantage of PrefixSpan. IncSpan+ makes an improvement to IncSpan. IncSpan+ claims that it can find complete sequential patterns. But our study finds this is incorrect. We prove this in section 4.

3 Preliminary Concepts

Let $I=\{i_1, i_2, \ldots, i_m\}$ be a set of items. A subset of I is named as itemset. A sequence $s=<e_1, e_2, \ldots, e_n>$ is an ordered list, where e_i is an itemset. e_i is also called an element of a sequence. A sequence $\alpha = <a_1, a_2, \ldots, a_n>$ is called a subsequence of another sequence $\beta = <b_1, b_2, \ldots, b_m>$ if there exists integers $1 \leq j_1 < j_2 < \ldots < j_n \leq m$ such that $a_1 \subseteq b_{j1}$, $a_2 \subseteq b_{j2}, \ldots$, $a_n \subseteq b_{jn}$. For brevity, we assume an element has only one item. When an element has multiple items, the results may be deduced by analogy.

A sequence database $D = \{s_1, s_2, \ldots, s_m\}$ is a set of tuples <sid, s>, where sid is a sequence id and s is a sequence. |D| denotes the number of sequences in D. The (absolute) support of a sequence α in D is the number of sequences in D which contain sequence α, denoted as support(α). The relative support of a sequence α in D is its absolute support divides by |D|. Given a positive integer min_support as the support threshold, if supports(α) > min_support , the sequence α is a sequential pattern in D. The task of sequential pattern mining is to find all sequential patterns in D.

In real world, many sequence databases update over time. Let **D** be the old database. Let **db** be the incremental part w.r.t D. Let **D'** be the updated database. obviously, D'=D+db. Incremental sequential pattern mining is to find all sequential patterns efficiently in D' when D updates to D'.

In general, an incremental mining algorithm should satisfy the following conditions [3]:Completeness and Efficiency.

There are three kinds of database updates:1)deleteing a sequence, 2)inserting a new sequence into D, denoted as INSERT. and 3)appending a new item or itemset to a existing sequence, denoted as APPEND. Like IncSpan ,IncSpan+ and other algorithms, we consider the updates of database only refer to INSERT and APPEND. Fig.1 shows the detail. We can regard the INSERT operator as a special case of APPEND, for inserting a new sequence is equal to append a new sequence to an empty sequence.

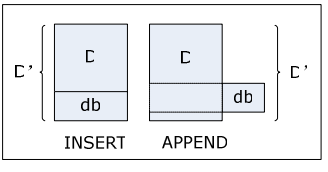


Fig. 1. Two kinds of database updates: one is INSERT, the other is APPEND

According to [2], given s=<$e_1,e_2...e_n$>∈ D, s_a=<$e'_1,e'_2...e'_m$>∈ db, if s'=s + s_a, s' is called an appended sequence of s. If s is empty, s'= s_a, which means insert a new sequence s_a to D. otherwise, it means append a sequence s_a to s. Obviously, s' ∈ D'. We define **LDB**={s'|s' ∈ D' and s' = s + s_a and s_a is not empty }.

4 The Weakness of IncSpan+

IncSpan+ proves that IncSpan provides incomplete results. The pruning technique that IncSpan+ used is as same as IncSpan, only correcting some error in IncSpan. IncSpan+ demonstrates that itself is correctness, which means IncSpan+ can find complete sequential patterns. The algorithm outline is list as follows [3]:

Algorithm 1. IncSpan+
Input: An updated database D', min_support, frequent set (FS) and SFS in D
Output: FS' SFS' in D'

```
(1)   SFS' =∅,FS'=∅ Determine LDB; calculate |D'|; adjust the min_support
(2)   Scan the whole D', add new frequent items into FS'; add new semi frequent items into SFS'
(3)   FOR each new item i in FS' DO PrefixSpan(i,D'|i, u*min_support, FS',SFS')
(4)   FOR each new item i in SFS' DO PrefixSpan(i,D'|i, u*min_support, FS',SFS')
(5)   FOR every pattern p in FS or SFS' DO
(6)     Check Δsup(p) = sup_db(p)
(7)     IF sup_D'(p)= sup_D(p)+ Δsup(p) ≥ min_support
(8)             INSERT(FS',p)
(9)             IF supLDB(p) ≥ (1-u)*min_support
(10)                    PrefixSpan(p,D'|p,u*min_support,FS',SFS')
(11)            ELSE IF sup_D'(p) ≥ u*min_support
(12)                    INSERT(SFS',p)
(13)                    PrefixSpan(p,D'|p,u*min_support,FS',SFS')
(14)  RETURN
```

The pruning technique used in algorithm 1 is based on theorem 1 in [2].

Theorem 1. For a frequent pattern p, if its support in LDB supLDB(p) <(1 – μ)*min_support, then there is no sequence p' having p as prefix changing from infrequent in D to frequent in D'. The proof of theorem 1 can be found in [2].

According to line (9) (10) in the algorithm of IncSpan+, the pruning is based on theorem 1 and can happen in every node. But theorem 1 only guarantees that a sequential pattern p cannot change from infrequent in D to frequent in D'. It cannot prevent an infrequent pattern p in D changes into semi frequent in D'. If a pattern p is currently infrequent in D, it has a chance to be semi frequent in D' when a new sequence appended. When D' is updated, p might be frequent in updated database D''. But IncSpan+ fails to discover this kind of patterns. So even a sequence p satisfies supLDB(p) < (1-μ)*min_support, it still needs to execute PrefixSpan(p, D'|p, u*min_sup,FS', SFS') to make sure that semi frequent patterns could be find completely. The IncSpan+ is not complete. We can illustrate by example 1.

Table 1. A running example of an sequence database

SeqID	Sequence
1	B A A C
2	A B A
3	A B A
4	A B C C

Example 1. Sequence database is shown in table 1[1]. Let min_support be 3, μ be 0.6. μ multiplies min_support is 1.8. According to algorithm 1, a sequence will be semi frequent or frequent when the support of that sequence is equal to or above 2. Assuming an item C is appended in the seqID 2. For a sequence AA, supLDB(AA) = 1 < (1-μ)*min_support=1.2, so PrefixSpan(AA,D'|AA, u*min_support, FS', SFS') is not executed. The semi frequent pattern AAC can not be added into SFS'.

5 PBIncSpan Algorithm

5.1 Prefix Tree of a Sequence Database

Constructing a prefix tree w.r.t a sequence database is as same as the procedure of mining sequential patterns in a sequence database using PrefixSpan algorithm. We illustrate how to build prefix tree by an example.

Example 2. Sequence database is shown in table 1. Let min_support be 2. The steps of construction a prefix tree w.r.t sequence database in table 1 is shown in Fig.2.

[1] For simplify, here we assume an element has only one item, so the sequence is shown as list of items, not itemsets.

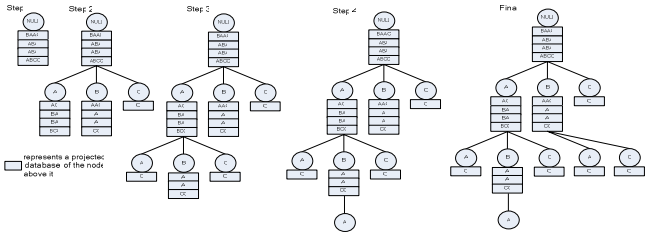


Fig. 2. The steps of construction prefix tree of the sequence database shown in table 1

5.2 Width Pruning

For every node in prefix tree as shown in Fig.2, its child node comes from the scan of the projected database with prefix of that node. Let α is a node in prefix tree, if α-projected database remains unchanged when D updates to D', its child node will not change.

Example 3. Sequence database is shown in table 1. Let min_support be 2. Assuming an item C is appended in the seqID 1. The A-projected database and B-projected database do not change after appending item C. Thus node A, B and their child nodes do not need to scan corresponding projected databases to find new sequential patterns when D updates to D'.

Definition 1(IASIDS). IASIDS (insert and append sequence id) is a set of seqID w.r.t the sequences in LDB. In example 3, the IASIDS={1}.

Theorem 2(width pruning). Let α be a prefix of any sequence in D, pSeqId={seqId | seqId ∈ α-projected database}. If IASIDS ∩pSeqId = ∅, then the node α and its child nodes in the prefix tree do not change when D updates to D'.

seqID plays important roles in pruning, the cost is that for every projected database, we have to check weather its seqIDs are included in IASIDS. One way to optimize the algorithm is to move the LDB to the top of D', maintaining a global various min_Seq_ID=size of LDB. When the first sequence id in a projected database, say α-projected database, is great than min_Seq_ID , which means α-projected database and LDB are mutually exclusive, we can use width pruning on node α.

5.3 Depth Pruning

Definition 2(incremental element set, IES). Let α be a prefix of any sequence in D, pSeqId={seqId | seqId ∈ α-projected database}.Let IPIDS = IASIDS∩pSeqId. IES is a set of items that append to D and their sequence ids are in IPIDS, denoted as IES_α

Theorem 3(depth pruning). Assuming node α's parent node does not insert any node as its child node during the scan of D', and $IES_\alpha \cap \{\alpha, \alpha\text{'s sibling nodes}\}=\emptyset$, then the node α and its child nodes in the prefix tree w.r.t D are as same as in the prefix tree w.r.t D'.

Example 4. Sequence database is shown in table 1. Let min_support be 2. Item B and F is appended in seqId 1, Item E is appended in seqId 2. Starting form the root, we

scan D'. There is no new node can be appended as root's child node. We calculate that IES_A={B,E,F}, IES_B={B,E,F}, and IES_C=∅. Because IES_C∩{C, C's sibling nodes}= IES_c∩{C, A, B} = ∅, C can be pruned (in this example, C can also be pruned by width pruning). Continue this step. The prefix BC can not be prune by using width pruning, but IES_{BC}={B,F}, BC's sibling node is sNode ={A, C}. IES_{BC}∩ {C}∩ sNode = ∅, so BC can be pruned by using depth pruning, do not need to scan BC-projected database.

5.4 The Algorithm of PBIncSpan

Based on width and depth pruning, PBIncSpan is given as follows:

Algorithm 2. PBIncSpan(root, D',db)
Input: D'; min_support; the root node of Prefix tree (PT) w.r.t D;
appended sequence db.
OutPut: Prefix tree w.r.t D'
(1) Calculate |D'|; adjust min_support; move LDB to the head of D'; min_Seq_ID=|LDB|;Flag = False (Flag is used to record whether a new node has been appended.)
(2) Scan D' and db once, get the Frequent Item Set (FIS) w.r.t the root of PT
(3) IF FIS= ∅ or db=∅
(4) RETURN root
(5) FOR every item t in FIS
(6) IF t ∈ root's child nodes
(7) Updated the support of t
(8) ELSE
(9) Create a new node t , insert to the root's child node
(10) Flag=True
(11) DoPBIncSpan(root,p,Flag,min_Seq_ID,D',db)
(12) Go through tree, out put the sequential patterns, delete node whose support is less than min_support
(13) RETURN root.

Algorithm 3. DoPBIncSpan(q,p,Flag,min_Seq_ID,D',db)
Input: parent node q, current node p, q's appending Flag (indicates whether q is appended with a new node), min_Seq_ID, q-projected database D' and db
Output: new subtree w.r.t D' after mining
(1) IF the first sequences id of p-projected database > min_Seq_ID
(2) RETURN //width pruning
(3) Scan D' and db, get p-projected database pD' and pdb respectively
(4) BOOL is_Depth_Prun = True;
(5) IF Flag=False //check whether depth pruning can be used here
(6) Scan pdb,caculate IES_p
(7) FOR every child node of q
(8) IF q∩IES_p≠ ∅
(9) is_Depth_Prun = False
(10) BREAK
(11) IF is_Depth_Prun = True
(12) RETURN //depth pruning
(13) BOOL pFlag = False
(14) Scan pD', get frequent item set FIS
(15) IF FIS = ∅ RETURN
(16) FOR every item t in FIS
(17) IF t∈ p's child nodes Update the support of t
(18) ELSE Create a new node t ; pFlag=True
(19) DoPBIncSpan(p,t,pFlag,min_Seq_ID,pD',pdb)
(20) RETURN

Theorem 4(Completeness of PBIncSpan). PBIncSpan outputs the complete set of sequential pattern.

6 Performance Study

The performance study is to check the efficiency of incremental mining algorithm. The incremental mining algorithm should use less time to mining sequential pattern than traditional one. Because PBIncSpan is based on PrefixSpan, we compare these two algorithms. We implement PrefixSpan using pseudoprojection. All experiments were conducted on a P4 2.5GHZ PC with 768 megabytes main memory, running windows XP Professional. All algorithms are implemented using C++ STL library with IDE vs2003.net.

For the real dataset, we get Gazelle from the author of BIBE. This dataset has been widely used in testing the performance of sequential pattern mining algorithm. This dataset contains 29,369 sequences, 87,546 sessions and 1423 items. More detail information could be found in [12].

We store the original sequence database D_1 as File 1. When D_1 appends or inserts new sequences, it becomes D_2 and is saved as File 2. The last updated sequence database D_k is saved as File k. The incremental ratio is m. when D_{k-1} updates to D_k, the inserted and appended sequences have a distribution d. In our test, we let k be four.

Every time when the sequence database is updated, we run PrefixSpan to mine sequential patterns from scratch. While PBIncSpan works in incremental way. The results are shown as in Fig 3 with different parameters.

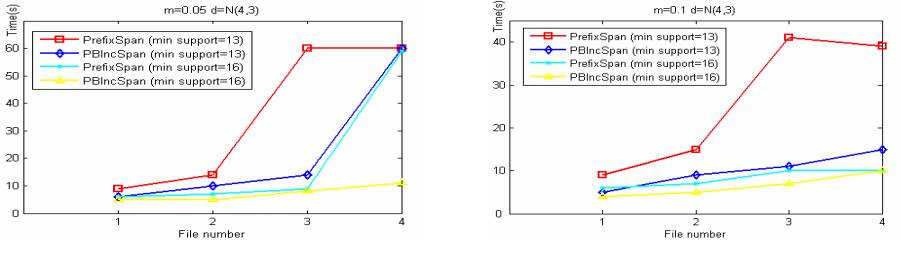


Fig. 3. The performance of PBIncSpan under different parameters

Form the figure 3, we can find that when the dataset is small, the performance of PBIncSpan just a slightly higher than PrefixSpan. This is because when dataset is small, the scan of the whole database is quickly, while pruning only saves a little time. When sequence database grows large, PBIncSpan outperformed PrefixSpan.

7 Conclusion and Future Work

This paper we first demonstrate that IncSpan+ can not find complete sequential pattern. Based on PrefixSpan, we use the prefix tree to maintain the sequential patterns. When database updated, we use width pruning and depth pruning to reduce the times

of scanning the updated database. But using prefix tree need more storage space when database is huge. Although we can store every branch into disk, but how to maintain a big tree is challenge. Another problem is that depth pruning is based on Apriori property and is not very effective when the prefix tree has lots of nodes. Further more, the stability of the algorithm should be strengthen. We will try to solve these problems in the future.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments and suggestions of this paper. This work is also supported by the Jiangsu Provincial Key Laboratory of Computer Information Processing Technology No.KJS0605.

References

1. Pei J., Han j.w., et al. mining sequential patterns by pattern-growth: the PrefixSpan approach. Knowledge and Data Engineering, IEEE Transactions On 16(11):1424-1440, 2004
2. Cheng H., Yan X., Han J.: Incremental mining of sequential patterns in large database. Proc. ACM KDD Conf. on Knowledge Discovery in Data, Washing ton(KDD'04),2004
3. Son Nguyen, X Sun, ME Orlowska Improvements of IncSpan: Incremental Mining of Sequential Patterns in Large Database. PAKDD 2005.
4. Agrawal R., Srikant R.:Mining sequential patterns: Generalization and performance improvements. In 5th EDBT,1996
5. J.Pei, J.Han, B.Mortazavi-Asl, et al. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. ICDE'01, pages 215-224,April 2001.
6. Garofalakis, M. N., Rastogi, R., and Shim, K. Spirit: Sequential pattern mining with regular expression constraints. In VLDB'99.
7. Lin, M.-Y. and Lee, S.-Y. Fast discovery of sequential patterns by memory indexing. In Proc. of 2002 DaWaK. 150-160, 2002
8. M. Zhang, B. Kao, D. Cheung, and C.L. Yip. Efficient algorithms for incremental update of frequent sequences. PAKDD2002, Taiwan, May 2002.
9. S.Parthasarathy, M.Zaki, M.Ogihara, and S.Dwarkadas. Incremental and interactive sequence mining. In Proc. of CIKM'99,Nov 1999
10. MASSEGLIA, F., PONCELET, P., and TEISSEIRE, M. Incremental mining of sequential patterns in large databases. Data & Knowledge Engineering, 2003,46(1):97-121.
11. Zaki, M. J. SPADE: An efficient algorithm for mining frequent sequences. Machine Learning 42, 1/2, 31-60, 2001
12. R. Kohavi, C. Brodley, B. Frasca, L. Mason, and Z. Zheng, "KDDCup 2000 Organizers' Report: Peeling the Onion," Proc. SIGKDD Explorations, vol. 2, pp. 86-98, 2000.

A Multiple Kernel Support Vector Machine Scheme for Simultaneous Feature Selection and Rule-Based Classification*

Zhenyu Chen[1,2] and Jianping Li[1,**]

[1] Institute of Policy & Management, Chinese Academy of Sciences, Beijing 100080, China
[2] Graduate University of Chinese Academy of Sciences, Beijing 100039, China
{zychen, ljp}@casipm.ac.cn

Abstract. In many applications such as bioinformatics and medical decision-making, the interpretability is important to make the model acceptable to the user and help the expert discover the novel and perhaps valuable knowledge hidden behind the data. This paper presents a novel feature selection and rule extraction method which is based on multiple kernel support vector machine (MK-SVM). This method has two outstanding properties. Firstly, the multiple kernels are described as the convex combination of the single feature basic kernels. It makes the feature selection problem in the context of SVM transformed into an ordinary multiple parameters learning problem. A 1-norm based linear programming is proposed to carry out the optimization of those parameters. Secondly, the rules are obtained in an easy way: only the support vectors necessary. It is demonstrated in theory that every support vector obtained by this method is just the vertex of the hypercube. Then a tree-like algorithm is proposed to extract the if-then rules. Three UCI datasets are used to demonstrate the effectiveness and efficiency of this approach.

1 Introduction

In the domain of data mining, it is very important for a model to make the results more acceptable to the user and help the expert more easily discover the novel and perhaps valuable knowledge or possible errors in the conclusions. As a popular machine learning method, support vector machine (SVM) has strong theoretical foundations and achieves success in many areas. As SVM is a "black-box" system, the explanation capacity hinders SVM from going further in the applications especially in the medical field.

Some researchers devote to propose SVM-based feature selection methods. In the greedy strategy [1] or pruning strategy [2], the features are added or removed

* This research has been partially supported by a grant from National Natural Science Foundation of China (#70531040), and 973 Project (#2004CB720103), Ministry of Science and Technology, China.

** Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 441–448, 2007.

according to some defined measure. It is their drawback that they are usually dependent on the threshold and the solutions are usually not optimal. Recently, the genetic algorithm [3] becomes the popular tool to search for the optimal feature subset. It needs to implement the optimization of SVM repeatedly and results in the expensive computational cost.

There are few papers published in the cases of rule extraction from SVM. A typical rule extraction approach treats SVM as a black-box and the output of SVM are used to train a machine learning method with explanation capacity such as decision tree to generate rules [4]. But there is lack of theoretical explanations to guarantee that the extracted rules can achieve good generalization performance. Another method uses the linear programming to optimize the vertexes of the hypercube based on the linear SVM [5]. This method is not suitable for the nonlinear situation while the nonlinear mapping and the kernel trick is one of the important characteristics of SVM.

In this paper, the feature selection and rule extraction are united in a scheme. This idea comes from the study of multiple kernel learning [6]. The single feature kernel is used as the basic kernel in this paper and then the feature selection problem is transformed into an ordinary multiple parameter learning problem [7]. The optimizations of these parameters (feature coefficients) are carried out by a 1-norm based linear programming. Every "support vector" is just the vertex of a hypercube. So a tree-like algorithm is proposed to adaptively extract the if-then rules. This paper is organized as follows: section 2 describes the MK-SVM based feature selection and rule extraction approach. Section 3 presents the experimental results on some UCI datasets.

2 Feature Selection and Rule Extraction from MK-SVM

Given a set of data points $G = \left\{\left(\vec{x}_i, y_i\right)\right\}_{i=1}^{n}$, $\vec{x}_i \in R^m$ and $y_i \in \{+1, -1\}$. The optimal separating hyperplane is found by solving the following regularized optimization problem which is identical with SVM:

$$\min J(\vec{w}, \vec{\xi}) = \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{n}\xi_i \tag{1}$$

$$\text{s.t.} \begin{cases} y_i\left(\vec{w}^T\phi(\vec{x}_i) + b\right) \geq 1 - \xi_i, i = 1, \cdots, n \\ \xi_i \geq 0 \end{cases} \tag{2}$$

where c is a regularization parameter and $\phi(x)$ is the nonlinear mapping.

By introducing the Lagrange function and differentiating with respect to $\vec{w}$ and ξ_i, the following dual programming is gotten:

$$\max\left\{\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j k(\vec{x}_i,\vec{x}_j)\right\} \tag{3}$$

$$s.t.\begin{cases}\sum_{i=1}^{n} y_i\alpha_i=0\\ 0\le\alpha_i\le C, i=1,\cdots,n\end{cases} \tag{4}$$

where $k(\vec{x}_i,\vec{x}_j)=\phi(\vec{x}_i)\circ\phi(\vec{x}_j)$ is called the kernel function.

If each basic kernel uses a single feature, the kernel function can be described as:

$$k(\vec{x}_i,\vec{x}_j)=\sum_{d=1}^{m}\beta_d k(x_{i,d},x_{j,d}) \tag{5}$$

where $x_{i,d}$ denotes the d^{th} component of the input vector $\vec{x}_i$. In equation (5), the parameter β_d represents the weight of each single feature kernel. So it is called feature coefficient. Then the feature selection problem is transformed into finding sparse feature coefficients $\vec{\beta}\subset R^m$.

When the kernel described in equation (5) is used, the optimization problem changes into:

$$\max\left\{\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j\sum_{d=1}^{m}\beta_d k(x_{i,d},x_{j,d})\right\} \tag{6}$$

$$s.t.\begin{cases}\sum_{i=1}^{n} y_i\alpha_i=0\\ 0\le\alpha_i\le C, i=1,\cdots,n\\ \beta_d\ge 0, d=1,\cdots,m\end{cases} \tag{7}$$

A two stage iterative procedure is used in this paper. The feature coefficient β_d is fixed and the Lagrange coefficients α_j can be gotten by solving the quadratic programming described in (6) and (7). The optimization of the feature coefficients β_d can be seen as a multiple parameter learning problem. They can be obtained by minimizing some estimates of the generalization errors of SVM [8]. A 1-norm soft margin error function is minimized to obtain the sparse solution:

$$\min J(\vec{\beta},\vec{\xi})=\sum_{d=1}^{m}\beta_d+\lambda\sum_{i=1}^{n}\xi_i \tag{8}$$

$$\text{s.t.}\begin{cases} y_i\left(\sum_{d=1}^{m}\beta_d\sum_{j=1}^{n}\alpha_j y_j k(x_{i,d},x_{j,d})+b\right)\geq 1-\xi_i \\ \xi_i\geq 0, i=1,\cdots,n \\ \beta_d\geq 0, d=1,\cdots,m \end{cases} \tag{9}$$

In equation (8), the regularized parameter λ controls the sparsity of the feature coefficients.

The dual of this linear programming is:

$$\max\sum_{i=1}^{n}u_i \tag{10}$$

$$\text{s.t.}\begin{cases} \sum_{i=1}^{n}u_i y_i(\sum_{j=1}^{n}\alpha_j y_j k(x_{i,d},x_{j,d}))\leq 1, d=1,\cdots,m \\ \sum_{i=1}^{n}u_i y_i=0 \\ 0\leq u_i\leq\lambda, i=1,\cdots,n \end{cases} \tag{11}$$

Algorithm 1

The procedures for feature selection using MK-SVM can be summarized as the following steps:

1. Initialization: set the regularization parameter γ, λ and the kernel parameter to some initial values. The feature coefficients β_d are set to $\left\{\beta_d^{(0)}=1 \mid d=1,\cdots,m\right\}$.
2. Solving the Lagrange coefficients $\alpha_j^{(t)}$: the Lagrange coefficients $\alpha_j^{(t)}$ are obtained by solving the quadratic programming described by equation (6) and (7) in which the feature coefficient $\beta_d^{(t-1)}$ is used.
3. Solving the feature coefficients $\beta_d^{(t)}$: the feature coefficients $\beta_d^{(t)}$ are obtained by solving a linear programming based on the Lagrange coefficient $\alpha_j^{(t)}$ solved in the last step. The dual problems of this linear programming are given in equation (10) and (11).
4. Calculating errors: the errors on the testing set are calculated according to the coefficients $\alpha_j^{(t)}$ and $\beta_d^{(t)}$ solved in above two steps. If the results are not convergent

according to the defined stopping criteria, go back to step 2 and implement the two-stage iteration optimization again. Go back to step 1 and tune the parameters until the output are optimal or satisfactory.

5. Output: the sparse feature coefficients $\vec{\beta}$, support vectors and the classification results. The features corresponding to the non-zero feature coefficients β_d are the selected features.

Definition 1: An optimal rule can be defined as that covers the hypercube with axis-parallel faces and has one vertex on the hyperplane.

Then a rule defined as above has the following formulation:

$$R = \left\{ \vec{x} \,\middle|\, \vec{l} \le \vec{x} \le \vec{u} \right\} \tag{12}$$

In equation (12), one of the lower bound ($\vec{l}$) and upper bound ($\vec{u}$) of $\vec{x}$ is the vertex lying on the hyperplane.

Proposition 1: For MK-SVM with linear kernel, each support vector is one vertex of a rule and the other vertex of this rule is one of the corners of the whole region.

Proof: According to the Kuhn-Tucker condition, the separating hyperplane of MK-SVM with linear kernel can be shown as:

$$\sum_{d=1}^{m} \beta_d \left(\sum_{j=1}^{n} \alpha_j y_j k(x_{j,d}, x_{i,d}) \right) + b = \pm 1, i \in SV \tag{13}$$

According to the definition 1, each support vector (SV) lies on the hyperplane and then it is one vertex of a rule. The rule with biggest volume is obtained when the other vertex of this rule is one of the corners of the whole region.

Proposition 2: For MK-SVM with nonlinear kernel, each support vector is one vertex of a rule and the other vertex of this rule is one of the corners of the region or another vector lying on the hyperplane.

Proof: According to the Kuhn-Tucker condition, the separating hyperplane of MK-SVM with nonlinear kernel can be viewed as the weighted sum of the nonlinear functions:

$$\sum_{d=1}^{m} \beta_d z_{i,d} + b = \pm 1, i \in SV \tag{14}$$

where $z_{i,d} = \sum_{j=1}^{n} \alpha_j y_j k(x_{j,d}, x_{i,d})$ is a nonlinear function.

According to the proposition 1, $\vec{z}_i$ is one vertex of a hypercube. So the bounds of the rule are the solutions of the following nonlinear equations:

$$R_{i,d} = \{x_{j,d} \mid \sum_{j=1}^{n} \alpha_j y_j k(x_{i,d}, x_{j,d}) \le z_{i,d}, i \in SV, 1 \le d \le M\} \tag{15}$$

or

$$R_{i,d} = \{x_{i,d} \mid \sum_{j=1}^{n} \alpha_j y_j k(x_{i,d}, x_{j,d}) \ge z_{i,d}, i \in SV, 1 \le d \le M\} \tag{16}$$

It is clear that each component of the support vectors ($x_{i,d}$) is one of the solutions of equations (15) and (16). According to the definition 1, every support vector $\vec{x}_i$ is one vertex of a hypercube. Considering the two situations about the interval solution of above univariate nonlinear equations, the other vertex of the hypercube may be the corner of the region or another vector lying on the hyperplane.

The following measures are defined to evaluate the quality of the extracted rules: classification accuracy (hit rate) and point coverage rate (number of samples covered by a rule and correctly classified divided by the total number of samples in that class).

Algorithm 2

1. Implement the algorithm 1 to get the support vectors with the selected features.
2. Derive the rules from support vectors according to proposition 1 and 2.
3. Calculate the evaluation measure (classification accuracy multiplied by point coverage rate) of every rule in the given region and keep the best rule.
4. Discard the region the extracted rules covered and get a new given region.
5. Go to step 3 to extract a new rule in the new region.
6. Stop.

3 Experiments

The performance of this method is measured on three widely used datasets: The breast cancer dataset, the heart disease dataset and the PIMA dataset.

Considering the different misclassification cost, we use the following three measures: average overall hit rate, sensitivity (number of negative samples correctly classified divided by the total number of negative samples) and specificity (number of positive samples correctly classified divided by the total number of positive samples) to evaluate the classification accuracy. In this experiment, the Gaussian kernel is used.

The selected features and three measures: average overall hit rate, sensitivity and specificity using MK-SVM are shown in table 1. Table 2 shows the experimental results of SVM using all of the features. From these two tables, it is seen that MK-SVM outperforms SVM in most of those measures. And the selected features are used to extract rules in follows.

Table 1. Experimental results of MK-SVM with selected features

Measures	Breast cancer	Heart disease	PIMA
Selected features	1,3,6	8,12,13	2,8
Overall hit rate %	97.51	87.40	77.29
Sensitivity %	98.04	87.91	59.59
Specificity %	97.26	86.82	87.40

Table 2. Experimental results of SVM with original features

Measures	Breast cancer	Heart disease	PIMA
Overall hit rate %	97.30	79.90	72.65
Sensitivity %	96.35	77.48	34.76
Specificity %	99.35	81.79	89.76

The extracted rules for MK-SVM on the breast cancer dataset are shown in table 3. Table 3 also shows the hit rate and point coverage rate for each rule and the average ones for each class. For MK-SVM, only two rules for the negative class and one rule for the positive class are extracted. For all these two classes, the average hit rate and point coverage rate are all superior to 90%. They are promising results.

Table 3. Rule extraction from the breast cancer dataset using MK-SVM

Number	Class	Rule body	Hit rate %	Coverage rate %
1	Pos.	$x_3 > 3$	93.79	89.54
2	Pos.	$x_1 > 5$ and $3 \geq x_3 \geq 2$	66.67	7.84
Overall	Pos.		92.85	97.38
3	Neg.	$x_3 < 4$	94.59	95.44
4	Neg.	$x_3 = 4$ and $x_6 \leq 5$	56.25	2.74
Overall	Neg.		92.86	98.18

Table 4 shows the number of extracted rules and coverage rate on the breast cancer dataset using MK-SVM, in compared with three neural network based rule extraction methods (see details in [8]). It is seen that MK-SVM extracts fewer rules and achieves higher coverage rate at most of cases.

Table 4. Rule extraction from the breast cancer dataset using MK-SVM and some other methods

Method	Class	Number of rules	Coverage rate %	Class	Number of rules	Coverage rate %
MK-SVM	Pos.	2	97.38	Neg.	2	98.18
NN1	Pos.	5	99.16	Neg.	6	95.27
NN2	Pos.	4	97.07	Neg.	5	96.17
NN3	Pos.	2	97.07	Neg.	3	95.72

The hit rate and point coverage rate on the heart disease and PIMA dataset are shown in table 5. The extracted rules are omitted here. It is seen that a small number of rules are extracted on each dataset. The coverage rate of the negative class can be improved by adding new rules, but it will result in a lower hit rate.

Table 5. Rule extraction from the heart disease and PIMA dataset using MK-SVM

Dataset	Class	Number of rules	Hit rate %	Coverage rate %
heart disease dataset	Pos.	3	82.35	90.32
heart disease dataset	Neg.	3	90.90	63.64
PIMA dataset	Pos.	3	82.18	81.10
PIMA dataset	Neg.	4	74.29	61.50

4 Conclusions

We propose a novel feature selection and rule extraction method based on multiple kernel support vector machine (MK-SVM). The most outstanding advantage of this method is that the rule is obtained in an easy way: only the support vectors necessary. Secondly, most rule extraction methods ignore the feature selection or leave it prior to the main task. This paper proposes a united system to carry out the rule extraction and feature selection simultaneously. In the experiments, the extracted rules with few selected features achieve good performance.

References

1. Liu Y., Zheng Y.F.: FS-SFS: a novel feature selection method for support vector machines. IEEE International Conference on Acoustics, Speech, Signal Processing, 5(2004) 797-780
2. Mao K.Z.: Feature subset selection for support vector machines though discriminate function pruning analysis. IEEE Transactions on SMC, part B, 34(2004) 60-67
3. Huang C.L., Wei C.J.: GA-based feature selection and parameters optimization for support vector machines. Expert Systems with applications, 31(2006) 231-240
4. He J., Hu H.J., Harrison R., et al: Rule generation for protein secondary structure prediction with support vector machines and decision tree. IEEE Transactions on nanobioscience, 5(2006) 46-53
5. Fung G., Sandilya S., Baharat R.: Rule extraction from linear support vector machines. 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2005) 32-40
6. Micchelli C.A., Pontil M.: Learning the kernel function via regularization. Journal of Machine Learning Research, 6(2005) 1099-1125
7. Chapelle O., Vapnik V., Bousquet O., Mukherfee S.: Choosing multiple parameters for support vector machines**.** Machine Learning, 46(2002) 131-159
8. Taha I.A., Ghosh J.: Symbolic interpretation of artificial neural networks. IEEE Transactions on knowledge and data engineering, 11(1999) 448-463

Combining Supervised and Semi-supervised Classifier for Personalized Spam Filtering

Victor Cheng and Chun-hung Li

Department of Computer Science, Hong Kong Baptist University, Hong Kong
{victor,chli}@comp.hkbu.edu.hk

Abstract. This paper addresses the problem of spam filtering for individual email user under the condition that only public domain labeled emails given as the training data and all emails from the user's email inbox are unlabeled. Owing to the difference of wordings and distribution of emails, conventional supervised classifier such as SVM cannot produce accurate result because it assumes the training and the testing data come from the same source and have the same distribution. We model these discrepancies as variation of decision hyperplane and come up with a criterion for selecting reliable emails with classified labels which are likely to be agreed by the user. A semi-supervised classifier then uses these emails as the training set and propagates the label information to other unlabeled emails by exploiting the distribution of them in feature space. Experimental result shows that this combined classifier strategy can classify emails for individual user with high accuracy.

1 Introduction

For most email users, email filtering seems to be an effective way to block spam. Traditional spam filters use rule-based techniques that discriminate spam from normal emails. This approach uses a combination of spammers' email addresses, IP addresses, header information, keywords of the subject line, and even the keywords in email contents to formulate the rules that identify spam emails. Machine learning, e.g. Bayesian learning [1], is another common approach to filter spam. Instead of specifying a set of rules explicitly, this approach uses a set of classified documents, including both spam and normal emails, to learn the rules implicitly. Sometime, domain specific properties, such as the presence of "!!!!" or "be over 21" are cooperated to make the filter more accurate. Recently, support vector machine (SVM) [4,5] is also a very popular technique in filtering emails. It works well even under high dimensional input space and sparse attributes. Combination of both approaches is now becoming popular, e.g. SpamAssassin [2].

With modern machine learning techniques, it is not difficult to achieve high accuracy in spam filtering. For example, Tretyakov [3] reported their simple SVM and multi-layer perceptron [6] had false positive and false negative values below 5%. These good results however require a working condition that the training data and the testing data are drawn randomly from the same source as most learning paradigms assume that all the data is drawn under the iid

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 449–456, 2007.

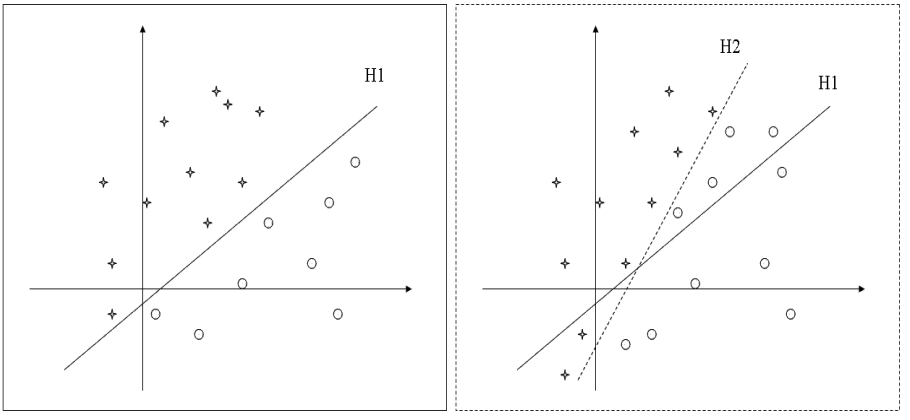


Fig. 1a Fig. 1b

Fig. 1. Fig.1a shows that the hyperplane H1 separating all the training patterns correctly. In Fig.1b H1 performs badly if the distribution of the testing patterns is different and H2 should be the right hyperplane.

(independent and identical distribution) condition. Fig. 1a shows the hyperplane H1 separating all the training patterns correctly can perform badly (see Fig. 1b) if the testing patterns have different distribution, in which H2 should be the correct separating hyperplane. To train a personalized spam filter for a user, both labeled spam and legitimate emails, getting from the same user's email inbox, are required. However, due to the privacy reason, it is difficult to get emails from the user. Even the user provides his emails, labeling them manually is a very costly and time consuming task. Without labeled emails from the user, the performance of a spam filter will be downgraded drastically. A spam filter test running on a dataset [7] shows that the accuracy drops from over 90% to about 78% when training are based on public domain email dataset only and no user's labeled emails are provided. The is due to the discrepancy between the word distributions of emails from public domain and that of the users' inbox.

In this paper, a combined SVM and semi-supervised classifier is proposed to label a user's emails. Firstly a SVM is trained with labeled public domain emails and it is used to classify a user's emails. The discrepancy between them and user's emails is modeled as variation of decision hyperplane and "reliable labeled emails" with classified labels which are likely to be agreed by the user are selected. A semi-supervised classifier [8] then uses these emails as the training set and propagates the label information to other unlabeled emails by exploiting the distribution of them in feature space. Throughout this paper, ranking of emails by classifier output values is used to indicate their likelihood to be spam, rather than using simple binary labeling, e.g. {-1,1}, unless specified otherwise. An email with higher classifier output value is ranked higher and it is more probable to be spam. The classification is evaluated with Area under the ROC

curve (AUC) metric [9]. In this case it can be regarded as the Wilcoxon-Mann-Whitney (WMW) statistic [10] given as:

$$W = \sum_{i=1}^{p} \sum_{j=1}^{n} I(x_i, y_j)/pn$$

$$I(x_i, y_j) = \begin{cases} 1 & \text{if} f(x_i) > f(y_j) \\ 0 & \text{otherwise} \end{cases},$$

where p is the number of spam emails and n is the number of legitimate emails, x_i and y_j represent a spam and a legitimate email respectively. The function f is a classifier which assigns a score to x_i and y_j for ranking. A view of the AUC value with this setting is the probability of spam email having higher ranking than a legitimate email.

The remainder of this paper is organized as follows. Section 2 discusses the classification of user's emails using SVM. Modeling of different users' email distributions and the criterion for selecting "reliable labeled emails" is also proposed. Section 3 presents the using of the semi-supervised learning algorithm to propagates the label information to unlabeled emails. Section 4 summarizes the testing results. Finally, a conclusion is given in Section 5.

2 Classification of Emails Using SVM

Support Vector Machine (SVM) [4,5] is a very popular classifying tool in recent years. SVM employs kernel function to map the input data into some much higher dimensional feature space implicitly in which data becomes linearly separable. The linear decision boundary is drawn in a manner that the margin, minimum distance between training examples and the boundary, is maximized. In case that the mapped data points are non-linearly separable, a cost is included to account for the wrongly classified examples and the margin is maximized while the cost is minimized. For spam filtering, linear kernels are found to have good performance and thus they are used in our study.

If we form a SVM with labeled public domain emails as the training dataset and classify a person's email, the classifier may not give accurate result because a considerable number of emails are classified wrongly due to the distribution difference as described in Section 1. Fig. 2 gives another example on the situation in a highly simplified two-dimensional plane. Referring to Fig. 2, points far away from the decision line $H1$ is less likely to be affected when the decision line is changed from $H1$ to $H2$. In fact, under certain conditions, the class label of these points is preserved. Consider a SVM classification in N dimensional feature space with decision hyperplane $H1$ which can be represented by

$$H1 : W'X + b = 0 \tag{1}$$

where W is the normal vector of the hyperplane and W' denotes its transpose. The distance d_1 between a point $X \in R^N$ (assume data normalized in feature space, i.e. $\| X \| = 1$) and the hyperplane is

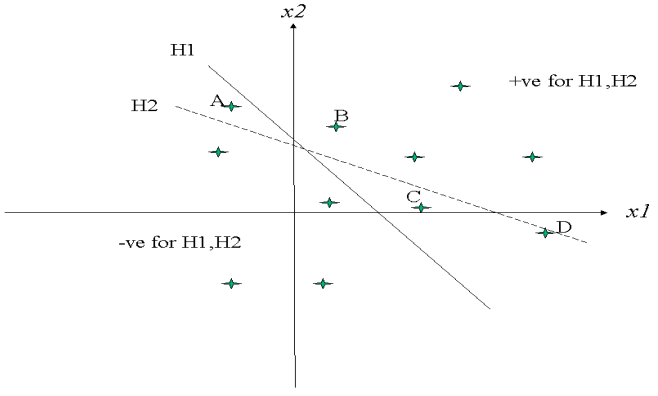


Fig. 2. Difference in wording causes different decision lines leading to different classifications. $H1$ and $H2$ are two different decision lines. Point C,D are classified as +ve data and point A is classified as -ve data if $H1$ is used. However, C,D are classified as -ve data and A as +ve data if $H2$ is used.

$$d_1 = \frac{\mid W'X + b \mid}{\parallel W \parallel} \quad . \tag{2}$$

If the decision hyperplane of the SVM is changed to $H2$,

$$H2 : (W + \delta W)'X + b + \delta b = 0 \ , \tag{3}$$

the new distance is

$$d_2 = \frac{\mid (W + \delta W)'X + b + \delta b \mid}{\parallel W + \delta W \parallel} \quad . \tag{4}$$

Without loss of generality, we assume $W'X + b > 0$, then

$$d_2 = \frac{\mid d_1 \parallel W \parallel + \delta W'X + \delta b \mid}{\parallel W + \delta W \parallel} \quad . \tag{5}$$

Label of X will not be changed if

$$d_1 > \frac{\mid \delta W'X + \delta b \mid}{\parallel W \parallel} \tag{6}$$

or, under the more strict condition,

$$d_1 > \frac{\parallel \delta W \parallel + \mid \delta b \mid}{\parallel W \parallel}. \quad \because \parallel X \parallel = 1, \quad \delta W'X \leq \parallel \delta W \parallel \tag{7}$$

Thus, class label of data points having the distance greater than $(\parallel \delta W \parallel + \mid \delta b \mid)/ \parallel W \parallel$ to H1 will be preserved even the decision hyperplane is switched to

H2. These data points are called "reliable" data points in the sequel. Practically, it is hard to identify reliable data points or even their existence because $\| \delta W \|$ and $| \delta b |$ are not known. The above formulation, however, points out class labels of data points are likely to be preserved if they are far from the original decision hyperplane.

As a result, if a person's emails having distribution not too different from the counterparts of the public domain, these discrepancies can be modeled as changing of decision hyperplane. Then a SVM can be trained with public domain emails and classify the person's emails. Classified personal emails far from the decision hyperplane can be selected as training data for other classifiers or next stage classification because they are likely to be reliable data points.

3 Semi-supervised Learning for Spam Filtering

Let $\{(x_1, y_1) \dots (x_l, y_l)\}$ be the labeled emails, with $x_i \in R^N$, $y \in \{-1, 1\}$, and $\{x_{l+1} \dots x_{l+u}\}$ the unlabeled emails. The problem is to label or assign a probability to the unlabeled emails such that a cost function is minimized. Let w_{ij} represents the similarity between x_i and x_j. Then a graph where nodes represent emails, $x_i, i \in \{1, 2, \dots, l+u\}$, and edges represent similarity w_{ij} between x_i and x_j can be created. Similarity is often evaluated with radial basis function

$$w_{ij} = exp\left(-\frac{\| x_i - x_j \|^2}{\sigma^2}\right) \tag{8}$$

or cosine similarity

$$w_{ij} = \frac{< x_i, x_j >}{\| x_i \| \| x_j \|}. \tag{9}$$

In this graph, the label of x_i can propagate through edges to another node x_j according to a transition probability

$$P_{ij} = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \tag{10}$$

and the transition of the whole graph can be represented by the $(l + u \times l + u)$ dimension matrix P. Define a label matrix Y with dimension $(l + u \times 2)$, whose $i'th$ row has two elements having values between 0.0 and 1.0. The first element indicates the probability that the $i'th$ email is a legitimate email and the second element indicates the probability that email is a spam, i.e. $y_{i,1} + y_{i,2} = 1.0$. Under this configuration, the class probability of unlabeled emails can be computed, by using the label propagation algorithm [8] given as follows.

1. Initialize the label matrix Y
   - If x_i is labeled spam, $y_{i,1} = 0, y_{i,2} = 1$.
   - If x_i is labeled legitimate, $y_{i,1} = 1, y_{i,2} = 0$.
   - If xi is unlabeled, randomize $y_{i,1}$, $y_{i,2}$ to a small values.
2. Update Y by Computing $Y_{n+1} = PY_n$.
3. Clamp the labels of labeled node to its original values.
4. Repeat 2, and 3 until Y_n converge.

This algorithm propagates the values of labeled nodes to class boundaries according to the distribution of the unlabeled emails. The convergence of the algorithm is guaranteed if the graph is connected. In addition, the convergence to trivial cases such as all $y_{i,1} = 0.0$ are avoided because there are both labeled spam and legitimate emails.

4 Testing Results

The testing dataset in [7] are used in evaluating the combined classifier strategy. In this dataset, there are 4,000 labeled emails coming from public domain and they are used as the training data. Three sets of unlabeled data, each containing 2,500 emails from three different users' email inboxes, are also provided for testing. Ground true of the emails are given for evaluation. It should be noted that the email distributions of public domain is different from that coming from individual users' inboxes. Direct usage of training data to train a classifier and to classify unlabeled emails will give unsatisfactory results. The goal is to rank the emails for each user such that spam emails should have higher ranking than legitimate emails. The correctness of ranking is measured with AUC value. It has maximum value 1.0 representing the perfect case that all spam emails are ranked higher than legitimate emails. In this dataset, there are also two additional sets of data, "E" and "F" but they are not used in the test. Table 1 summarizes the properties of the dataset.

Table 1. Data in the testing dataset

	Dataset	No. of emails (50% spam and 50% legitimate emails)	Labels +1:spam, -1:legitimate
"A"	Training Emails obtained from public domain	4000	Labeled
"B"	Emails from User 00	2500	Unlabeled
"C"	Emails from User 01	2500	Unlabeled
"D"	Emails from User 02	2500	Unlabeled
"E"	Tuning Emails obtained from public domain	4000	Labeled
"F"	Tuning Emails from User 00	2500	Labeled
"G"	Ground true labels of Emails "B", "C", and "D".	n.a.	n.a.

A SVM is first trained with Dataset "A" and then it is used to classify Dataset "B","C","D" with the distance of each email to the decision hyperplane is then evaluated. As described in Section 2, emails that far from the hyperplane are likely to be "reliable data point". A number of classified emails farthest from the

hyperplane are selected forming the training set of a semi-supervised classifier. Ranking of emails are obtained from the output of the classifier. Finally, AUC of the ranking is evaluated. The testing result is given in Table 2. Referring to this table, it is clear that the proposed approach performs much better than SVM or SVM1. Cases with different number of emails which are far from the hyperplane are tested and the result shows that the proposed approach is quite stable to the variation of this number. Finally, it is worth noting that in Table 2 the AUC value drops as the number of "reliable data point" is over 300. It is because the additional selected "reliable data point" is no longer "reliable". However, the AUC for User 2 is still very high because its email distribution is a bit similar to that of the public domain.

Table 2. AUC values for classifying emails of User 00, User 01 and User 02. Number of SVM classified emails (farthest to the decision hyperplane) for semi-supervised learning is given in parentheses.

	Algorithms	AUC values		
		User 00	User 01	User 02
SVM	Use SVM only (no preprocessing)	0.73	0.78	0.89
SVM1	Use SVM only (data preprocessed to 0/1 vector and normalized)	0.84	0.87	0.94
SVMSSL	SVM1 + Semi-supervised classifier (+ve / -ve "reliable" Sample)			
	(100 / 100)	0.913	0.947	0.983
	(200 / 200)	0.912	0.952	0.987
	(300 / 300)	0.907	0.957	0.989
	(600 / 600)	0.861	0.942	0.992

5 Conclusion

Spam or junk emails, are very annoying to email users and filtering is one of the ways to block spam. However, tuning a spam filter for individual user is costly and time consuming and sometimes impractical because of privacy reason. This paper proposes a combined supervised and semi-supervised classifier that helps the labeling/ranking of user's emails. As the distribution of public domain emails is different from that of emails of individual user, classical supervised classifiers such as SVM or naïve Bayes classifier do not gives satisfactory results. We model the discrepancy of distribution by variation of decision hyperplanes and come up with a criterion selecting some "reliable" SVM classified emails as training examples for next stage classification. A semi-supervised classifier using these examples together with exploiting user's email distribution to classify the unlabeled emails. Interestingly, this simple approach can classify user's emails

with a high accuracy, in AUC metric. Of course, the war between spammers and anti-spammers is not over. Some spammers are now using hyperlinks to divert people to their websites or even put their messages in image format so that they cannot be detected easily.

References

1. Sahami, M., Dumais, S., Heckerman, D., and Horvitz, E.: A Bayesian Approach to Filtering Junk E-mail. AAAI Workshop on Learning for Text Categorization, Madison, Wisconsin, July 1998.
2. The Apache SpamAssassin Project, http://spamassassin.apache.org/, accessed July 2006.
3. Tretyakov, K.: Machine Learning Techniques in Spam Filtering. Data Mining Problem-oriented Seminar, MTAT.03.177, May 2004, pp. 60-79.
4. Schölkopf, B.: Statistical Learning and Kernel Method. MSR-TR 2000-23, Microsoft Research (2000)
5. Drucker, H., Wu, D., and Vapnik, V.N.: Support Vector Machines for Spam Categorization. IEEE Trans. On Neural Networks, Vol. 10, No. 5, Sept. 1999.
6. Omidvar, O., Dayhoff, J.: Neural Networks and Pattern Recognition, Academic Press, 1998.
7. Discovery Challenge, ECMLPKDD2006, http://www.ecmlpkdd2006.org/challenge.html, accessed July 2006.
8. Zhu, X.:Semi-Supervised Learning with Graphs. Doctoral thesis, CMU-LTI-05-192, May 2005
9. Bradley, A.P.: The Use of the Area Under the ROC curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition, 30:1145-1159,1997.
10. Wilcoxon, F.: Individual Comparisons by Ranking Methods, Biometrics, 1:80-83, 1945.
11. Mann, H.B., and Whitney, D.R.: On a Test Whether One of Two Random Variables is Stochastically Larger than the Other. Annals of Mathematical Statistics, 18:50-60, 1947.
12. Mictchell, T.M.: Machine Learning. McGraw-Hill Companies, Inc., 1997.

Qualitative Simulation and Reasoning with Feature Reduction Based on Boundary Conditional Entropy of Knowledge

Yusheng Cheng[1,2], Yousheng Zhang[1], Xuegang Hu[1], and Xiaoyao Jiang[3]

[1] School of Computer Science, Hefei University of Technology, Hefei 230009, China
[2] School of Computer Science, Anqing Teachers College, Anqing 246011, China
[3] Computer Dept. of Nanjing Audit College, Nanjing 210029, China
chengyshaq@163.com,{zhangyos,xueghu}@mail.hf.ah.cn

Abstract. The present paper discusses a new definition of knowledge rough entropy based on boundary region from the aspect of Pawlak topology. This definition accurately reflects an idea that the uncertainty of set can be described by boundary region. It thus proves an important conclusion that boundary conditional entropy of knowledge monotonously reduces with the diminishing of information granularity. Combining qualitative reasoning technology with knowledge information entropy based on rough sets theory, a heuristic algorithm for feature reduction is proposed which can be used to eliminate the redundancy in the qualitative description and the qualitative differential equations are obtained. The result shows that the rough sets theory (RST) is of good reliability and prospect in qualitative reasoning and simulation.

Keywords: qualitative reasoning; qualitative simulation; RST; feature reduction; boundary conditional entropy.

1 Introduction

RST, as a new mathematical tool to deal with inexact, uncertain knowledge, has been successfully employed in machine learning, data mining and other fields since it was put forward by Pawlak[1]. It is established on the basis of classification mechanism, which takes classification according to equivalence relation [2]. On the other hand, RST believes that knowledge has granularity, the smaller, the more concepts precisely expressed. Meanwhile, Uncertainty and its measure have always been important issues in the study of RST[1,2,3]. Wierman[4] introduces the definition of granularity measure, connecting Shannon entropy[5] with uncertainty measure. Besides, Miao[6]discusses the relation between knowledge roughness and information entropy, proving the monotony of knowledge rough entropy; Wang[7,8] defines the equivalence of feature reduction from the aspect of informational view and algebraic view of RST and provides reduction algorithm of decision table based on conditional information entropy[8]. Liang [9] defines a new information entropy, which can be better used for measure rough set and rough classification.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 457–464, 2007.

In the above study, knowledge rough entropy is failing to show accurately the reason that causes conceptual uncertainty–the existence of boundary region [1-3]. The present paper defines a new knowledge rough entropy and conditional entropy based on boundary region and is an attempt to solve measure uncertainty from the angle of set topology (Pawlak topology[2,3]). It provides feature reduction algorithm of decision table based on boundary conditional entropy which will be used in qualitative reasoning and simulation[10]. Qualitative reasoning is to ignore the details instead of collecting specific values of the system's variables at different time to simulate the system's behavior. But this method has a relatively bigger knowledge redundancy. Thus, it is advisable to delete the problem of knowledge redundancy by using feature reduction method in RST. The qualitative simulation of spring physical system uses attribute significance as heuristic algorithm for feature reduction together with the technology of qualitative reasoning and simulation.

2 General Meaning of Conditional Entropy of Knowledge

An information system is usually denoted as a triplet $S = (U, C \cup D, f)$, which is called a decision table, where U is the universe which consists of a finite set of objects, C is the set of condition attributes and D the set of decision attributes. With every attribute $a \in C \cup D$, set of its values V_a is associated. Each attribute a determines an information function $f : U \to V_a$ such that for any $a \in C \cup D$, and $x \in U, f(x) \in V_a$. Each non-empty subset $B \subseteq C$ determines an indiscernible relation $R_B = \{(x, y) : \forall a \in B, f_a(x) = f_a(y), x, y \in U\}$. R_B is called an equivalence relation and partitions U into a family of a disjoint subsets. U/R_B called a quotient set of U: $U/R_B = \{[x]_B : x \in U\}$, where $[x]_B$ denotes the equivalence class determined by x with respect to B, i.e., $[x]_B = \{y \in U : (x, y) \in R_B\}$. $B \subseteq C$ is a subset of attributes, and $X \subseteq U$is a subset of discourse, the sets $\underline{B}(X) = \{x \in U : [x]_B \subseteq X\}, \overline{B}(X) = \{x \in U : [x]_B \cap X \neq \phi\}$ are called B-lower approximation and B-upper approximation respectively. Given a decision system $S = (U, C \cup D, f)$,partition of condition attributes $U/R_C = \{X_1, X_2 \cdots X_n\}$, $U/R_B = \{Y_1, Y_2 \cdots Y_k\}$ and partition of decision attributes $U/R_D = \{D_1, D_2 \cdots D_m\}$,the set $\underline{B}(D_1) \cup \underline{B}(D_2) \cup \cdots \cup \underline{B}(D_m)$ is called the B-positive region of classification induced by D and is denoted by $POS_B(D)$.The set $BN_B(D) = U - POS_B(D)$ is called the B-boundary of classification induced by D.

Definition 2.1 [6-9]. The information entropy of knowledge B is defined as follows,

$$H(B) = -\sum_{i=1}^{k} p(Y_i) \log_2 p(Y_i)$$

Definition 2.2 [8]. Conditional information entropy of knowledge C with respect to D is defined as follows,

$$H(D|C) = -\sum_{i=1}^{n} p(X_i) \sum_{j=1}^{m} p(D_j|X_i) \log_2 p(D_j|X_i)$$

Where $p(X_i) = \frac{|X_i|}{|U|}$, $p(D_j|X_i) = \frac{|X_i \cap D_j|}{|X_i|}$.

From Definition 2.2, we get $H(D|C) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p(X_i \cap D_j)[\log_2 p(D_j \cap X_i) - \log_2 p(X_i)]$, thus when $X_i \in POS_C(D)$, we have $\log_2 p(Y_j \cap X_i) - \log_2 p(X_i) = 0$, therefore, positive region of decision system has no effect on $H(D|C)$.

3 Conditional Entropy Based on Boundary Region

According to the definition of set topology [2], set uncertainty is mainly caused by the existence of boundary region. If it is empty, then the set is accurate; otherwise, it is rough [1,2,3]. Therefore, it is quite reasonable to describe knowledge uncertainty by boundary region.

Decision system $S = (U, C \cup D, f)$, $P, Q \subseteq C$, define partial order relation $\leq$: $P \leq Q \Leftrightarrow U/R_P \subseteq U/R_Q$, then P is more refined than Q (or: Q is rougher than P). If $P \leq Q$, and $P \neq Q$, then P is strictly more refined then Q (or: Q is strictly rougher than P), shown as $P \prec Q$.

Definition 3.1. Decision system $S = (U, C \cup D, f)$, $B \subseteq C$, partition of condition attributes B is $U/R_B = \{X_1, X_2 \cdots X_m\}$ and B's boundary region against knowledge D is $BN_B(D)$, the corresponding classification is $BN_B(D)/B = \{G_1, G_2, \cdots, G_t\}$, then B's boundary entropy against D and B's boundary conational entropy against D are defined as follows respectively:

$$E_{BN}(B) = \sum_{i=1}^{t} p(G_i) \log_2 |G_i|$$
$$E_{BN}(D|B) = -\sum_{i=1}^{t} p(G_i) \sum_{j=1}^{m} p(X_j|G_i) \log_2 p(X_j|G_i)$$

Proposition 3.1. $E_{BN}(B \cup D) = E_{BN}(D|B) - E_{BN}(B)$.

Proof. $E_{BN}(B \cup D) = -\sum_{i=1}^{t} \sum_{j=1}^{m} p(X_j \cap G_i) \log_2 p(X_j \cap G_i)$

$$E_{BN}(D|B) = -\sum_{i=1}^{t} p(G_i) \sum_{j=1}^{m} p(X_j|G_i) \log_2 p(X_j|G_i)$$
$$= -\sum_{i=1}^{t} \sum_{j=1}^{m} p(X_j \cap G_i)[\log_2 p(X_j \cap G_i) - \log_2 p(G_i)]$$
$$= -\sum_{i=1}^{t} \sum_{j=1}^{m} p(X_j \cap G_i) \log_2 p(X_j \cap G_i) + \sum_{i=1}^{t} \sum_{j=1}^{m} p(X_j \cap G_i) \log_2 p(G_i)$$

Additionally because $\sum_{j=1}^{m} p(X_j \cap G_i) = p(G_i)$.

So $E_{BN}(D|B) = E_{BN}(B \cup D) + \sum_{i=1}^{t} p(G_i)\log_2 p(G_i) = E_{BN}(B \cup D) + E_{BN}(B)$
That's $E_{BN}(B \cup D) = E_{BN}(D|B) - E_{BN}(B)$. (Finished)

Proposition 3.2. Decision system $S = (U, C \cup D, f)$, $A, B \subseteq C$. If $A \leq B$, then $E_{BN}(D|A) \leq E_{BN}(D|B)$.

Proof. Order $U/R_D = \{D_1, D_2 \cdots D_m\}$, $BN_A(D)/A = \{G_1, G_2, \cdots, G_t\}$. Because $A \leq B$, then $BN_A(D) \subseteq BN_B(D)$, so $BN_A(D)/A \leq BN_B(D)/B$. Suppose $BN_B(D)/B = \{G_1, G_2, \cdots, G_{p-1}, G_{p+1}, \cdots G_{q-1}, G_{q+1}, \cdots G_t, G_p \cup G_q\}$.

According to proposition 3.1:

$$E_{BN}(D|A) = -\sum_{i=1}^{m}\sum_{j=1}^{t} p(D_i \cap G_j)\log_2 p(D_i \cap G_j) + \sum_{j=1}^{t} p(G_j)\log_2 p(G_j),$$

$$E_{BN}(D|B) = E_{BN}(D|A)$$
$$- \sum_{i=1}^{m} p[(G_p \cup G_q) \cap D_i]\log_2 p[(G_p \cup G_q) \cap D_i] + p(G_p \cup G_q)\log_2 p(G_p \cup G_q)$$
$$+ \sum_{i=1}^{m} p(G_p \cap D_i)\log_2 p(G_p \cap D_i) - p(G_p)\log_2 p(G_p)$$
$$+ \sum_{i=1}^{m} p(G_q \cap D_i)\log_2 p(G_q \cap D_i) - p(G_q)\log_2 p(G_q)$$

so

$$\Delta E = E_{BN}(D|B) - E_{BN}(D|A)$$
$$= - \sum_{i=1}^{m} p[(G_p \cup G_q) \cap D_i]\log_2 p[(G_p \cup G_q) \cap D_i] + p(G_p \cup G_q)\log_2 p(G_p \cup G_q)$$
$$+ \sum_{i=1}^{m} p(G_p \cap D_i)\log_2 p(G_p \cap D_i) - p(G_p)\log_2 p(G_p)$$
$$+ \sum_{i=1}^{m} p(G_q \cap D_i)\log_2 p(G_q \cap D_i) - p(G_q)\log_2 p(G_q)$$

Additionally because $\sum_{i=1}^{m} p(D_i \cap G_p) = p(G_p), \sum_{i=1}^{m} p(D_i \cap G_q) = p(G_q)$

Thus

$$\Delta E = - \sum_{i=1}^{m} p[(G_p \cup G_q) \cap D_i]\log_2 p[(G_p \cup G_q) \cap D_i]$$
$$+ \sum_{i=1}^{m} p[(G_p \cup G_q) \cap D_i]\log_2 p(G_p \cup G_q)$$
$$+ \sum_{i=1}^{m} p(G_p \cap D_i)\log_2 p(G_p \cap D_i) - \sum_{i=1}^{m} p(G_p \cap D_i)\log_2 p(G_p)$$
$$+ \sum_{i=1}^{m} p(G_q \cap D_i)\log_2 p(G_q \cap D_i) - \sum_{i=1}^{m} p(G_q \cap D_i)\log_2 p(G_q)$$
$$= \sum_{i=1}^{m} p(G_p \cap D_i)\{\log_2 p(G_p \cap D_i) + \log_2 p(G_p \cup G_q) - \log_2 p(G_p)$$
$$- \log_2 p[(G_p \cup G_q) \cap D_i]\} + \sum_{i=1}^{m} p(G_q \cap D_i)\{\log_2 p(G_q \cap D_i)$$
$$+ \log_2 p(G_p \cup G_q) - \log_2 p(G_q) - \log_2 p[(G_p \cup G_q) \cap D_i]\}$$

$$= \frac{1}{|U|}\sum_{i=1}^{m}\{|G_p \cap D_i| \log_2 \frac{|G_p \cap D_i||G_p \cup G_q|}{|G_p|(|G_p \cap D_i|+|G_q \cap D_i|)} + |G_q \cap D_i| \log_2 \frac{|G_q \cap D_i||G_p \cup G_q|}{|G_q|(|G_p \cap D_i|+|G_q \cap D_i|)}\}$$

Order $|G_p| = x$, $|G_q| = y$, $|G_p \cap D_i| = ax$, $|G_q \cap D_i| = by$, obviously get $x > 0, y > 0, 0 \leq a \leq 1, 0 \leq b \leq 1$, then

$$\Delta E = \frac{1}{|U|}\sum_{i=1}^{m}\{ax \log_2 \frac{ax+ay}{ax+by} + by \log_2 \frac{bx+by}{ax+by}\} = \frac{1}{|U|}\sum_{i=1}^{m} f_i$$

If $a \times b = 0$, get $f_i \geq 0$.
$0 < a \leq 1, 0 < b \leq 1$ shall be only considered in the following:
Order $ax = \lambda$, $by = \beta$, $\frac{a}{b} = \theta$, obviously get $\lambda > 0, \beta > 0, \theta > 0$ and

$$f_i = \lambda \log_2 \frac{\lambda + \theta\beta}{\lambda + \beta} + \beta \log_2 \frac{\beta + \theta^{-1}\lambda}{\lambda + \beta}$$

then

$$\frac{d(f_i)}{d(\theta)} = \frac{\lambda\beta(\theta - 1)}{\theta(\lambda + \theta\beta)}$$

so, $\frac{d(f_i)}{d(\theta)} < 0, 0 < \theta < 1$; $\frac{d(f_i)}{d(\theta)} = 0, \theta = 1$; $\frac{d(f_i)}{d(\theta)} > 0, \theta > 1$.
when $\theta = \frac{a}{b} = 1$, function f_i gets the minimal $f_i|_{\theta=1} = 0$.

The above shows, $\Delta E \geq 0$, then $E_{BN}(D|A) \leq E_{BN}(D|B)$ is proved. The proposition shows that boundary conditional entropy of knowledge monotonously reduces with the diminishing of information granularity.

4 Qualitative Simulation and Reasoning with Feature Reduction Based on Boundary Conditional Entropy

It shows that boundary conditional entropy of knowledge decreases with the information granularity. It proposes a greedy algorithm for feature reduction, based on conditional entropy reduction associated to boundary region related features. It then applies the algorithm in the field of qualitative description of systems simulated by qualitative differential equations in order to diminish redundancy.

4.1 Heuristic Algorithm for Feature Reduction

The significance of attribute is defined as follows:

Definition 4.1. Decision system $S = (U, C \cup D, f)$, $B \subseteq C$, the significance of b in B with respect to D is defined as follows,

$$Sig_{B\backslash\{b\}}(D|\{b\}) = E_{BN}(D|B\backslash\{b\}) - E_{BN}(D|B)$$

We know such important conclusions as information entropy is monotonously reducing with the diminishing of information granularity. Because B is more

refined than $B\backslash\{b\}$, therefore, $Sig_{B\backslash\{b\}}(D|\{b\}) \geq 0$. Definition 4.1 shows that b is important in B can be measured based on the increment of boundary conditional entropy. Especially, when system $S = (U, C \cup D, f)$ is not a decision system, i.e., $D = \phi$, then we can consider $E_{BN}(\phi|B) = H(B)$.

Proposition 4.1. b is necessary in B when $Sig_{B\backslash\{b\}}(D|\{b\}) > 0$.

Algorithm KIEBAFR (**K**nowledge **I**nformation **E**ntropy-**B**ased **A**lgorithm for **F**eature **R**eduction)
Input: decision system $S = (U, C \cup D, f)$
Output: a feature reduction Red of decision system $S = (U, C \cup D, f)$
Step1. calculate the boundary conditional entropy $E_{BN}(D|C)$
Step2. for any $c \in C$, calculate the significance of c in C:$Sig_{C\backslash\{c\}}(\{c\})$ and then obtain Re$d = \{c|Sig_{C\backslash\{c\}}(\{c\}) > 0\}$
Step3. Repeat:
Step3.1 calculate boundary conditional entropy $E_{BN}(D|\text{Re}d)$,if $E_{BN}(D|C) = E_{BN}(D|\text{Re}d)$ output a reduction set Red and Stop; Otherwise, continue Step3.2
Step3.2 for each attribute $a \in C\backslash\text{Re}d$,calculate $Sig_{\text{Re}d}(\{a\})$,select attribute a_0 to make $Sig_{\text{Re}d}(\{a\})$ the maximal and compute Red = Re$d \cup \{a_0\}$, goto step3.1
The time complexity of the algorithm is $O(|C|^3|U|^2)$.
For example as shown in table 1 which has a reduction set $\{a, e\}$ by CEBARKNC or CEBARKCC [8]:

Table 1. A decision system

U	a	b	c	e	d
1	1	0	1	1	0
2	0	1	0	1	1
3	0	0	0	0	0
4	0	0	0	1	1
5	0	0	0	1	1
6	0	0	0	1	1
7	0	0	1	1	1
8	0	0	1	0	0
9	0	0	1	1	1

The decision classes of objects are: $D_1 = \{1, 3, 8\}, D_2 = \{2, 4, 5, 6, 7, 9\}$,
The condition classes of objects are: $X_1 = \{1\}, X_2 = \{2\}, X_3 = \{3\}, X_4 = \{4, 5, 6\}, X_5 = \{7, 9\}, X_6 = \{8\}$. Because $BN_C(D)/C = \phi$, then $E_{BN}(D|C) = 0$
Next, we give how to calculate the significance of e in attributes C: because $BN_{C\backslash\{e\}}(D)/C\backslash\{e\} = \{\{3, 4, 5, 6\}, \{7, 8, 9\}\}$, therefore,

$$Sig_{C\backslash\{e\}}(D|\{e\}) = -\{\frac{4}{9}(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}) + \frac{3}{9}(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3})\} = 0.198$$

Similarly, we can get
$Sig_{C\backslash\{c\}}(D|\{c\}) = 0$; $Sig_{C\backslash\{b\}}(D|\{b\}) = 0$; $Sig_{C\backslash\{a\}}(D|\{a\}) = 0.09$. So Re$d$ = $\{a, e\}$, Compute $E_{BN}(D|\text{Re}d) = 0$, because $E_{BN}(D|C) = E_{BN}(D|\text{Re}d)$, output a result: $\{a, e\}$, Stop.

4.2 Application of KIEBAFR in Qualitative Simulation and Reasoning

Take the qualitative simulation of spring physical system for example[10], as shown in Fig 1. The following four variables can be used to describe: (1) x,means the position of the object; (2) v, means velocity of the object: v=dx/dt; (3) a, means acceleration of the object: a=dv/dt; (4) f, means strength by pulling object. The qualitative analysis obtains knowledge expression system for qualitative description of spring physical system shown as Table 2[10], i.e., $S = (U, C = \{[x], [f], [a], [v]\})$.

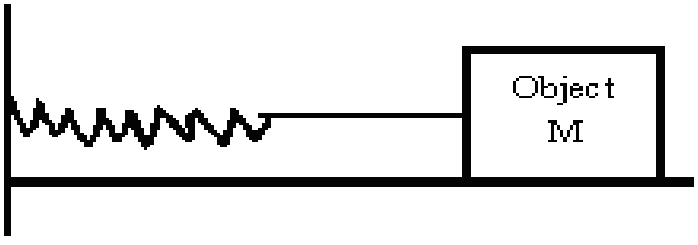


Fig. 1. Spring Physical System

Table 2. A Qualitative Descriptive Knowledge System

U	$[x]$	$[f]$	$[a]$	$[v]$
$s1$	+	−	−	+
$s2$	+	−	−	0
$s3$	+	−	−	−
$s4$	0	0	0	+
$s5$	0	0	0	0
$s6$	0	0	0	−
$s7$	−	+	+	+
$s8$	−	+	+	0
$s9$	−	+	+	−

The qualitative differential equations of spring physical system can be obtained by KIEBAR algorithm. i.e., $[f] = [a]$, $[f] = [x]$, $[x] = [a]$.

The explanation is as follows: in qualitative expression information system, $\{[v], [x]\}$ is the reduction of original qualitative expression system (Figure 1), which shows it makes no difference to the classification ability of the original knowledge expression system whether to delete [a]'s attribute or [f]'s attribute, so [a] and [f] have consistent effect on information system, marked as [f]=[a], and the first qualitative differential equation is obtained. So [f]=[x], [a]=[x].The result is in accordance with that of the qualitative differential equation after the qualitative calculation of $f = ma$ (m is the mass of the object)and $f = -kx$(k is the modulus of spring flexibility)[10].

5 Conclusion

The existence of boundary region is the major cause of set uncertainty. The information entropy and rough set entropy in general meaning can't explain it clearly. Based on this, the present paper puts forward the definition of knowledge boundary rough entropy and boundary conditional entropy, and describes some algebraic view in RST by using the method of boundary conditional entropy, establishes the connection with algebraic view of RST. These important conclusions also guarantee feature reduction algorithm based on boundary conditional entropy. Qualitative simulation of spring physical system shows that RST is a powerful method in data mining and of good reliability and prospect in qualitative reasoning and qualitative simulation.

Acknowledgements. This work is a part of the project "Pre-warning Fault and Intelligence Diagnosis Based on Rule and Exception Model (070412061)" which is supported by the Natural Science Foundation of Anhui Province.

References

1. Pawlak Z.: *Rough sets: theoretical aspects of reasoning about data*. Kluwer Academic Publishers, Boston (1991)
2. Zhang W.X., Wu W.Z., et al.: *Rough set theory and approach*. Beijing:Science Press(in Chinese) ,(2001)
3. Li D.Y., et al.: *Artificial intelligence with uncertainty*. Beijing:National Defense Industry Press(in Chinese),2005
4. Wierman M.J.: Measuring uncertainty in rough set theory. *International Journal of General Systems*. **28(4)** (1999) 283-297
5. Shannon C.E.: The mathematical theory of communication. *The Bell System Technical Journal*. **27(3-4)** (1948) 373-423
6. Miao D.Q., Wang J.: An information representation of the concepts and operations in rough set theory. *Journal of Software*(in Chinese). **10(2)** (1999) 113-116
7. Wang G.Y.: Algebra view and information view of rough set theory. *Proceeding of SPIE*. **(**4384) (2001) 200-207
8. Wang G.Y., et al.: Decision table reduction based on conditional information entropy . *Chinese journal of Computers*(in Chinese). **25(7)** (2002) 759-766
9. Liang J.Y., Dang C.Y., et al.: A new method for measuring of rough sets and rough relational databases. *Information Sciences*. **31(4)** (2002) 331-342
10. Shi C.Y., Liao S.Z.: *Qualitative Reasoning Methods* .Beijing:Tsinghua University Press (in Chinese),(2002)

A Hybrid Incremental Clustering Method-Combining Support Vector Machine and Enhanced Clustering by Committee Clustering Algorithm

Deng-Yiv Chiu[1] and Kong-Ling Hsieh[2]

Department of Information Management, ChungHua University
Hsin-Chu, Taiwan 300, R.O.C.
chiuden@chu.edu.tw,
mi89041@mi.chu.edu.tw

Abstract. In the study, a new hybrid incremental clustering method is proposed in combination with Support Vector Machine (SVM) and enhanced Clustering by Committee (CBC) algorithm. SVM classifies the incoming document to see if it belongs to the existing classes. Then the enhanced CBC algorithm is used to cluster the unclassified documents. SVM can significantly reduce the amount of calculation and the noise of clustering. The enhanced CBC algorithm can effectively control the number of clusters, improve performance and allow the number of classes to grow gradually based on the structure of current classes without clustering all of documents again. In empirical results, the proposed method outperforms the enhanced CBC clustering method and other algorithms. Also, the enhanced CBC clustering method outperforms original CBC.

1 Introduction

Most of the early clustering algorithms use single clustering technique. There are many restrictions on those algorithms. For example, K-mean algorithm [1] is easily influenced by noise and outlier. DBSCAN algorithm [2] requires the user to enter the parameters. Therefore, recent researches combine different clustering algorithms to improve the quality of clustering. For example, the BRIDGE algorithm [3] combines K-means algorithm and DBSCAN algorithm. K-mean algorithm is simple and fast. DBSCAN algorithm is not easily influenced by outliers. However, the clustering quality of BRIDGE algorithm is still influenced by the different input parameters.

To solve problems above, some researches combine estimation formula of similar clusters into clustering algorithms so that similar clusters can be merged. For example, the Relative Interconnectivity (RI) method and the Relative Closeness (RC) method proposed by Karypis [4], have considered the distance of the two clusters (RI) and closeness (RC) into the calculation and led out a composite index. According to the experimental results, this technique reaches a better result of merging. However, the time consumption is still high.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 465–472, 2007.

A good clustering algorithm must be able to deal with the noise, find any form of clustering, get high quality of clusters and low time complexity. For this reason, Patrick Pantel [5] has proposed clustering by committee algorithm (CBC). This algorithm can automatically find out the proper number of clusters, increase performance of clustering and classify documents to multi clusters.

Though the performance of CBC algorithm is excellent, however, it must re-cluster all of documents again when new documents come in. In view of this, this study tries to combine the classification and clustering technologies, and proposes a new clustering algorithm: hybrid incremental clustering method. First, the new document is classified by Support Vector Machine (SVM) according to the current classes. The reason of choosing SVM is that it has the characteristic of fast, stable and it does not require much training to get good classification result. Besides, in many researches, its performance is better than other methods [6]. Then the enhanced CBC algorithm proposed by this study is used to cluster the unclassified documents to form new classes, and the new classes will be added to the existing classes so that the number of classes will increase gradually.

In the algorithm, SVM can significantly reduce the amount of calculations and the noise of clustering. Enhanced CBC algorithm can effectively control the number of clusters, improve performance and allow the number of classes to grow gradually based on the structure current classes without clustering all of documents again. According to the experimental results, the hybrid incremental clustering performance is not just better than the enhanced CBC algorithm, it is obviously better than other algorithms. Besides, the enhance CBC algorithm outperforms the original CBC algorithm.

2 Hybrid Incremental Clustering Method

The structure of hybrid incremental clustering is shown in Figure 1. First, the experimental data and structure of classification are collected from the internet. The collected documents are segmented by the Chinese Words Database and CKIP Chinese Word Segmentation System. According to the morphological features, lexicons are divided into several word classes and the necessary word classes are captured. Next, Chi-square is used to find out the features and to establish the vector space. The vector space is sent to SVM to try to classify the incoming documents into any existing class, if possible. After processing all of incoming documents, we check if there is any unclassified document. If so, TFIDF is used to find out the features of the unclassified documents and establish the vector space. Enhanced CBC algorithm is then used to cluster the unclassified documents. If there are still any unclustered documents, they will be used as a part of input for next cycle. Finally, the evaluation is performed.

2.1 First Phase: Classification with SVM

Support Vector Machine (SVM) is developed from Statistical Learning Theory (SLT) [7]. The goal is to find linear functions in high dimensional space,which can

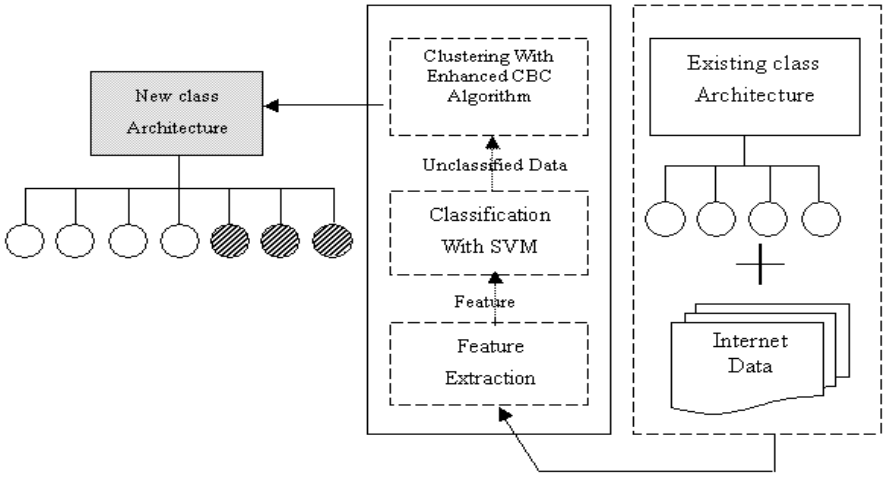


Fig. 1. Structure of hybrid incremental clustering method

discriminate information. Those functions can be used to represent the support vector of the information clusters and some extreme values will be rejected in advance.The basic definition of SVM is as follows:

Provided the existing training data: $(x_1, y_1), ...(x_p, y_p)$, where $x_i \epsilon R^n, y_i \epsilon\{1, -1\}$, p is the number of data and n is the number of vector spaces. When y equals to 1, the document belongs to the class; when y equals to -1, the document does not belong to the class. In the linear analysis, in an optimal hyperplane, $(w \cdot x) + b = 0$ can completely separate the sample into two conditions shown as below, where w is the weight vector and b is a bias.

$$(w \cdot x) + b \geq 0 \rightarrow y_i = +1,$$

$$(w \cdot x) + b \leq 0 \rightarrow y_i = -1,$$

In the linear separation, it is a typical quadratic programming problem. Lagrange formula can be used to find the solution, where α is a Lagrange multiplier.

$$L(w, b, \alpha) = \tfrac{1}{2}\|w\|^2 - \textstyle\sum_{i=1}^{p} \alpha_i [y_i(w \cdot x_i + b) - 1],$$

In the linear analysis, the original problem can be considered as a dual problem. To find the optimal solution, the approach is:

$$max \quad W(\alpha) = \textstyle\sum_{i=1}^{p} \alpha_i - \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} \alpha_i \alpha_j y_i y_j x_i^T x_j.$$

Constraint:

$$\textstyle\sum_{i=1}^{p} \alpha_i y_i = 0, \quad \alpha_i > 0, \quad i = 1, 2, ..., p.$$

By solving the quadratic programming, the classification formula applied to classification can be obtained as shown below.

$$f(x) = Sign(\sum_{i=1}^{p} y_i \alpha_i (x \cdot x_i) + y_i - w \cdot x_i) \tag{1}$$

Any functions that meet Mercer's condition can be kernel functions. We adopt Radial kernel function below as kernel function of SVM.

$$K(s,t) = exp(-\frac{1}{10}\|s-t\|^2) \tag{2}$$

2.2 Second Phase: Clustering with Enhanced CBC Algorithm

Then enhanced CBC algorithm is used to cluster the unclassified documents. Since CBC algorithm has excellent performance on clustering, therefore, this study modifies original CBC algorithm to determine beginning number of the clusters. Then the merging process of similar clusters is applied to merge clusters and determine the final number of clusters. The similarity of each document is first calculated. The value of TFIDF of each word is calculated and sorted in descending order. Taking discrimination and time consumption into consideration, threshold of TFIDF value is set to two. Then the similarity matrix of each document is calculated by VSM. Finally, in order to avoid missing the documents with high degree of similarity, the number of desired clusters is replaced by similarity threshold in this study.

First, the clusters of documents with high degree of similarity are temporary stored in committee candidates (L) (Step 1), and are sorted in descending order according to the similarity of average-link. It is the basis of selecting committees. The clusters in the L are examined and the similarity formula is used to calculate the density of the clusters. If similarity between each document and the centroid is greater than threshold θ_1, then those documents form a committee (Step 3). And make them form a committee. It becomes the new basis of classification for other documents. In this approach, the numbers of committee are fixed so that the amount of calculation can be reduced.

Then, all of documents that are not classified into any committee are examined. If the similarity between a document and the central point of a committee is greater than the threshold θ_2, the document will be put to the corresponding committee. Otherwise, the document will be classified to set R (Step 4). Next, it will check whether set C or set R is empty, or whether limited CBC algorithm has been executed twice. If it is true, then clustering is ended and enter next step. Otherwise, it will go back to the first step and the documents in set R are regarded as a part of new input for next cycle. According to the experimental results, most of the documents in set R are outliers at this stage. If it takes too much time to force clustering of these documents in set R, the performance of clustering algorithm will be affected and time consumption will increased. Therefore, recursive_times is set to not more than two.

Finally, it will check whether set C is empty. If not, the merging mechanism will be activated (Step 5). Some studies have further discussed in merging mechanism [8]. Through the study and experiment, we choose V as the central vector of a cluster. θ_3 is the coefficient. If θ_3 is large, the similarity degree of two clusters must be high so that they can be merged (When θ_3 equals to 2, it means that V_1 and V_2 must be the same in order to be merged).

In the process of testing, we find that significant performance can be reached when θ_1 is set to 0.4, θ_2 is set to 0.35 and θ_3 is set to 1.92 with trial and error method. Therefore, those coefficients are adopted.

Enhanced CBC algorithm

Input:

$E = \{e_1, e_2, \cdots, e_n\}$ //Set of elements
S // A similarity database
θ_1,θ_2,θ_3 // Thresholds

Output:

C // Set of committees

First phase: Limited CBC algorithm

Repeat

1. For each e ∈E do
   Cluster the top-similar elements of e from S by average-link
   For each c do
   Compute the score:| c|* avgsim(c)
   // |c| is the number of elements in c
   // avgsim(c) is the average pairwise similarity between each e
   Store the highest-scoring cluster (c) in L
   // L contains committee candidates
2. Sort each c's score in L by descending order
3. Set C to ∅ //Set of committees, initially empty
   Compute centroid of each c
   For each c ∈ L in sorted order do
   Compute similarity between each e and the centroid, e ∈c
   If the similarity is greater than θ_1 then
   C = C ∪ c
4. If C != ∅ then
   Compute centroid of each c
   For each e ∈ E do
   Compute similarity between each e and the centroid
   If e's similarity to every c ∈ C is greater than θ_2 then
   R = R ∪ e // Set of residues
   E = R

Until (C = ∅) or (R = ∅) or (Recursive_times > 2)

Second phase: Expanded clustering merging

5. If C != ∅ then
   For each I ∈ C do

For each J ∈ C, J≠I do
Compute : $\| V_1 + V_2 \|$,$\| V_1 \|$,$\| V_2 \|$
// V_1 is I's centroid vector, V_2 is J's centroid vector
If $\| V_1 + V_2 \| > (\theta_3 * \| V_1 \|)$ and $\| V_1 + V_2 \| > (\theta_3 * \| V_2 \|)$ then
Merge I and J
Return C

3 Experiments

The news webpage of YAHOO are used as the targets for verification. The collection duration is from 2006/1/1 to 2006/1/13. There are totally 12 classes and 4187 documents. Besides, the F-measure formula in experiment is set to 2PR/P+R, where P is precision and R is recall.

3.1 Evaluation of Enhanced CBC Algorithm

We compare the enhanced CBC algorithm with the original CBC algorithm in terms of precision, recall and F-measure. The experimental results are shown in Table 1. In Table 1, the average performances of precision, recall and F-measure have increased 13%, 5% and 14% respectively. The number of clusters is also under control, and the average number of clusters has dropped from 60.7 to 4.7. When the degree of similarity is 0.3, the performance is excellent. This experiment shows that mergence of the similar clusters can increase the performance and reduce the number of clusters. Besides, we find when the threshold value of similarity degree is set to 0.7 in merging mechanism of similar clusters, most of the documents can be classified to proper classes fast.

Table 1. Performance of enhanced CBC algorithm and original CBC algorithm

Similarity	Enhanced CBC algorithm				Original CBC algorithm			
	Number of clusters	Precision	Recall	F-measure	Number of clusters	Precision	Recall	F-measure
0	5	0.29	0.52	0.37	64	0.12	0.35	0.18
0.1	4	0.29	0.52	0.37	64	0.12	0.35	0.17
0.2	4	0.29	0.52	0.37	64	0.13	0.38	0.19
0.3	3	0.29	0.52	0.37	64	0.10	0.38	0.15
0.4	4	0.29	0.40	0.34	63	0.10	0.36	0.15
0.5	2	0.29	0.40	0.34	63	0.08	0.39	0.14
0.6	4	0.29	0.40	0.34	63	0.09	0.38	0.15
0.7	7	0.09	0.34	0.14	58	0.08	0.39	0.13
0.8	7	0.09	0.34	0.14	54	0.09	0.40	0.14
0.9	7	0.09	0.34	0.14	50	0.08	0.42	0.13
Average	4.70	0.23	0.43	0.29	60.70	0.10	0.38	0.15

Table 2. Performance comparisons of proposed method and enhanced CBC algorithm

Similarity	Proposed method			Enhanced CBC algorithm (without SVM)		
	Precision	Recall	F-measure	Precision	Recall	F-measure
0	0.56	0.48	0.52	0.29	0.52	0.37
0.1	0.56	0.48	0.52	0.29	0.52	0.37
0.2	0.56	0.48	0.52	0.29	0.52	0.37
0.3	0.56	0.48	0.52	0.29	0.52	0.37
0.4	0.59	0.45	0.51	0.29	0.40	0.34
0.5	0.59	0.45	0.51	0.29	0.40	0.34
0.6	0.59	0.45	0.51	0.29	0.40	0.34
0.7	0.32	0.44	0.37	0.09	0.34	0.14
0.8	0.32	0.44	0.37	0.09	0.34	0.14
0.9	0.32	0.44	0.37	0.09	0.34	0.14
Average	0.50	0.46	0.47	0.23	0.43	0.29

3.2 Evaluation of Proposed Hybrid Incremental Clustering Method

The performances of the proposed method and enhanced CBC algorithm are shown in Table 2. Table 3 shows the average performance of the proposed method and others.

In Table 2, performance of the proposed method raise obviously. The average performances of precision, recall and F-measure have increased 27%, 3% and 18% respectively. When the degree of similarity is 0.7, the performance is excellent. The reason for this result is that all documents have been filtered in the first phase of classification. Therefore, the noise of clustering in second phase has been reduced a lot.

In Table 3, the values of precision, recall and F-measure in the proposed method rise 13.4%, 3.8% and 8.4% compared with the average value (36.6%, 42.2% and 38.6%) of Chameleon, Average-link, Buckshot, Bisecting K-means and Complete-link.

Table 3. Average performances of the proposed method and others

Approach	Precision	Recall	F-measure
Proposed method	0.5	0.46	0.47
Chameleon	0.41	0.43	0.42
Average-link	0.43	0.40	0.41
Buckshot	0.38	0.44	0.40
Bisecting K-means	0.37	0.41	0.39
Complete-link	0.24	0.43	0.31

4 Conclusions

This study proposes a new hybrid incremental clustering method. The method applies SVM, enhanced CBC algorithm and merging mechanism of similar

clusters. The noise of clustering is reduced. The number of clusters is under control. It simplifies the steps so that the speed of clustering is increased. The number of clustering classes grows gradually without re-clustering existing documents again when new documents come in. Those improvement can be referenced for future studies.

Acknowledgement

This research was supported by the Chung Hua University under the grant no. CHU-95-M-20.

References

1. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability(1967)281-297
2. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings of the Knowledge Discovery and Data Mining(1996)226-231
3. Dash, M., Liu, H., Xu, X.: Merging Distance and Density based Clustering. In: Proceedings of the Database Systems for Advanced Applications(2001)18-20
4. Karypis, G., Han, E.-H., Kumar, V.: Hierarchical Clustering Using Dynamic Modeling. IEEE Computer 32(1999)68-75
5. Pantel, P., Lin, D.: Document Clustering with Committees. In: Proceedings of the 25^{th} International ACM SIGIR Conference on Research and Development in Information Retrieval(2002)199-206
6. Davidov, D., Gabrilovich, E., Markovitch, S.: Parameterized Generation of Labeled Datasets for Text Categorization based on a Hierarchical Directory. In: Proceedings of the 27^{th} Annual International ACM SIGIR(2004)250-257
7. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag, New York(1995)
8. Vats, N., Skillicorn, D.B.: Information Discovery within Organizations Using the Athens System. In: Proceedings of the 2004 Conference of the Center for Advanced Studies on Collaborative Research(2004)282-292

CCRM: An Effective Algorithm for Mining Commodity Information from Threaded Chinese Customer Reviews

Huizhong Duan, Shenghua Bao, and Yong Yu

Department of Computer Science
Shanghai Jiao Tong University
Shanghai 200240, P.R. China
{summer, shhbao, yyu}@apex.sjtu.edu.cn

Abstract. This paper is concerned with the problem of mining commodity information from threaded Chinese customer reviews. Chinese online commodity forums, which are developing rapidly, provide a good environment for customers to share reviews. However, due to noises and navigational limitations, it is hard to have a clear view of a commodity from thousands of related reviews. Further more, due to different characters between Chinese and English, Researching approaches may vary a lot. This paper aims to automatically mine out key information from commodity reviews. An effective algorithm, i.e. Chinese Commodity Review Miner (CCRM) is proposed. The algorithm can be divided into two parts. First, we propose an efficient rule based algorithm for commodity feature extraction as well as a probabilistic model for feature ranking. Second, we propose a top-to-down algorithm to reorganize the extracted features into hierarchical structure. A prototype system based on CCRM is also implemented. Using CCRM, users can easily acquire the outline of a commodity, and navigate freely in it.

Keywords: Commodity feature extraction, ranking, reorganization, algorithm.

1 Introduction

Nowadays, Chinese online commodity forums have been increasing at an incredible speed both in number and in size. These are not only great places for customers to review the commodities they concern about, but also good resources for prospective buyers as well as competing companies to survey their relative commodities. However, to perform such survey is not an easy task. Customer reviews often contain a large sum of noises. This is especially true in Chinese commodity forums. Besides, when searching for a commodity, the forum often returns hundreds or even thousands of reviews. With the limited operation and the lack of overall structure, it is too time and energy consuming to browse all the reviews. A full survey is nearly impossible.

In this paper, an effective algorithm called CCRM is proposed. CCRM can be divided into two stages: 1). automatic commodity feature extraction and ranking, 2) commodity feature reorganization.

The first stage also involves two steps. For the extraction step, we proposed an efficient rule based algorithm. This is a semi supervised algorithm. We use seeds to

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 473–480, 2007.

prepare an original rule set. Then we propose an improved association rule mining algorithm (IARM) to mine out useful and representative rules from the original rule set. These rules are then used to extract commodity features from reviews. For the commodity feature ranking step, a probabilistic model is proposed. We use total probability formula to calculate the probability of each extracted feature.

In the second stage, we propose a top-to-down algorithm, which finds out the sub events of each commodity feature and map them into the extracted feature set. This algorithm is based on an observation of the characteristic of Chinese language.

CCRM automatically mines out the reviewed features of a commodity from its threaded relative reviews, and then reorganize these features into hierarchical structure, which serves as an outline of the commodity. As a case study we applied the CCRM into a corpus of the pconline product forum[1], experimental results show that with the help of CCRM, users can easily find out what aspects of the commodity are reviewed.

The rest of this paper is organized as follows. Section 2 discusses some related but different work. In Section 3 and 4, we present the algorithm for commodity feature extraction and reorganization in detail. Section 5 shows the experiment results. Finally, we give a conclusion in Section 6.

2 Related Work

Mining commodity information from customer reviews has been studied a lot in recent years. But to our knowledge, there has been no similar algorithm as CCRM, even in English.

Former researches mostly focus on the extraction of sentiment information and the classification of sentiment polarity for customer reviews [1, 2, 7, 8, 15]. Their purposes are to perform judgment on a certain commodity. However, as Liu *et al*. [14] point out, sentiment classification is based on the extraction of commodity features, for every sentiment information is related to a feature, instead of the commodity as a whole. In this paper, we focus on the extraction of commodity feature. We leave sentiment information extraction and sentiment classification to our future work.

Commodity feature extraction, to certain extent, is similar to the task of keyword extraction. They are all aimed at finding the representative words from a passage. There are mainly three approaches for keyword extraction. The first one is based on the statistical information of words [3, 6, 11, 12], e.g. term frequency, position and POS tag. The second approach is to build a keyword dictionary; words will only be extracted according to the dictionary. The third one is the rule based approach, which aims at discovering the general rules of keywords from their context. As the commodity features lack similar statistical information, the first approach tends to produce lots of meaningless terms, the extraction result can be very poor [10]. The disadvantage of the second approach is obvious: it cannot find unrecorded terms. In this paper, we use the rule based method to extract the commodity feature because of the observation that reviewers tend to use similar phrases when they comment.

[1] http://itbbs.pconline.com.cn/

[10] first applied association rule mining algorithm (ARM) into commodity feature extraction, but it does not rank the extracted features or build hierarchical structure. Ranking has always been a hot topic in IR research. Studies have been made to rank the web pages and search results [4, 5, 9, 13]. Our task is to rank commodity features extracted by rules, which differ a lot from web pages.

For commodity feature reorganization, it is a novel try. Existing clustering techniques, e.g. [16], can also be used to build dynamic cluster structure of web content. However, the precision is not high enough to put into practical use.

3 Automatic Commodity Feature Extraction

3.1 Rule Mining and Feature Extraction

We use semi supervised rule mining to build target rule set. By "semi" here, it means we do not manually label all the training data; instead, we use seeds to automatically form original rule set. This saves much time and energy. The steps are as follows:

1. We first manually extracted features from reviews of two commodities. These features are used as seeds to extract all the sentences containing them from the training data. Below is an example of the sentences, we will use this example to describe the following processes.

Sentence: "5022最大的亮点就是屏幕" ("The most outstanding point of 5022 is the screen"). The feature word is "屏幕" ("screen").

Stopwords and digits are then removed from these sentences, POS tagging is also performed. After this, the sentence is formatted into below:

"<adv>最\<adj>大\<n>亮点\<verb>是\<n>屏幕"

("<adv>most\<adj>outstanding\<n>point\<verb>is\<n>screen")

2. Sentences with POS tags are then segmented into triples using the rules below:

(1). The feature term with two words before it.

(2). The feature term with one word before it and one word after it.

(3). The feature term with two words after it.

Then we replace the feature term with a general term "[feature]". Below are the accepted segments of the example sentence:

"<n>亮点\<verb>是\<n>[feature]" ("<n>point\<verb>is\<n>[feature]")

"<verb>是\<n>[feature]" ("<verb>is\<n>[feature]")

Each segment is transformed into a rule:

<n>亮点, <verb>是 → <n>[feature] (<n>point, <verb>is → <n>[feature])

<verb>是 → <n>[feature] (<verb>is → <n>[feature])

3. The result of step 2 is seen as the original rule set. We then perform IARM to mine out useful and representative rules. The reason to propose this Improved Association Rule Mining algorithm is that, based on our observation, rules generated by Association Rule Mining algorithm are too specific and therefore the recall of commodity feature extraction can be very low. In order to complement this, we have to lower the minimal confidence and support to get more rules. However, the cost of doing so is the great decrease in precision and efficiency. Through observation, we

find that most rules could be generalized. The utmost generalization is to use the POS tag of words to form rules. This is obviously infeasible because the precision is too low. Based on this observation we improve ARM as follows:

The ARM way is, for each rule candidate "A, B → C", calculate the confidence and support separately. If the two values are all above the minimal threshold, the rule is selected.

The IARM way is, for each rule in the original rule set, e.g.:

<n>亮点, <verb>是 → <n>[feature] (<n>point, <verb>is → <n>[feature])

We note it as: <posA>A, <posB>B → <posC>C

We separate it into two rules: A, B → C

<posA>, <posB> → <posC>

For these two rules, we calculate confidence and support separately; and then we use linear addition to merge them together:

$$confidence = \lambda \cdot confidence_{word} + (1-\lambda) \cdot confidence_{pos} \tag{1}$$

$$support = \lambda \cdot support_{word} + (1-\lambda) \cdot support_{pos} \tag{2}$$

We empirically use $\lambda = 0.999$ in IARM.

4. Other consideration

In our algorithm, we also allow rules formed by bigram tuples. In order to balance their confidence and support, we multiply each of them by a punishment function.

When we use rules for mining commodity features, we also allow gaps in rules.

3.2 Feature Ranking

In this step, we propose a statistical model for commodity feature ranking. To rank the extracted feature, we calculate the probability of each feature $P(f)$. The detailed model is described below.

$$\begin{aligned} P(f) &= \sum_{r \in R} P(f \mid r) \cdot P(r) = \sum_{r \in R} \left[\sum_{d \in D} P(f \mid r, d) \cdot P(d \mid r) \right] \cdot P(r) \\ &\approx \sum_{r \in R} \sum_{d \in D} P(f \mid r, d) \cdot P(d) \cdot P(r) \end{aligned} \tag{3}$$

In the formula, R is the rule set and D is the review document set. We assume that R and D are independent. We first apply the total probability formula to $P(f)$ in the rule space, and then in the review document space. For $P(r)$ in the formula, we use the support value of rule r; and for $P(d)$, we see it as constant. Then we calculate $P(f \mid r,d)$ using the formula below:

$$P(f \mid r, d) = \frac{freq(f, r, d)}{L_{r,d}} \tag{4}$$

In this formula, *freq*(*f, r, d*) is the count that commodity feature *f* is matched in document *d* by rule *r*. $L_{r,d}$ is the count of all the commodity features matched in document *d* by rule *r*.

Then we use *P*(*f*) to rank the extracted features, and by setting a threshold we are able to filter out most meaningless terms. Besides, we notice that in Chinese, terms formed by single character can not have exact meaning, so we also filter these terms.

4 Commodity Feature Reorganization

For this stage, we propose a top-to-down algorithm to build the ranked features into hierarchical structure. The basic idea of this algorithm is based on the observation:

Observation: In Chinese language, for two successively appearing events A and B in a sentence (A appears before B), it is normal that B is affiliated to A, but it barely happens that A is affiliated to B.

Table 1. Reorganization Algorithm

Algorithm Reorganization (*F*)	
Input	Given a collection of extracted commodity features *F*.
1	**For** each commodity feature *f* in *F* **Do**
1-1	**For** each related review of *f* **Do**
1-2	Find out sentences *S* containing *f*.
1-3	Find out all the events that appear after f in *S*.
1-4	If event *e* is successfully mapped into *F*, add a sub feature *sf* named *e* for *f*, move related reviews from *e* and *f* to *sf*.
1-5	**For** each commodity feature *f* in *F* **Do**
	If *f* has no related reviews and no sub feature, delete *f*.
2	Repeat 1 to the maximal defined level.
Return	Hierarchy *H* for *F*.

Based on the observation, we propose the reorganization algorithm in Table 1.

5 Evaluations

5.1 Experiment Preparation

We crawl down a part of the mobile phone and notebook forum from pconline product BBS as our experiment data. We randomly divide the data into two parts, one for training and the other for testing. Detailed data description can be seen in Table 2.

A proto type System is implemented based on CCRM. Using this data, we evaluated the extraction and ranking algorithm of CCRM over three widely accepted information retrieval metrics, namely *Precision, Recall,* and *Bpref.*

Table 2. Detailed Data Description

	Commodities	Mobilephone	Notebook	Threads	Reviews
All	23	9	14	6987	13006
Training	18	7	11	5845	10946
Testing	5	2	3	1142	2060

5.2 Experimental Results

Table 3 and Table 4 show the mining precision, recall and No. of rules for ARM and IARM with different minimal support. The ARM here is similar to the method in [10], but we adopted semi supervised learning. For minimal confidence, we empirically set it to 0.5. In Table 3 we see, with lower minimal support, ARM tends to generate large number of rules and get a higher recall, but meantime, the precision decreases a lot. For IARM in Table 4, we see the lowering of minimal support does not result in great decrease in precision; instead, the increase of recall is remarkable. This means IARM effectively filters out most of useless rules and mines out the really representative ones. Finally we tested the filtering function based on ranking, it has overall improvement on the precision, but has a negative effect on recall.

Table 3. Commodity Feature Extraction Results Using ARM

	NO. of labeled features	ARM									
		minsp=0.001		minsp=0.002		minsp=0.003		minsp=0.004		minsp=0.005	
		Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
commodity 1	72	0.776	0.820	0.793	0.694	0.804	0.625	0.818	0.625	0.823	0.583
commodity 2	64	0.779	0.828	0.840	0.656	0.844	0.593	0.833	0.547	0.825	0.516
commodity 3	60	0.813	0.650	0.857	0.500	0.879	0.483	0.900	0.450	0.885	0.383
commodity 4	82	0.769	0.854	0.787	0.720	0.797	0.671	0.781	0.610	0.806	0.610
commodity 5	38	0.765	0.684	0.821	0.605	0.846	0.579	0.857	0.474	0.842	0.421
Avg.	**63**	**0.78**	**0.77**	**0.82**	**0.64**	**0.83**	**0.59**	**0.84**	**0.54**	**0.84**	**0.50**
No. of rules		478		199		111		76		64	

Table 4. Commodity Feature Extraction Results Using IARM

	NO. of labeled features	IARM (λ=0.999)						IARM (λ=0.999) with filtering	
		minsp=0.002		minsp=0.003		minsp=0.004		minsp=0.002	
		Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
commodity 1	72	0.822	0.833	0.828	0.736	0.818	0.625	0.845	0.833
commodity 2	64	0.841	0.828	0.870	0.734	0.870	0.625	0.864	0.797
commodity 3	60	0.864	0.633	0.875	0.583	0.903	0.467	0.860	0.617
commodity 4	82	0.791	0.829	0.792	0.744	0.776	0.634	0.805	0.805
commodity 5	38	0.818	0.711	0.839	0.684	0.875	0.553	0.844	0.711
Avg.	**63**	**0.83**	**0.77**	**0.84**	**0.70**	**0.85**	**0.58**	**0.84**	**0.75**
No. of rules		429		238		123		429	

We note that [10] reported much higher minimal support. That is mainly because it uses supervised mining algorithm and manually labels all the training sentences. Labeling itself filters out most of the noise rules. Apparently our algorithm is much more practical and energy saving.

Table 5. Ranking Results of Probability Model

	commodity 1	commodity 2	commodity 3	commodity 4	commodity 5	Avg.
bpref	0.685	0.610	0.595	0.703	0.941	0.71

Table 5 shows the ranking evaluation of our proposed probability model. Here we use bpref as the measuring function and IARM with filtering as the mining approach. On the 5 testing commodities, CCRM achieves an average bpref of 0.71.

Fig. 1. A Sample Output of Compaq Presario B2803tx in the Prototype System

For the reorganization algorithm, as there is no convincing metric, we do not perform quantitative evaluation. Instead, as a case study, we consider the result of Compaq Presario B2803tx in the prototype system in Figure 1. We note that there are 71 extracted commodity features and frequent commented features like "机子" (machine) may still have tens or even hundreds of related reviews. If there lacks a hierarchical structure, it would still be inconvenient for visualization and navigation. In this case, "机子" (machine) has 13 sub features, each containing about 5 reviews. In practice we find that more than 2 level hierarchy will result in a tree too big to traverse, thus a maximal level of 2 is just appropriate for this algorithm.

6 Conclusion

In this paper, we concern about mining key information from threaded Chinese customer reviews. In order to provide users a clear overview of a commodity, we propose a novel algorithm named CCRM. CCRM mainly has two contributions:

- It proposes an improved association rule mining algorithm for automatic commodity feature extraction, and a probability model for commodity feature ranking and filtering.
- It proposes a top-to-down algorithm for commodity feature reorganization. Using this algorithm, features can be automatically organized into a hierarchical structure, which serves as an outline of the commodity involved.

References

[1] Bai, X., Padman, R., Airoldi, E.: On Learning Parsimonious Models for Extracting Consumer Opinions. Page 75b of: In Proc. of HICSS-05(2005)Page75b.
[2] Baron, F., Hirst, G.: Collocations as Cues to Semantic Orientation. In Proc. of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications(2004).
[3] Bourigault, D.: Lexter: A terminology extraction software for knowledge acquisition from texts. In Proc. of KAW-95(1995).
[4] Clemencon, S., Lugosi, G., Vayatis-Manuscript, N.: Ranking and scoring using empirical risk minimization. In Proc. of the 18th Annual Conference on Learning Theory(2005).
[5] Cohen, W. W., Schapire, R. E., Singer, Y.: Learning to order things. Journal of Artificial Intelligence Research, 10(1999)243-270.
[6] Daille, B.: Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. The Balancing Act: Combining Symbolic and Statistical Approaches to Language. MIT Press, Cambridge(1996).
[7] Dave, K., Lawrence, S., Pennock, D. M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proc. of WWW-03(2003)519-528.
[8] Gamon, M., Aue, A., Corston-Oliver, S., Ringger, E.: Pulse: Mining Customer Opinions from Free Text. In Proc. of IDA-05(2005)121-132.
[9] Haveliwala, T. H.: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. IEEE Transactions on Knowledge and Data Engineering(2003).
[10] Hu, M., Liu, B.: Mining and summarizing customer reviews. In Proc. of KDD-04(2004).
[11] Jacquemin, C., Bourigault, D.: Term extraction and automatic indexing. In R. Mitkov, editor, Handbook of Computational Linguistics. Oxford University Press(2001).
[12] Justeson, J., Katz, S.: Technical Terminology: some linguistic properties and an algorithm for identification in text. Natural Language Engineering 1(1)(1995)9-27.
[13] Lei, M., Wang, J., Chen, B., Li, X.: Improved relevance ranking in WebGather. Journal of Computer Science and Technology(September 2001)410-417.
[14] Liu, B., Hu, M., Cheng, J.: Opinion Observer: Analyzing and Comparing Opinions on the Web. In Proc. of WWW-05(2005).
[15] Morinaga, S., Yamanishi, K., Tateishi, K., and Fukushima, T.: Mining Product Reputations on the Web. In Proc. of KDD-02(2002).
[16] Zeng, H., He, Q., Chen, Z., Ma, W., Ma, J.: Learning to cluster web search results. In Proc. of ACM SIGIR-04(2004)210-217.

A Rough Set Approach to Classifying Web Page Without Negative Examples

Qiguo Duan, Duoqian Miao, and Kaimin Jin

Department of Computer Science and Technology, Tongji University, Shanghai, 201804,China
The Key Laboratory of "Embedded System and Service Computing", Ministry of Education, Shanghai, 201804,China
dqgcn@126.com,miaoduoqian@163.com,jinkaimin@163.com

Abstract. This paper studies the problem of building Web page classifiers using positive and unlabeled examples, and proposes a more principled technique to solving the problem based on tolerance rough set and Support Vector Machine (SVM). It uses tolerance classes to approximate concepts existed in Web pages and enrich the representation of Web pages, draws an initial approximation of negative example. It then iteratively runs SVM to build classifier which maximizes margins to progressively improve the approximation of negative example. Thus, the class boundary eventually converges to the true boundary of the positive class in the feature space. Experimental results show that the novel method outperforms existing methods significantly.

Keywords: Web page classification, rough set, Support Vector Machine.

1 Introduction

With the rapid growth of information on the World Wide Web, automatic classification of Web pages has become important for effective retrieval of Web documents. The common approach to building a Web page classifier is to manually label some set of Web page to pre-defined categories or classes, and then use a learning algorithm to produce a classifier. The main bottleneck of building such a classifier is that a large number of labeled training Web page is needed to build accurate classifiers. In most cases of automatic Web page classification, it is normally easy and inexpensive to collect positive and unlabeled examples, however, arduous and very time consuming to collect negative training examples and label them by user's own hands.

In this paper, we focus on the problem to classifying Web page with positive and unlabeled data and without labeled negative data. Recently, a few techniques for solving this problem were proposed in the literature. Liu et al. proposed a method (called S-EM) to solve the problem in the text domain [7]. In [8], Yu et al. proposed a technique (called PEBL) to classify Web pages given

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 481–488, 2007.

positive and unlabeled pages. This paper proposes a more effective and robust technique to solve the problem. Experimental results show that the new method outperforms existing methods significantly. Throughout the paper, we call the class of Web page that we are interested in positive and the complement set of samples negative.

The rest of the paper is organized as follows: Section 2 presents the concepts of the tolerance rough set briefly. Section 3 describes proposed technique. Section 4 reports and discusses the experimental results. Finally, Section 5 concludes the paper.

2 Tolerance Rough Set

Rough set theory is a formal mathematical tool to deal with incomplete or imprecise information [2]. The classical rough set theory is based on equivalence relation that divides the universe of objects into disjoint classes. By relaxing the equivalence relation to a tolerance relation, where transitivity property is not required, a generalized tolerance space is introduced below [3],[4],[5],[6].

Let $I : U \rightarrow P(U)$ to denote a tolerance relation, if and only if $x \in I(x)$ for $x \in U$ and $y \in I(x) \Leftrightarrow x \in I(y)$ for any $x, y \in U$, where $P(U)$ are sets of all subsets of U. Thus the relation $xIy \Leftrightarrow y \in I(x)$ is a tolerance relation (i.e. reflexive, symmetric) and $I(x)$ is a tolerance class of x. Define the tolerance rough membership function $\mu_{I,V}$, as $x \in U, X \subseteq U$,

$$\mu_{I,V}(x, X) = \nu(I(x), X) = \frac{|I(x) \bigcap X|}{|I(x)|} . \tag{1}$$

The tolerance rough set for any $X \subseteq U$ are then defined as

$$L_R(X) = \{x \in U | \nu(I(x), X) = 1\} . \tag{2}$$

$$U_R(X) = \{x \in U | \nu(I(x), X) > 1\} . \tag{3}$$

With its ability to deal with vagueness and fuzziness, tolerance rough set seems to be promising tool to model relations between terms and documents. The application of tolerance rough set in classifying Web page using positive and unlabeled examples was proposed as a way to enrich feature and document representation and extract reliable negative examples for improvement of classification.

2.1 Tolerance Space of Terms in Unlabeled Set

Let $U = \{d_1, ..., d_M\}$ be a set of unlabeled Web pages and $T = \{t_1, ..., t_N\}$ set of terms for U. The tolerance space is defined over a universe of all terms for U. The idea of terms expansion is to capture conceptually related terms into classes. For

this purpose, the tolerance relation is determined as the co-occurrence of terms in all Web pages from U.

2.2 Tolerance Class of Term

Let $f_U(t_i, t_j)$ denotes the number of Web pages in U in which both terms t_i and t_j occurs. The uncertainty function I with regards to co-occurrence threshold θ defined as

$$I_\theta(t_i) = \{t_j | f_U(t_i, t_j) \geq \theta\} \cup \{t_i\} \ . \tag{4}$$

Clearly, the above function satisfies conditions of being reflexive: $t_i \in I_\theta(t_j)$ and symmetric: $t_j \in I_\theta(t_i) \Leftrightarrow t_i \in I_\theta(t_j)$ for any $t_i, t_j \in T$. Thus, $I_\theta(t_i)$ is the tolerance class of term t_i.Tolerance class of terms is generated to capture conceptually related terms into classes. The degree of correlation of terms in tolerance classes can be controlled by varying the threshold θ.The membership function μ for $t_i \in T, X \subseteq T$ is then defined as:

$$\mu(t_i, X) = \nu(I_\theta(t_i), X) = \frac{|I_\theta(t_i) \cap X|}{|I_\theta(t_i)|} \ . \tag{5}$$

Finally, the lower and upper approximations of any subset $X \subseteq T$ can be determined with the obtained tolerance relation respectively as [5],[6]:

$$L_R(X) = \{t_i \in T | \nu(I_\theta, X) = 1\} \ . \tag{6}$$

$$U_R(X) = \{t_i \in T | \nu(I_\theta, X) > 0\} \ . \tag{7}$$

2.3 Expansion the Web Pages on Tolerance Class of Term

In tolerance space of term, an expanded representation of Web document can be acquired by representing Web document as set of tolerance classes of terms it contains. This can be achieved by simply representing Web document with its upper approximation, e.g., the Web page $d_i \in U$ is represented by:

$$U_R(d_i) = \{t_i \in T | \nu(I_\theta(t_i), d_i) > 0\} \ . \tag{8}$$

The usage of tolerance space and upper approximation to enrich Web page and term relation allows the proposed technique to discover subtle similarities between positive examples in positive set and latent positive examples in unlabeled set.

3 The TRS-SVM Algorithm

We use TRS-SVM to denote the proposed classification techniques that employ the method based on tolerance rough set to extract reliable negative set and SVM to build classifier. The TRS-SVM algorithm is composed by following steps:

Step1: Preprocessing of Web page in set P and U.

A preprocessing procedure is done as follows: Remove the HTML tag and extract plain text from each Web page. All the extracted words are stemmed. Use a stop list to omit the most common words. Finally, extract term set from positive set P and unlabeled set U respectively, let PT be a term set for P and UT a term set for U.

Step2: Positive feature selection.

This step builds a positive feature set PF which contains terms that occur in the term set PT more frequently than in the term set UT. The decision threshold σ is normally set to 1 but can be adjusted. Here $freq(t_i, X)$ denotes the number of occurrence of term t_i in set X and $|X|$ denotes the total number of Web pages in set X.The detail algorithm is given as follows.

1. Generating the set $\{t_1, \cdots, t_n\}, t_i \in UT \cup PT$;
2. $PF = \emptyset$;
3. For $i = 0$ to n
4. $\quad f_p^i = freq(t_i, P)/|P|, f_u^i = freq(t_i, U)/|U|$;
5. $\quad$ If $f_p^i / f_u^i > \sigma$ then $PF = PF \cup \{t_i\}$;
6. $\quad$ End If
7. End For

Step3: Generating tolerance class of term in unlabeled set and enriching Web page representation.

The goal of this step is to determine for each term in UT, the tolerance class which contains its related terms with regards to the tolerance relation. In our experiment we set $\theta = 7$ for good result. Then, the Web page in unlabeled set is represented with its upper approximation, e.g. the Web page $d \in U$ is represented by $U_R(d)$.

Step4: Expansion the positive feature set on tolerance class of term.

The tolerance class of term in unlabeled set which contains the positive feature term in PF will be merged with PF.The algorithm is given as follows.

1. For each $t_i \in PF \cap UT$;
2. $\quad PF = PF \cup I_\theta(t_i)$;
3. End For

Step5: Generating reliable negative set.

This step tries to filter out possible positive Web pages from U. A Web page in U which upper approximation does not have any positive feature in PF is regarded as a reliable negative example. The algorithm is given as follows.

1. $RN = U$;
2. For each Web page $d \in U$;
3. $\quad$ If $\exists x_j freq(x_j, U_R(d)) > 0$ and $x_j \in PF$ then $RN = RN - d$;
4. $\quad$ End If
5. End For

Step6: building classifier.

This step builds the final classifier by running SVM iteratively with the sets P and RN. The basic idea is to use each iteration of SVM to extract more possible negative data from $U - RN$ and put them in RN. Let Q be the set of remaining unlabeled Web pages, $Q = U - RN$. The algorithm for this step is given as follows.

1. Every Web page in P is assigned the class label +1;
2. Every Web page in RN is assigned the label -1;
3. $i = 1, Pr_0 = 0$;
4. Loop
5. Use P and RN to train a SVM classifier C_i;
6. Classify Q using C_i;
   Let the set of Web pages in Q that are classified as negative be W;
7. Classify positive set P with C_i;
   Set Pr_i as classification precision of P;
8. If $(|W| = 0 || Pr_i < Pr_{i-1})$
   then store the final SVM classifier, exit loop;
9. else $Q = Q - W$;
   $RN = RN \cup W$;
   $i = i + 1$;
10. End If
11. End Loop

The reason that we run SVM iteratively is that the reliable negative set RN extracted by the method based on tolerance rough set may not be sufficiently large to build the best classifier. SVM classifiers can be used to iteratively extract more negative Web pages from Q.There is, however, a danger in running SVM iteratively. Since SVM is very sensitive to noise, if some iteration of SVM goes wrong and extracts many positive Web pages from Q and put them in the negative set RN, then the last SVM classifier will be extremely poor. This is the problem with PEBL, which also runs SVM iteratively. In our algorithm, the iteration stops when there is no negative Web page that can be extracted from Q or the classification precision decreases which indicates that SVM has gone wrong.

4 Experimental Evaluation

4.1 Experiment Datasets

To evaluate the proposed techniques, we use the WebKB data set[1], which contains 8282 Web pages collected from computer science departments of various universities. The pages were manually classified into the following categories: student, faculty, staff, department, course, project, other. In our experiments,

[1] http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

we used only the four most common categories: student, faculty, course, other (respectively abbreviated here as St, Fa, Co, Ot). Each category is employed as the positive class, and the rest of the categories as the negative class. This gives us four datasets. Our task is to identify positive Web pages from the unlabeled set. The construction of each dataset for our experiments is done as follows: Firstly, we randomly select 10% of the Web pages from the positive class and the negative class, and put them into test set to evaluate the performance of classifier. Then, the rest are used to create training sets. For each dataset, a% of the Web pages from the positive class is randomly selected as the positive set P. The rest of the positive Web pages and negative Web pages form the unlabeled set U. Our training set consists of P and U. In our experiments, we range from 10%-70% respectively to create a wide range of settings.

4.2 Performance Measures

To analyze the performance of classification, we adopt the popular F1 measure on the positive class. F1 measure is combination of recall (Re) and precision (Pr), F1=2.Re.Pr/(Re+Pr). Precision means the rate of documents classified correctly among the result of classifier and recall signifies the rate of correct classified documents among them to be classified correctly. The F1 measure which is the harmonic mean of precision and recall is used in this study since it takes into account effects of both quantities.

4.3 Experimental Results and Discussion

We now present the experimental results. For comparison, we include the classification results of the naive Bayesian method (NB)[1], S-EM, OSVM [9] and PEBL. Here, NB treats all the Web pages in the unlabeled set as negative. For SVM implementation, we used the LIBSVM[2]. We set Gaussian kernel as default kernel function of SVM because of its high accuracy. PEBL and OSVM also used LIBSVM. We set $\theta = 7$ for good result in generating tolerance class.

We summarize the average F value results of all a settings in Figure 1. We observe that TRS-SVM outperforms NB, S-EM, OSVM and PEBL. In fact, PEBL performs poorly when the number of positive Web pages is small. When the number of positive Web pages is large, it usually performs well. TRS-SVM performs well consistently. We also ran SVM with positive set and unlabeled set. It for the noisy situation (unlabeled set U as negative set) performs poorly (its F values are mostly close to 0) because SVM does not tolerate noise well. Due to space limitations, its results are not listed.

From Figure 1, we can draw the following conclusions: OSVM gives very poor results (in many cases, F value is around 0.3-0.5). PEBL's results are extremely poor when the number of positive Web pages is small. We believe that this is because its strategy of extracting the initial set of reliable negative Web pages could easily go wrong without sufficient positive data. S-EM's results are worse than TRS-SVM. The reason is that the negative Web pages extracted from U by

[2] http://www.csie.ntu.edu.tw/∼ cjlin/libsvm/

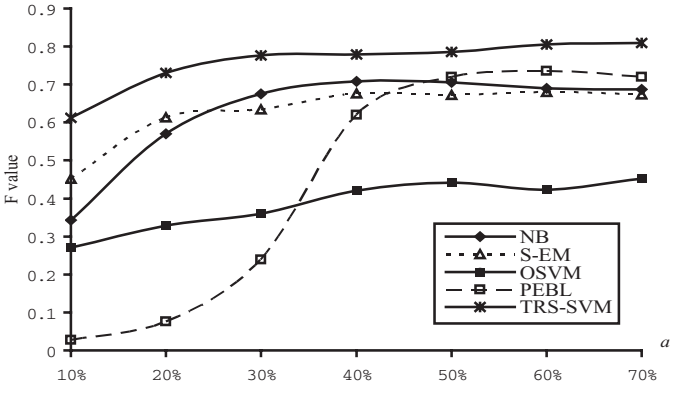


Fig. 1. Average results for all a settings

its spy technique are not reliable. We observe that a single NB slightly outperforms S-EM. TRS-SVM performs well with different numbers of positive Web pages.

Sensitiveness to co-occurrence threshold parameter: Co-occurrence threshold parameter θ is rather important to our TRS-SVM. From definition of tolerance class it is not difficult to get such deduction that inadequate co-occurrence threshold can decrease the performance of the classification results: on one hand, too small co-occurrence threshold can make too many negative examples be extracted as positive examples, on the other hand, too large co-occurrence threshold can make too little latent positive examples be identified from U, both cases can lead to worse performance.

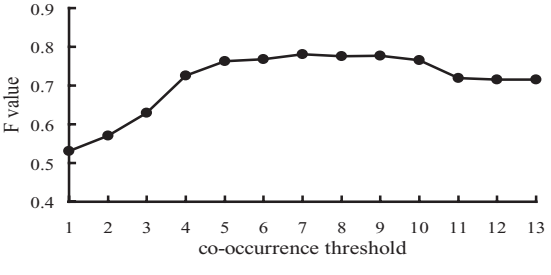


Fig. 2. Sensitiveness to co-occurrence threshold

From Figure 2 we can understand our experimental result corresponds to our deduction: when co-occurrence threshold equals value between 5 and 10, the performance is better, however, when it is out of the interval, the performance is worse (here, a=60% and for other a values, the results are similar).

5 Conclusions

This paper studied the problem of Web page classification with only partial information, i.e., with only one class of labeled Web pages and a set of unlabeled Web pages. An effective technique is proposed to solve the problem. Our algorithm first utilizes the method based on tolerance rough set to extract a set of reliable negative Web pages from the unlabeled set, and then builds a SVM classifier iteratively. The experiment we have carried has showed that the method based on tolerance rough set it offers can extract reliable negative examples by discovering subtle information among unlabeled data, which have positive effects on classification quality. Experimental results show that the proposed technique is superior to S-EM and PEBL.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No.60475019) and the Ph.D. programs Foundation of Ministry of Education of China (No.20060247039).

References

1. Lewis, D., Ringuette, M.: A Comparison of Two Learning Algorithms for Text Categorization. Third annual symposium on document analysis and information retrieval (1994) 81-93
2. Pawlak, Z.: Rough sets: Theoretical Aspects of Reasoning about Data. Kluwer Dordrecht (1991)
3. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27 (1996) 245-253
4. Kryszkiewicz, M.: Rough set approach to incomplete information system. Information Sciences, (1998)112:39-49
5. Tu Bao Ho, Ngoc Binh Nguyen: Nonhierarchical Document Clustering based on A Tolerance Tough Set Model. International Journal of Intelligent Systems, Vol. 17 (2002) 199-212
6. Ngo Chi Lang: A Tolerance Rough Set Approach to Clustering Web Search Results. In: J.-F. Boulicaut et al. (eds.): PKDD 2004. Springer-Verlag, Berlin Heidelberg (2004) 515-517
7. Liu, B., Lee, W. S., Yu, P., and Li, X.: Partially Supervised Classification of Text Documents. ICML-02 (2002)
8. H. Yu, J. Han, and K.C.-C. Chang: PEBL: Web Page Classification without Negative Examples. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 1, 1 (2004) 70-81
9. L.M. Manevitz and M. Yousef: One-Class SVMs for Document Classification. J. Machine Learning Research, vol. 2 (2001) 139-154

Evolution and Maintenance of Frequent Pattern Space When Transactions Are Removed

Mengling Feng[1], Guozhu Dong[2], Jinyan Li[3], Yap-Peng Tan[1], and Limsoon Wong[4]

[1]Nanyang Technological University, [2]Wright State University, [3]Institute for Infocomm Research, & [4]National University of Singapore
[1]{feng0010, eyptan}@ntu.edu.sg, [2]guozhu.dong@wright.edu, [3]jinyan@i2r.a-star.edu.sg, [4]wongls@comp.nus.edu.sg

Abstract. This paper addresses the maintenance of discovered frequent patterns when a batch of transactions are removed from the original dataset. We conduct an in-depth investigation on how the frequent pattern space evolves under transaction removal updates using the concept of equivalence classes. Inspired by the evolution analysis, an effective and exact algorithm TRUM is proposed to maintain frequent patterns. Experimental results demonstrate that our algorithm outperforms representative state-of-the-art algorithms.

1 Introduction

Update is a fundamental data management activity. Data updates allow users to remove expired data, to correct data, and to insert new data. Maintenance of a dynamic dataset and its corresponding discovered knowledge is more complicated compared to the knowledge discovery of a stable dataset. Updates may induce new knowledge and invalidate discovered information. Re-execution of discovery algorithms from scratch every time when a database is updated causes significant computation and I/O overheads. Therefore, effective algorithms to maintain discovered knowledge on the updated database without re-execution of mining algorithms are very desirable.

Databases can be updated in several manners. We focus here on the case when a batch of transactions are removed from the existing database. A novel method is proposed to update and maintain discovered frequent patterns [1].

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of distinct literals called "items". An "itemset", or a "pattern", is a set of items. A "transaction" is a non-empty set of items. A "dataset" is a non-empty set of transactions. A pattern P is said to be contained or included in a transaction T if $P \subseteq T$. A pattern P is said to be contained in a dataset $\mathcal{D}$, denoted as $P \in \mathcal{D}$, if there is $T \in \mathcal{D}$ such that $P \subseteq T$. The "support" of a pattern P in a dataset $\mathcal{D}$, denoted $sup(P, \mathcal{D})$, is the number of transactions in $\mathcal{D}$ that contain P. A pattern P is said to be *frequent* in a dataset $\mathcal{D}$ if $sup(P, \mathcal{D})$ is greater than or equal to a pre-specified threshold ms. Given a dataset $\mathcal{D}$ and a support threshold ms, the collection of all frequent itemsets in $\mathcal{D}$ is called the "space of frequent patterns", and is denoted by $\mathcal{F}(ms, \mathcal{D})$.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 489–497, 2007.

The "space of frequent patterns" can be large. As a result, maximum patterns [3, 7], closed patterns [8], key patterns [11] (also known as generators), and borders of equivalence classes [10] have been proposed to concisely represent the space of frequent patterns. Borders of equivalence classes are arguably the most flexible succinct lossless representation of the frequent pattern space [10]. Conceptually, it partitions the frequent pattern space into equivalence classes that are convex. Then the entire space is represented by the most general and most specific patterns of these equivalence classes. As it turns out, these most general patterns are precisely the key patterns, and these most specific patterns are precisely the closed patterns.

The task of frequent pattern maintenance is to update the "space of frequent patterns" according to the updates of the dataset.

Incremental maintenance, where new transactions are inserted, has attracted intensive research attention. Current incremental maintenance algorithms can be categorized into two main approaches: *Apriori*-based [5, 6, 2] and sliding window filtering (*SWF*) [4, 9]. The performance of both *Apriori*-based and *SWF* algorithms is limited by the candidate-generation-elimination framework, which involves multiple data scans and unnecessary computations on infrequent candidates.

To achieve more efficient updates, algorithms are proposed to incrementally maintain only frequent maximum patterns. ZIGZAG[1] [12] is one effective representative. ZIGZAG is inspired by its related work GenMax [7]. It incrementally maintains maximum patterns by a backtracking search, which is guided by the outcomes of previous maintenance iteration.

Decremental maintenance, where old transactions are removed, on the other hand, has not received as much research attention. Zhang et al. [13] proposed an algorithm, named DUA, to address the decremental maintenance problem. DUA maintains frequent patterns by a pairwise comparison of original frequent patterns and patterns included in the removed transactions. Since the number of frequent patterns is usually enormous, the pairwise comparisons cause heavy computations. In addition, algorithms FUP2H [6], Borders [2], ZIGZAG can also be applied to decremental maintenance with some parameter changes.

It is observed that most previous methods are proposed as an extension of some effective data mining algorithms or data structures. E.g. FUP [5] and Borders [2] are developed based on *Apriori* , and ZIGZAG is inspired by GenMax. Unlike these previous works, our algorithm is proposed based on an in-depth study on the evolution of the frequent pattern space.

2 Basic Properties of Frequent Pattern Space

In [10], we found that the space of frequent patterns can be decomposed into sub-spaces — equivalence classes, as shown in Figure 1 (a).

[1] We thank Adriano Alonoso Veloso, Professor Srinivasan Parthasarathy and Professor Mohammed J. Zaki for providing the ZIGZAG source codes.

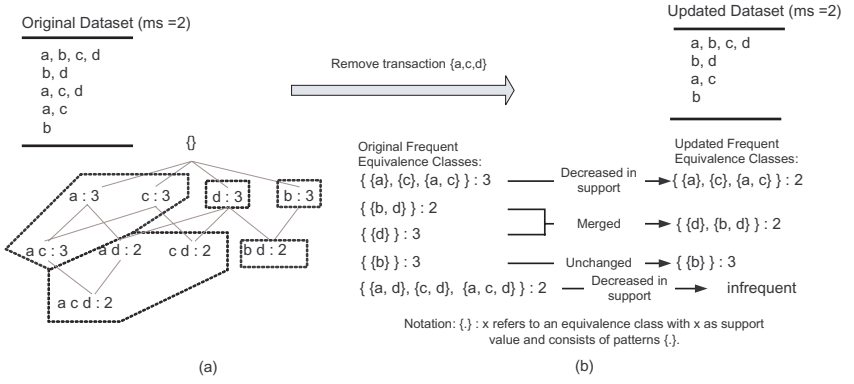


Fig. 1. (a)Demonstration of how a space of frequent patterns, which contains 9 patterns, is decomposed into 5 frequent equivalence classes; (b)demonstration of how equivalence classes may evolve when a transaction is removed

Definition 1. *Let the "filter", $f(P, \mathcal{D})$, of a pattern P in a dataset $\mathcal{D}$ be defined as $f(P, \mathcal{D}) = \{T \in \mathcal{D} \mid P \subseteq T\}$. Then the "equivalence class" $[P]_{\mathcal{D}}$ of P in a dataset $\mathcal{D}$ is the collection of patterns defined as $[P]_{\mathcal{D}} = \{Q \mid f(P, \mathcal{D}) = f(Q, \mathcal{D})$, Q is a pattern in $\mathcal{D}\}$. Note that under this definition, $[Q]_{\mathcal{D}} = \emptyset$ if Q does not appear in $\mathcal{D}$. For convenience in some of our proofs, we also use the traditional notion of an equivalence class, and write it as $[P]^*_{\mathcal{D}} = \{Q \mid f(P, \mathcal{D}) = f(Q, \mathcal{D})\}$.*

In other words, two patterns are "equivalent" in the context of a dataset $\mathcal{D}$ iff they are included in exactly the same transactions in $\mathcal{D}$. Thus the patterns in a given equivalence class have the same support. So we extend the notations and write $sup(C, \mathcal{D})$ to denote the support of an equivalence class and $C \in \mathcal{F}(ms, \mathcal{D})$ to mean the equivalence class is frequent. Figure 1 (a) presents the frequent pattern space for the original dataset with $ms = 2$. In addition, it graphically demonstrates how the space of frequent patterns can be structurally decomposed into frequent equivalence classes.

Structural decomposition of frequent pattern space inspired us to solve the maintenance problem in a divide-and-conquer manner. Instead of maintaining the pattern space as a whole, which is computationally costly, we attack the problem by maintaining each frequent equivalence class. Compared with the frequent pattern space, an equivalence class is much smaller and easier to update. Moreover, not all the equivalence classes are affected by the updates. If we can efficiently locate only those equivalence classes that are affected by the updates, we can solve the problem effectively by updating only the affected equivalence classes. In addition, a nice property of equivalence classes of patterns is that they are convex and they can be concisely represented by their borders. The border of an equivalence class consists of a closed pattern and a group of key patterns [10]. Thus, the corresponding closed and key patterns form the border of and define an equivalence class.

Definition 2. *A pattern P is a "key pattern" in a dataset $\mathcal{D}$ iff for every $P' \subset P$, it is the case that $sup(P', \mathcal{D}) > sup(P, \mathcal{D})$. In contrast, a pattern P is a "closed pattern" in a dataset $\mathcal{D}$ iff for every $P' \supset P$, it is the case that $sup(P', \mathcal{D}) < sup(P, \mathcal{D})$.*

3 Evolution of Frequent Pattern Space

We investigate in this section how frequent patterns, key patterns, closed patterns, equivalence classes and their support values evolve when multiple transactions are removed from an existing dataset. We use the following notations: $\mathcal{D}_{org}$ is the original dataset, $\mathcal{D}_{dec}$ is the set of old transactions to be removed, and $\mathcal{D}_{upd-} = \mathcal{D}_{org} - \mathcal{D}_{dec}$ is the updated dataset. We assume without loss of generality that $\mathcal{D}_{dec} \subseteq \mathcal{D}_{org}$.

An existing equivalence class can evolve in exactly three ways, as shown in Figure 1 (b). The first way is to remain unchanged without any change in support. The second way is to remain unchanged but with a decreased support. If the support of an existing frequent equivalence class drops below the minimum support threshold, the equivalence class will be removed. The third way is to grow—by merging with other classes, where at most one of the merging classes has the same closed pattern and the same support as the resulting equivalence class and all other merging classes have lower support. In short, after the decremental update, the support of an equivalence class can only decrease and the size of an equivalence class can only grow by merging.

In order to have an in-depth understanding of the three ways that an existing equivalence class may evolve, we now provide the exact conditions for each of these ways to occur.

Theorem 1. *For every frequent equivalence class $[P]_{\mathcal{D}_{org}}$ in $\mathcal{D}_{org}$, exactly one of the 6 scenarios below holds:*

1. *P is frequent in $\mathcal{D}_{org}$, P is not in $\mathcal{D}_{dec}$, and $f(P, \mathcal{D}_{org}) \neq f(Q, \mathcal{D}_{org}) - f(Q, \mathcal{D}_{dec})$ for all Q in $\mathcal{D}_{dec}$, corresponding to the scenario where an equivalence class has remained totally unchanged. In this case, $[P]_{\mathcal{D}_{upd-}} = [P]_{\mathcal{D}_{org}}$, $sup(P, \mathcal{D}_{upd-}) = sup(P, \mathcal{D}_{org})$, $f(P, \mathcal{D}_{upd-}) = f(P, \mathcal{D}_{org})$, and the closed pattern of $[P]_{\mathcal{D}_{upd-}}$ is the same as that of $[P]_{\mathcal{D}_{org}}$. The key patterns of $[P]_{\mathcal{D}_{upd-}}$ are the same as that of $[P]_{\mathcal{D}_{org}}$.*
2. *P is frequent in $\mathcal{D}_{org}$, P is not in $\mathcal{D}_{dec}$, and $f(P, \mathcal{D}_{org}) = f(Q, \mathcal{D}_{org}) - f(Q, \mathcal{D}_{dec})$ for some Q occurring in $\mathcal{D}_{dec}$, corresponding to the scenario where the equivalence class of Q has to be merged into the equivalence class of P. In this case, let all such Q's in $\mathcal{D}_{dec}$ be grouped into n distinct equivalence classes $[Q_1]^*_{\mathcal{D}_{dec}}$, ..., $[Q_n]^*_{\mathcal{D}_{dec}}$, having representatives Q_1, ..., Q_n satisfying the condition on Q. Then $[P]_{\mathcal{D}_{upd-}} = [P]_{\mathcal{D}_{org}} \cup \bigcup_i [Q_i]_{\mathcal{D}_{org}}$, $sup(P, \mathcal{D}_{upd-}) = sup(P, \mathcal{D}_{org})$, $f(P, \mathcal{D}_{upd-}) = f(P, \mathcal{D}_{org})$, and the closed pattern of $[P]_{\mathcal{D}_{upd-}}$ is the same as the closed pattern of $[P]_{\mathcal{D}_{org}}$. The key patterns of $[P]_{\mathcal{D}_{upd-}}$ are the most general ones among the key patterns of $[P]_{\mathcal{D}_{org}}$, $[Q_1]_{\mathcal{D}_{org}}$, ..., $[Q_n]_{\mathcal{D}_{org}}$. Furthermore, $[Q_i]_{\mathcal{D}_{upd-}} = [P]_{\mathcal{D}_{upd-}}$ for $1 \leq i \leq n$.*
3. *P is frequent in $\mathcal{D}_{org}$, P is in $\mathcal{D}_{dec}$, and $|f(P, \mathcal{D}_{upd-})| < ms$, corresponding to the scenario where the equivalence class is removed.*

4. *P is frequent in $\mathcal{D}_{org}$, P is in $\mathcal{D}_{dec}$, and $f(Q, \mathcal{D}_{org}) = f(P, \mathcal{D}_{org}) - f(P, \mathcal{D}_{dec})$ for some Q that is frequent in $\mathcal{D}_{org}$ but not in $\mathcal{D}_{dec}$, corresponding to the scenario where the equivalence class of P has to be merged into the equivalence class of Q. This scenario is complement to Scenario 2. In this case, the equivalence class, support, key, and closed patterns of $[P]_{\mathcal{D}_{upd-}}$ is the same as that of $[Q]_{\mathcal{D}_{upd-}}$, as computed in Scenario 2.*
5. *P is frequent in $\mathcal{D}_{org}$, P is in $\mathcal{D}_{dec}$, $|f(P, \mathcal{D}_{upd-})| > ms$, $f(Q, \mathcal{D}_{org}) \neq f(P, \mathcal{D}_{org}) - f(P, \mathcal{D}_{dec})$ for all Q in $\mathcal{D}_{org}$ and not in $\mathcal{D}_{dec}$, and $f(P, \mathcal{D}_{org}) - f(P, \mathcal{D}_{dec}) \neq f(Q, \mathcal{D}_{org}) - f(Q, \mathcal{D}_{dec})$ for all Q in $\mathcal{D}_{dec}$ and $Q \notin [P]_{\mathcal{D}_{org}}$, corresponding to the situation where the equivalence class has remained unchanged but has decreased in support. In this case, $[P]_{\mathcal{D}_{upd-}} = [P]_{\mathcal{D}_{org}}$, $f(P, \mathcal{D}_{upd-}) = f(P, \mathcal{D}_{org}) - f(P, \mathcal{D}_{dec})$, $sup(P, \mathcal{D}_{upd-}) = sup(P, \mathcal{D}_{org}) - sup(P, \mathcal{D}_{dec})$, and the closed pattern of $[P]_{\mathcal{D}_{upd-}}$ is the same as that of $[P]_{\mathcal{D}_{org}}$. The key patterns of $[P]_{\mathcal{D}_{upd-}}$ are the same as that of $[P]_{\mathcal{D}_{org}}$.*
6. *P is frequent in $\mathcal{D}_{org}$, P is in $\mathcal{D}_{dec}$, $|f(P, \mathcal{D}_{upd-})| > ms$, $f(Q, \mathcal{D}_{org}) \neq f(P, \mathcal{D}_{org}) - f(P, \mathcal{D}_{dec})$ for all Q in $\mathcal{D}_{org}$ and not in $\mathcal{D}_{dec}$, and $f(P, \mathcal{D}_{org}) - f(P, \mathcal{D}_{dec}) = f(Q, \mathcal{D}_{org}) - f(Q, \mathcal{D}_{dec})$ for some Q in $\mathcal{D}_{dec}$ and $Q \notin [P]_{\mathcal{D}_{org}}$, corresponding to the situation where the equivalence classes of P and Q have to be merged. In this case, let all such Q's in $\mathcal{D}_{dec}$ be grouped into n distinct equivalence classes $[Q_1]^*_{\mathcal{D}_{dec}}$, ..., $[Q_n]^*_{\mathcal{D}_{dec}}$, having representatives Q_1, ..., Q_n satisfying the condition on Q. Then $[P]_{\mathcal{D}_{upd-}} = [P]_{\mathcal{D}_{org}} \cup \bigcup_i [Q_i]_{\mathcal{D}_{org}}$, $sup(P, \mathcal{D}_{upd-}) = sup(P, \mathcal{D}_{org}) - sup(P, \mathcal{D}_{dec})$, and $f(P, \mathcal{D}_{upd-}) = f(P, \mathcal{D}_{org}) - f(P, \mathcal{D}_{dec})$. The closed pattern of $[P]_{\mathcal{D}_{upd-}}$ is the most specific pattern among the closed patterns of $[P]_{\mathcal{D}_{org}}$, $[Q_1]_{\mathcal{D}_{org}}$, ..., $[Q_n]_{\mathcal{D}_{org}}$. The key patterns of $[P]_{\mathcal{D}_{upd-}}$ are the most general ones among the key patterns of $[P]_{\mathcal{D}_{org}}$, $[Q_1]_{\mathcal{D}_{org}}$, ..., $[Q_n]_{\mathcal{D}_{org}}$. Furthermore, $[Q_i]_{\mathcal{D}_{upd-}} = [P]_{\mathcal{D}_{upd-}}$ for $1 \leq i \leq n$.*

Proof. Refer to the Appendix in the full online version of this paper at http://www.ntu.edu.sg/home5/feng0010/FullPaper.pdf.

This theorem describes in detail how the space of frequent patterns evolves when a group of transactions are removed. Moreover, it describes how to derive equivalence classes in $\mathcal{D}_{upd-}$ from existing equivalence classes in $\mathcal{D}_{org}$, which is an extremely constructive result for the maintenance of frequent patterns.

4 Proposed Algorithm: TRUM

An algorithm for maintaining the frequent pattern space after some transactions are removed from the original database is proposed in Figure 2. In the proposed algorithm TRUM, we use notations $X.closed$ to mean the closed pattern of an equivalence class, $X.keys$ to mean the set of keys of an equivalence class, and $X.sup$ to denote the support value of an equivalence class. The algorithm addresses the maintenance problem effectively by working on the borders of equivalence classes, instead of the entire pattern space. The proposed algorithm is proved to be correct and complete. (Refer to the Appendix in the full online version of this paper at http://www.ntu.edu.sg/home5/feng0010/FullPaper.pdf.)

With proper implementation techniques, the computational complexity of TRUM can be approximated as $O(|\mathcal{D}_{dec}|)$, where $|\mathcal{D}_{dec}|$ denotes the size of the

Input: The set $\mathcal{O} = O_1, ..., O_n$ of frequent equivalence classes in $\mathcal{D}_{org}$,represented by their borders—viz., the corresponding key and closed patterns and supports—and identified by their unique closed patterns, and the set $\mathcal{T} = T_1, ..., T_m$ of transactions in $\mathcal{D}_{dec}$, and the minimum support threshold ms.
Output: The set $O'_1, ..., O'_n$ (if they still exist) of updated frequent equivalence classes in $\mathcal{D}_{upd-}$ represented by their borders and identified by their unique closed patterns.
Method:

```
1: O'_1 := O_1; ...; O'_n := O_n;
2: for all T ∈ T, O ∈ O do
3:    if O.closed ⊆ T then
4:       O'.sup := O'.sup − 1;
5:    end if
6: end for
7: for all O'_i ∈ {O'_1, ..., O'_n} (if they exist) do
8:    if O'_i.sup < ms then
9:       Remove O'_i, continue;
10:   end if
11:   for all O'_j ∈ {O'_{i+1}, ..., O'_n} (if they exist) do
12:      if O'_i.sup = O'_j.sup & O'_j.closed ⊂ O'_i.closed then
13:         O'_i.keys := min{K|K ∈ O'_i.keys orK ∈ O'_j.keys}
14:         Remove O'_j;
15:      end if
16:      if O'_i.sup = O'_j.sup & O'_j.closed ⊃ O'_i.closed then
17:         O'_j.keys := min{K|K ∈ O'_i.keys orK ∈ O'_j.keys}
18:         Remove O'_i;
19:      end if
20:   end for
21: end for
    return O'_1, ..., O'_n (if they still exist);
```

Fig. 2. TRUM: a novel algorithm for maintaining frequent patterns after some transactions are removed from the original database

decremental dataset. This shows that TRUM is much more computationally effective, compared to previous works, like [7, 12], whose computational complexity is $O(N_{FP})$, where N_{FP} refers to the number of frequent patterns. This is because $O(|\mathcal{D}_{dec}|) \ll N_{FP}$. (Some implementation techniques are suggested in our full paper http://www.ntu.edu.sg/home5/feng0010/FullPaper.pdf).

4.1 Experimental Studies

Extensive experiments were performed to evaluate the proposed algorithm. TRUM was tested using several benchmark datasets from the *FIMI* Repository, http://fimi.cs.helsinki.fi. Due to space constraints, only the results of $T10I4D100K$, *mushroom* and *gazelle* are presented in this paper. These datasets form a good representative of both synthetic and real datasets.

We varied two parameters in our experiments: minimum support ms and update interval. For each employed ms, we preformed multiple execution of the algorithm, where each execution employed a different update interval. Moreover, the performance of the algorithm varies slightly when different sets of transactions are removed. To have a stable performance measure, for each update interval, 5 random sets of transactions were employed, and the average performance of the algorithm was recorded. The experiments were run on a PC with 2.8GHz processor and 2GB main memory.

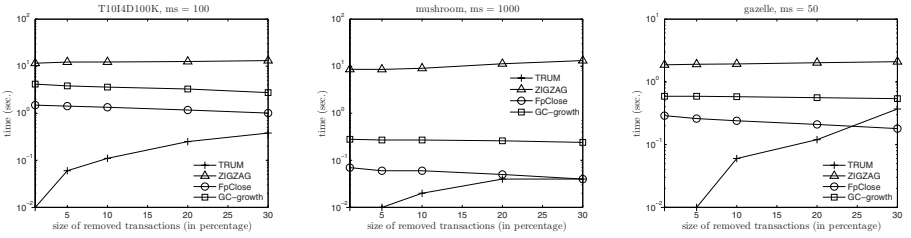


Fig. 3. Average run time comparison of ZIGZAG, FpClose, GC-growth and TRUM

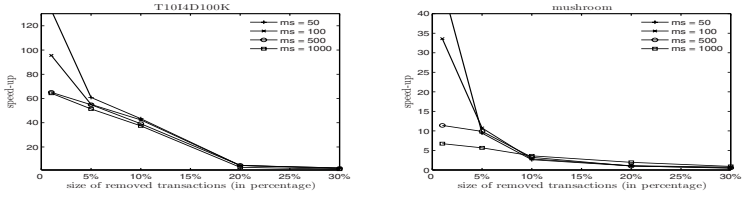


Fig. 4. Speed-up achieved by TRUM against FpClose over various ms thresholds

To justified the effectiveness of the proposed algorithm, we compared its performance against some state-of-art frequent pattern discovery and maintenance algorithms. These algorithms includes ZIGZAG [12], FpClose [8] and GC-growth [10]. Results of the performance comparison is presented in Figure 3.

We observe that TRUM outperforms ZIGZAG by at least an order of magnitude over all update intervals. The advantage of the proposed algorithm is most obvious in *mushroom* dataset. For *mushroom* dataset, TRUM, on average, outperforms ZIGZAG 200 times. It is measured that, for both $T10I4D100K$ and *gazelle*, TRUM achieves around 80 and 20 times average speed-up.

TRUM is also more effective compared to re-discovering all patterns using FpClose and GC-growth. E.g. TRUM is, on average, 30 times faster than FpClose and 100 times faster than GC-growth for $T4I10D100K$ dataset. However, we also observe that as the size of the removed transactions increases, the advantage of TRUM diminishes. This is because, corresponding to the complexity analysis, the execution time of TRUM increases as more transactions are removed. In contrast, due to the shrinkage of data size, the execution time of re-discovery approaches drops when more transactions are removed. Combining these two effects, it is logical that the speed-up gained by our maintenance approach diminishes as the size of removed transactions goes up.

The performance of the proposed algorithm was also evaluated under different support thresholds ms. The results are presented in Figure 4. It demonstrates that TRUM remains effective compared to FpClose over a wide range of minimum support thresholds. Nevertheless, the achieved speed-up drops slightly for higher

ms thresholds. When ms is high, the frequent pattern space becomes smaller, which makes the discovery process much easier. As a result, the advantage of TRUM becomes less obvious.

5 Closing Remarks

This paper has investigated how the space of frequent patterns, equivalence classes, closed and key patterns will evolve when transactions are removed from a given dataset. It was shown that the equivalence classes can evolve in three ways: (1) remain unchanged with the same support value, (2) remain unchanged with decreased support value, and (3) grow by merging with others. Based the evolution analysis, an effective maintenance algorithm TRUM is proposed. TRUM maintains the frequent pattern space using the concept of equivalence classes. TRUM addresses the problem efficiently by updating only the affected equivalence classes. The effectiveness of the proposed algorithm is validated by experimental evaluations.

This paper, to our best knowledge, is the first to study the evolution of frequent pattern space under data updates. The proposed algorithm outperforms the state-of-the-art algorithms at least an order of magnitude over a wide range of support thresholds and update sizes. In the future, it is interesting to exploit the evolution of frequent pattern space under other types of updates, e.g. addition of transaction and items, or removal of items. Solving these maintenance problems with an equivalence class approach could be promising.

References

[1] R. Agrawal, et al. Mining association rules between sets of items in large databases. In *SIGMOD*, pages 207–216, 1993.
[2] Y. Aumann, et al. Borders: An efficient algorithm for association generation in dynamic databases. In *JIIS*, (12) page 61-73, 1999.
[3] R. J. Bayardo. Efficiently mining long patterns from databases. In *SIGMOD*, pages 85–93, 1998.
[4] C. Chang, et al. Enhancing SWF for incremental association mining by itemset maintenance. In *PAKDD*, pages 301–312, 2003.
[5] D. Cheung, et al. Maintenance of discovered association rules in large databases: An incremental update technique. In *ICDE*, pages 106–114, 1996.
[6] D. Cheung, et al. A general incremental technique for maintaining discovered association rules. In *Proc. 1996 DASFAA*, pages 185–194, 1997.
[7] K. Gouda, et al. GenMax: An efficient algorithm for mining maximal frequent itemsets. In *Data Mining and Knowledge Discovery: An International Journal*, 11: 1-20, 2005.
[8] J. Han, et al. Mining frequent patterns without candidates generation. In *SIGMOD*, pages 1–12, 2000.

[9] C. Lee, et al. Sliding window filtering: An efficient method for incremental mining on a time-variant database. *Information Systems*, 30(3):227-244, 2005.
[10] H. Li, et al. Relative risk and odds ratio: A data mining perspective. In *PODS*, pages 368–377, 2005.
[11] N. Pasquier, et al. Discovering frequent closed itemsets for association rules. In *ICDT*, pages 398–416, 1999.
[12] A.A. Veloso, et al. Mining frequent itemsets in evolving databases. In *SIAM*, 2002.
[13] S. Zhang, et al. A decremental algorithm for maintaining frequent itemsets in dynamic databases. In *DaWak*, pages 305–314, 2005.

Establishing Semantic Relationship in Inter-query Learning for Content-Based Image Retrieval Systems

Chun Che Fung and Kien-Ping Chung

School of Information Technology, Murdoch University
South St, Murdoch, Perth, Western Australia, Australia
{k.chung, l.fung}@murdoch.edu.au

Abstract. Use of relevance feedback (RF) in the feature vector model has been one of the most popular approaches for fine tuning query for content-based image retrieval (CBIR) systems. This paper proposes a framework that extends the RF approach to capture the inter-query relationship between current and previous queries. By using the feature vector model, this approach avoids the need of "memorizing" actual retrieval relationship between the actual image indexes and the previous queries. This implies that the approach is more suitable for image database application where images are frequently added or removed. This paper has extended the authors' previous work [1] by applying a semantic structure to connect the previous queries both visually and semantically. In addition, active learning strategy has been used in this paper to explore images that may be semantically similar while visually different.

Keywords: Content-Based Image Retrieval System, Inter-Query Learning, Statistical Discriminant Analysis.

1 Introduction

In the last decade, query tuning using relevance feedback (RF) has gained much attention in the research area of content-based image retrieval (CBIR) systems. This is largely due to RF's ability to refine the user query through a sequence of interactive sessions. Various approaches [2] have been introduced and they have yielded certain degrees of success. However, most research works have focused on query tuning in *a single* retrieval session. This is commonly known as *intra-query* learning. In contrast, *inter-query learning*, also known as *long-term learning,* attempts to analyze the relationship between the current and past retrieval sessions. By accumulating knowledge learned from the previous sessions, inter-query learning aims at further improve the retrieval performance of the current and future sessions. One may view that inter-query is an extension of the intra-query. Although intra-query in CBIR has been a topic of research for the last decade, inter-query in CBIR has only begun to attract interests in the last few years and it is yet to be fully explored.

Previously, the authors have developed an inter-query learning framework based on the statistical discriminant analysis approach to represent the characteristics of a visual group during a retrieval session [1]. Such approach is more suitable for

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 498–506, 2007.

database applications where images are added or removed on a regular basis. It is because that the approach avoids the needs of establishing relationships between each image in the database. This is a common approach used in most inter-query learning frameworks. A weakness with this framework is that it can only merge clusters with similar visual characteristics. The framework is unable to capture the semantic relationship between clusters. Thus, it lacks the capability of establishing relationships between clusters that are semantically similar and yet visually different.

This paper extends the existing framework by introducing a semantic structure to connect clusters that are semantically similar. In addition, active learning strategy is used to explore the semantic structure for the maximum coverage on possible images that are semantically similar to the query image. This paper begins with a discussion on the background of the problem studied in this study. It is then followed by a description of the overall proposed framework. Experiment results are then presented and followed by the conclusion.

2 Problem Background

A feature vector based inter-query learning framework has been proposed in [1]. In the proposed framework, a cluster is formed after each retrieval session. The cluster is described by the feature space created by statistical discriminant analysis and the boundary of the cluster is defined by the furthest positive labeled image from the positive centroid. Since the cluster contains the visual information common to the previously selected positively labeled images, it is assumed that the two retrieval sessions are similar when the majority of the images gathered from the short-term learning algorithm fall within the boundary of a selected cluster. One may view this as a way measuring visual similarity between selected cluster and the short term learning algorithm. Experiments in [1] have shown that the developed framework improves the retrieval accuracy of the system.

At the end of a retrieval session, a cluster merging policy has been proposed. The decision for merging is based on two criteria: based on the measurement of the visual similarity, and, the visual similarity between the two positive centroid points. Table 1 is the summary of the cluster merging policy as proposed previously. From the table, two clusters will only be merged when both are semantically and visually similar. While such strategy is appropriate for condition 3 and 4 where the clusters are not semantically related, it may not be totally suitable for condition 2. In other words, although they are cases that the clusters are not visually similar, they may be semantically related. Such information can be valuable for future retrieval process. Thus, such information should also be recorded by the system.

To resolve the issue with the condition 2, one may apply the same merging algorithm as used in condition 1 but with a more relax merging condition. However, this may be problematic. In statistical discriminant analysis, a visual group is generally captured and represented by a distribution function. A single distribution modal may not be able to capture image samples that are not visually related. Thus, merging of the two clusters which are not visually related may result in losing the visual characteristics of the original clusters.

Table 1. Possible Outcomes of the Merging Policy

	Semantic Similarity	Visual Similarity	Merge Cluster
1	Yes	Yes	Yes
2	Yes	No	No
3	No	Yes	No
4	No	No	No

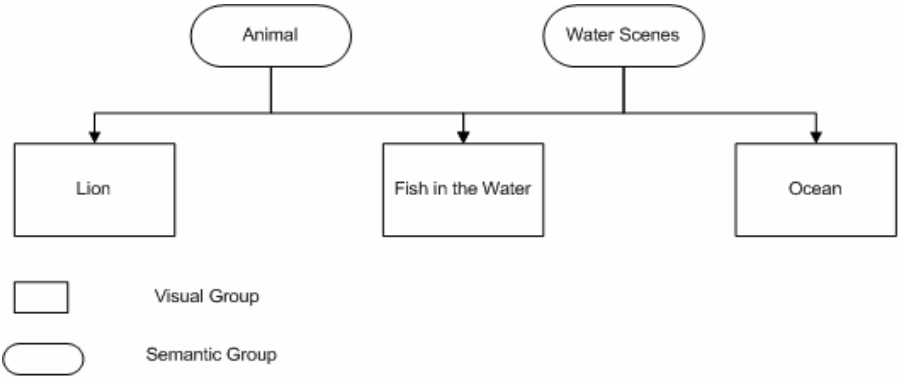


Fig. 1. Example Relationships between different Visual and Semantic Image Groups

One may consider this issue as a multimodal density analysis problem [3, 4]. Such approach is often based on heuristic rules and manual interaction is usually required to set the parameters which are necessary in analyzing the number of distribution modals needed. Moreover, the values of the parameters are often derived through trial and error. Thus, it may not be suitable for generic database. Furthermore, such approach is assuming that a visual group will only belong to a semantic group. This is not necessarily true as a visual group may belong to multiple semantic groups and each semantic group may not be directly related to each other. For instance, Figure 1 shows that while the semantic groups "Animal" and "Water" both contain the visual group "Fish in the Water", but "Animal" may not be directly related to "Water".

Alternatively, one may record the semantic relationship of clusters via a semantic link. The semantic relationship of the two clusters can be determined through the labeled image samples as gathered through user feedback cycles. The use of semantic relationship has the advantage of recording the relationship of the two clusters while preserving the visual characteristics of the clusters. Such approach will be discussed in more detail in the following section.

3 Proposed Framework

3.1 Cluster Merging Scheme

Figure 2 depicts the logical flow of the proposed clustering merging process which an extension of the original framework with the additional semantic link module for

establishing semantic links between the selected clusters. The selected clusters are only visually merged if and only if both clusters are semantically and visually similar. If the two clusters are only semantically similar and yet visually different, then the two clusters will be semantically link by the additional module.

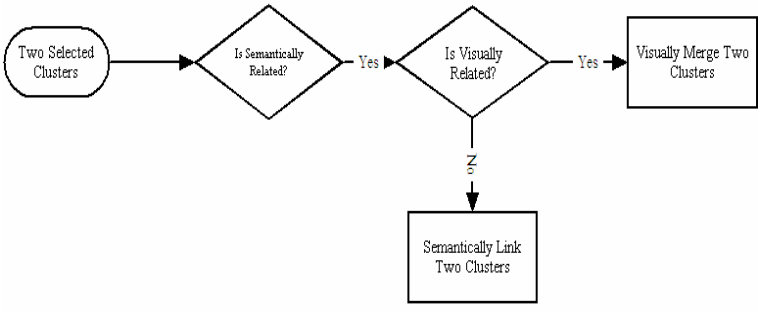


Fig. 2. Proposed Cluster Merging Process

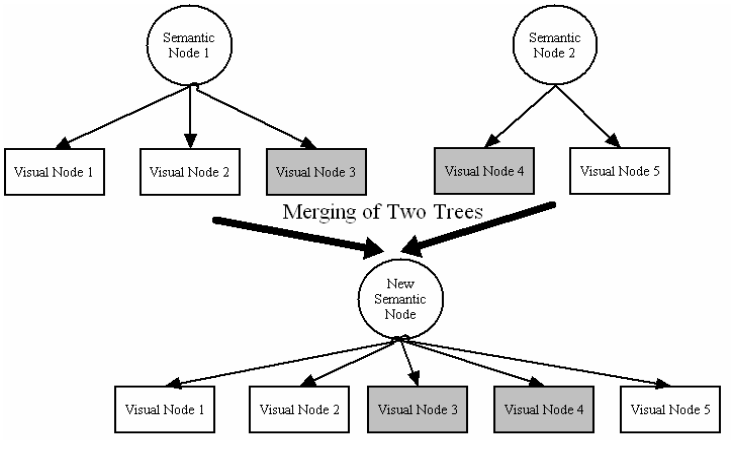


Fig. 3. The Logical View of the Structure of the clusters

In this paper, the proposed semantic structure is represented by a tree hierarchy structure and each cluster acts as a *visual* node in the tree. If a tree contains only one visual node, then the visual node will also be the root node of the tree. A *semantic* node is used if the tree contains more than one visual node. The semantic node is merely a connection node which acts as a connection bridge between all the visual nodes that are semantically related. One may view this as a two layer tree structure framework where the semantic and visual nodes are the root and leaf of the tree respectively.

Figure 3 presents the logical view of a typical semantic tree structure and also, the merging product of the two semantic trees. When two visual nodes are tested to be semantically similar such as visual node 3 and 4 as shown in the figure, the trees

containing the nodes will be merged. The merging process is intuitive. A new merged tree is merely the collection of all the visual nodes under the two original trees. Such relationship implies that all the visual nodes under the same tree are either directly or indirectly related to each other.

3.2 Cluster Search and Explore Schemes

Figure 4 depicts the flow sequence of the proposed clusters searching and exploring scheme of the new semantic framework. The proposed framework is an extension from the existing searching framework with two additional modules. The two additional modules are mainly used for identifying the visual nodes to be explored and the implementation of the visual node exploring strategy. The visual nodes selection strategy is based on the cluster searching criteria as described in [1] and it can be mathematically expressed as:

$$T_p = \frac{N_{n_i}}{N_p} \tag{1}$$

where N_p denotes the total number of positive samples gathered during the feedback cycle, and N_{n_i} is the number of positive samples that fall within the boundary of cluster *i*.

It should be noted that the two additional modules are only activated when no more visual nodes are selected. The system will always explore the visual content first before semantic relationship is considered. There are two reasons to support this design. Firstly, an assumption is made that if certain numbers of positive samples fall within a selected node, then it is very likely that the selected node contains information related to the searched topic. Secondly, it is required to gather as many visual nodes as possible before the system can effectively select visual nodes which are semantically related to the explored nodes. The selection strategy of the visual nodes with related semantic content is described in the following paragraph.

To explore the semantic relationship of the visual nodes, one has to first rank the explored trees. This is based on the fact that a semantic tree consists of visual nodes that are likely to be semantically related to each other. This implies while each visual node is semantically related to each other while they are not necessary interpreted with the same semantic content. Thus, it is possible for the system to explore the wrong node under the same tree. The ranking of the trees is a mechanism designed to minimize chances of exploring visual nodes with different semantic content.

In this paper, it is proposed the ranking of the tree is done by employing a scoring system. The scoring system is based on the ratio of the number of verified visual nodes, N_{V_i}, in a semantic tree, *i*, versus number of visual node, N_{E_i}, previous explored by the system within the same tree. The mathematical expression can be written as:

$$ratio_i = \frac{N_{V_i}}{N_{E_i}} \tag{2}$$

where

$$N_{V_i} \in N_{E_i} \tag{3}$$

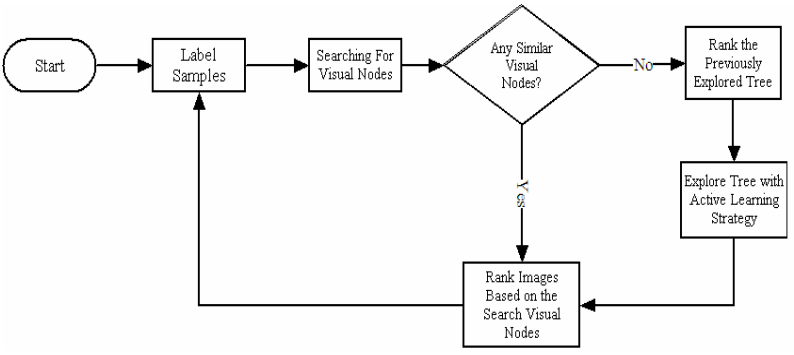


Fig. 4. Flow Diagram of the Proposed Clusters Searching and Exploring Scheme

Such scoring system is used as an indication on the number of nodes that have been explored, and within these nodes, how many have been falsely explored. The minimum score is zero and it implies that all the visual nodes under the selected tree have been falsely explored. Conversely, the maximum score of one indicates that all the nodes within the tree have been correctly explore. Thus, a semantic tree with a higher score will always rank higher than one with a comparatively lower score. For consistency purpose, the searching criterion expressed in (1) is used for node verification. Once the trees have been ranked, the system will then choose the top trees for the exploring of the semantically related nodes. To explore the nodes, one first has to select a node exploring strategy. The traditional *most probable* approach often results in limiting the selection of the images into a narrow region near the query images. This restricts the system for exploring images with different visual characteristics. This is in conflict with the goal of the proposed semantic structure. On the other hand, *active learning* is a strategy with an objective to gather the most informative samples from the given data available. This implies that instead of selecting a narrow region of images, active learning strategy aims to select images in the unexplored regions where the images are possibly semantically related. As for this framework, active learning can be implemented by selecting visual nodes that have the biggest visual differences from the gathered samples. This can be determined by the number of labeled samples that are clustered by the selected visual node. The node with the least clustered samples will be the one with the biggest visual differences.

4 Experiment Results

4.1 Test Environment

In order to test the performance of the proposed approach, three systems have been implemented. They are (i) the proposed semantic framework, (ii) the visual merging framework as proposed in [1], and (iii) a short term learning framework based on KBDA as reported in reference [5]. To evaluate the validity of the experiment, the environment and parameters used by all three systems are identical. The image features and the generalized eigenvector calculation method are the same and the

same parameters are also used in the kernel transformation algorithm for all three systems. In this experiment, five visual features have been selected for the analysis of shape, color and texture of the images. The five features are the water-filling edge histogram algorithm [6], HSV color coherent vector [7], HSV histogram , global edge detection algorithm [8], HSV color moments [9] and color intensity histogram. Each feature comprises a number of elements. A total number of sixty-five feature elements have been used. Lastly, Gaussian Radial Basis Function (RBF) is selected as the kernel transformation matrix for the KBDA approach. This is suggested by literatures [5, 10] as both claimed that RBF yields the best accuracy performance out of all the other kernel transformation approaches.

4.2 Experiment Procedures and Test Data

In this experiment, 500 images of the Corel image database were used. Within these images, 300 images are classified under seven different themes and each consists of several different visual groups. The inter-relationship of each theme is depicted in Figure 5. The themes “bird” and “cat” are subset of “animal”, “fish” is the subset for both the “animal” and “water”, while “water” also comprise of “water scene”. Lastly, “yellow flower” is independent from all the other themes. The inter-relationship between each theme is designed to emulate the complexity of the semantic relationship between each object in the real world.

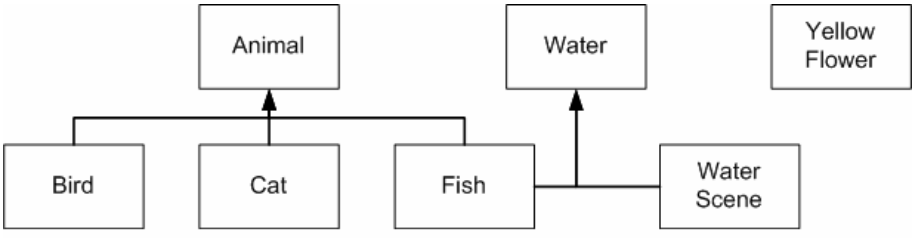


Fig. 5. Relationship between different themes in the Test Data Set

The retrieval performance of the frameworks was measured via three different tests. The tests were generated by randomly selecting 300 positive labeled images from the labeled images as an input entry point to the system. The same data set was then applied to two more tests with different random sequences. The average of the three tests is to ensure the consistency of the test results. The retrieval accuracy is used as the main factor to compare the performance of the systems.

Figure 6 shows the average retrieval accuracies of the three frameworks after three random sequences of 300 search sessions. As shown from the figure, the test framework with the semantic framework is the most effective of the three in terms of retrieval accuracy. With the exception of the theme “yellow flower”, the average retrieval accuracies for the semantic structure are all better than the visual merging as proposed from the previous work. From the figure, it shows that the average retrieval result of the theme “yellow flower” for the visual merging framework is slightly better than the semantic framework. However, the advantage of visual merging framework is only marginal. Furthermore, Table 2 shows that the advantage of the

visual merging scheme on the theme "yellow flower" is inconclusive. Of the three sequences, the performance of the semantic structure in the first sequence was actually better than the visual merging scheme. Thus, one may only conclude that the performance of the two frameworks on "yellow flower" is compatible, neither can claim to be more accurate over the other.

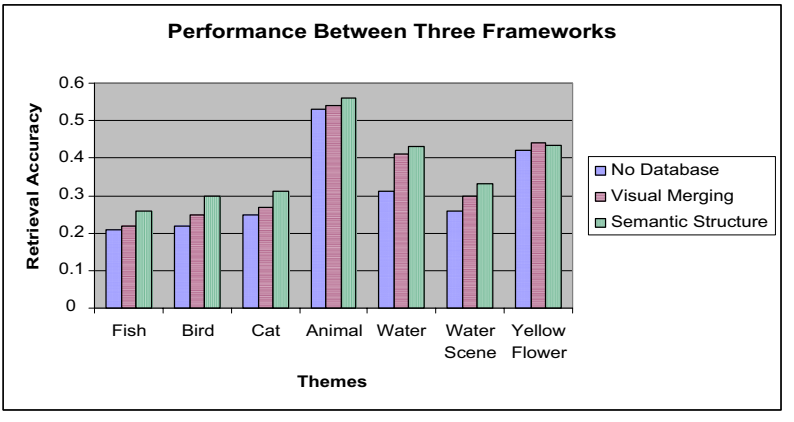


Fig. 6. Retrieval Performance of the Three Frameworks

Table 2. Actual Retrieval Accuracy of the theme "Yellow Flower" for the Two Frameworks

	Visual Merging (Retrieval Accuracy)	Semantic Structure (Retrieval Accuracy)
Random Sequence 1	0.41	0.44
Random Sequence 2	0.45	0.41
Random Sequence 3	0.44	0.41

5 Conclusion

A semantic structure framework for inter-query learning in CBIR system has been introduced. The proposed framework provides the building block for constructing complicated relationship between visual clusters. The complex relationships between different image groups are captured by using a semantic structure to connect different visual image groups that are semantically related. In addition, active learning has also been introduced as the strategy for the selection of visual nodes with related semantic content. The test results have demonstrated that while the retrieval performance between the two frameworks is compatible for simple data sets, the proposed semantic framework is more superior in handling data sets with a more complex relationship. Such framework can be easily modified and expanded by associating keywords to the selected clusters using during the image during retrieval session. The keyword, in turn, may also link to word dictionary database to further improve the rigor of keywords used in the search session. The incorporation of keyword annotation to the user log is one of the future research directions for this study.

Reference

[1] Chung, K.-P., Wong, K. W., and Fung, C. C., Reducing User Log Size in an Inter-Query Learning Content-Based Image Retrieval (CBIR) System with a Cluster Merging Approach, in International Joint Conference on Neural Networks, Vancouver, Canada, 16 - 21 July (2006).

[2] Zhou, X. S. and Huang, T. S., Relevance Feedback in Image Retrieval: A Comprehensive Review, *ACM Multimedia Systems Journal*, vol., 8, (2003), 536-544.

[3] Dong, A. and Bhanu, B., A New Semi-Supervised EM Algorithm for Image Retrieval, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Wisconsin, United States, June 18 - 20 (2003).

[4] Qian, F., Li, M., Zhang, L., Zhang, H. J., and Zhang, B., Gaussian Mixture Model for Relevance Feedback in Image Retrieval, in Proceedings of IEEE International Conference On Multimedia & Expo, Lausanne, Switzerland, August 26-29 (2002).

[5] Zhou, X. S. and Huang, T. S., Small Sample Learning During Multimedia Retrieval using BiasMap, in IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, United States, December (2001).

[6] Zhou, X. S. and Huang, T. S., Edge-based Structural Features for Content-based Image Retrieval, *Pattern Recognition Letters*, vol., 22, (2001), 457 - 468.

[7] Pass, G., Zabih, R., and Miller, J., Comparing Images Using Color Coherence Vectors, in The 4th ACM International Conference on Multimedia, Boston, Massachusetts, United States, November 18-22 (1997).

[8] Park, D. K., Jeon, Y. S., and Won, C. S., Efficient Use of Local Edge Histogram Descriptor, in Proceedings of the 2000 ACM workshops on Multimedia, Los Angles, Calfornia, United States (2000).

[9] Stricker, M. and Orengo, M., Similarity of Color Images, in Storage and Retrieval for Image and Video Databases III, San Diego/La Jolla, CA, USA, February 5-10 (1995).

[10] Wang, L., Chan, K. L., and Xue, P., A Criterion for Optimizing Kernel Parameters in KBDA for Image Retrieval, *IEEE Transactions on Systems, Man, and Cybernetics*, vol., 35, (2005), 556 - 562.

Density-Sensitive Evolutionary Clustering

Maoguo Gong, Licheng Jiao, Ling Wang, and Liefeng Bo

Institute of Intelligent Information Processing, Xidian University,
Xi'an 710071, China
Maoguo_Gong@hotmail.com

Abstract. In this study, we propose a novel evolutionary algorithm-based clustering method, named density-sensitive evolutionary clustering (DSEC). In DSEC, each individual is a sequence of real integer numbers representing the cluster representatives, and each data item is assigned to a cluster representative according to a novel density-sensitive dissimilarity measure which can measure the geodesic distance along the manifold. DSEC searches the optimal cluster representatives from a combinatorial optimization viewpoint using evolutionary algorithm. The experimental results on seven artificial data sets with different manifold structure show that the novel density-sensitive evolutionary clustering algorithm has the ability to identify complex non-convex clusters compared with the K-Means algorithm, a genetic algorithm-based clustering, and a modified K-Means algorithm with the density-sensitive distance metric.

1 Introduction

Many clustering approaches, such as the K-Means Algorithm[1], partition the data set into a specified number of clusters by minimizing certain criteria. Therefore, they can be treated as an optimization problem. As global optimization techniques, Evolutionary algorithms (EAs) have been used for clustering tasks commonly in literature.[2][3][4] The solution representation and dissimilarity measure are the main difficulties in designing EA for clustering. Many researchers have used a representation approach that borrows from the K-Means algorithm: the representation codes for cluster center only, and each data item is subsequently assigned to a cluster representative according to an appointed dissimilarity measure.[5] The most popular dissimilarity measure is the Euclidean distance. By using Euclidean distance as a measure of dissimilarity, these evolutionary clustering methods as well as the K-Means algorithm have a good performance on the data set with compact super-sphere distributions, but tends to fail in the data set organized in more complex and unknown shapes, which indicates that this dissimilarity measure is undesirable when clusters have random distributions. As a result, it is necessary to design a more flexible dissimilarity measure for clustering. Su and Chou [6] proposed a nonmetric measure based on the concept of point symmetry, according to which a symmetry-based version of the K-Means algorithm is given. This algorithm assigns data points to a cluster center if they present a symmetrical structure with respect to the cluster center. Therefore, it is suitable to clustering data sets with clear symmetrical structure. Charalampidis [7] recently developed a dissimilarity measure for directional patterns

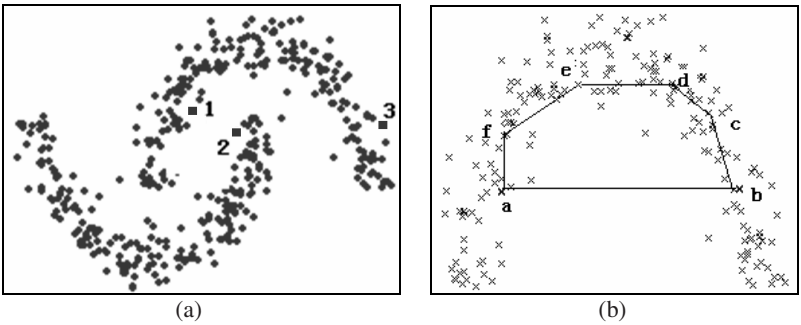
Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 507–514, 2007.

represented by rotation-variant vectors and further introduced a circular K-Means algorithm to cluster vectors containing directional information.

In this study, we propose a novel evolutionary algorithm-based clustering technique, named density-sensitive evolutionary clustering (DSEC), by using a novel representation method and a density-sensitive dissimilarity measure. In DSEC, each string is a sequence of the cluster representatives selected from all the data items. The density-sensitive dissimilarity measure can describe the distribution characteristic of data clustering. The experimental results on seven artificial data sets show that the novel density-sensitive evolutionary clustering algorithm is very suitable to identify complex non-convex clusters compared with the K-Means algorithm [1], a genetic algorithm-based clustering [3], and a modified K-Means algorithm with the density-sensitive distance metric [8].

2 A Novel Density-Sensitive Dissimilarity Measure

For real world problems, the distribution of data points takes on a complex manifold structure, which results in the classical Euclidian distance metric can only reflect the local consistency which refers that data points close in location will have a high affinity, but fail to describe the global consistency which refers that data points locating in the same manifold structure will have a high affinity. We can illustrate this problem by the following example. As shown in Fig. 1(a), we expect that the affinity between point 1 and point 3 are higher than that of point 1 and point 2. In other words, point 1 is much closer to point 3 than to point 2 according to some distance metric. In terms of Euclidian distance metric, however, point 1 is much closer to point 2, thus without reflecting the global consistency. Hence for complicated real world problems, simply using Euclidean distance metric as a dissimilarity measure can not fully reflect the characters of data clustering.



(a) (b)

Fig. 1. (a) An illustration of that the Euclidian distance metric can not reflect the global consistency; (b) An illustration of that the global consistency of clustering does not always satisfy the triangle inequality under the Euclidean metric

Here, we want to design a novel dissimilarity measure with the ability of reflecting both the local and global consistency. As an example, we can observe from the data distribution in Fig. 1(a) that data points in the same cluster tend to lie in a region of

high density, and there exists a region of low density where there are a few data points. We can design a data-dependent dissimilarity measure in terms of that character of local data density.

At first, data points are taken as the nodes V of a weighted undirected graph $G=(V,E)$. Edges $E=\{W_{ij}\}$ reflect the affinity between each pair of data points. We expect to design a dissimilarity measure that ascribes high affinity to two points if they can be linked by a path running along a region of high density, and a low affinity if they cannot. This concept of dissimilarity measure has been shown in experiments to lead to significant improvement in classification accuracy when applied to semi-supervised learning [9][10]. We can illustrate this concept in Fig 1(a), that is, we are looking for a measure of dissimilarity according to which point 1 is closer to point 3 than to point 1. The aim of using this kind of measure is to elongate the paths that cross low density regions, and simultaneously shorten those that not cross.

To formalize this intuitive notion of dissimilarity, we need first define a so-called density adjusted length of line segment. We have found a property that a distance measure describing the global consistency of clustering does not always satisfy the triangle inequality under the Euclidean metric. In other words, a direct connected path between two points is not always the shortest one. As shown in Fig 1(b), to describe the global consistency, it is required that the length of the path connected by shorter edges is smaller than that of the direct connected path, i.e. $\overline{af}+\overline{fe}+\overline{ed}+\overline{dc}+\overline{cb}<\overline{ab}$. Enlightened by this property, we define a density adjusted length of line segment as follows.

Definition 1. The density adjusted length of line segment (x_i,x_j) is defined as

$$L(x_i,x_j)=\rho^{dist(x_i,x_j)}-1 \tag{1}$$

where $dist\left(x_i,x_j\right)$ is the Euclidean distance between x_i and x_j, $\rho>1$ is the flexing factor.

Obviously, this formulation possesses the property mentioned above, thus can be utilized to describe the global consistency. In addition, the length of line segment between two points can be elongated or shortened by adjusting the flexing factor ρ.

According to the density adjusted length of line segment, we can further introduce a new distance metric, called density-sensitive distance metric, which measures the distance between a pair of points by searching for the shortest path in the graph.

Definition 2. Let data points be the nodes of graph $G=(V,E)$, and $p\in V^l$ be a path of length $l=\left|p\right|-1$ connecting the nodes p_1 and $p_{|p|}$, in which $(p_k,p_{k+1})\in E$, $1\le k<\left|p\right|$. Let $\boldsymbol{P}_{i,j}$ denote the set of all paths connecting nodes x_i and x_j. The density-sensitive distance metric between x_i and x_j is defined as

$$D(x_i,x_j)=\min_{p\in\boldsymbol{P}_{i,j}}\sum_{k=1}^{|p|-1}L(p_k,p_{k+1}) \tag{2}$$

Thus $D(x_i, x_j)$ satisfies the four conditions for a metric, i.e. $D(x_i, x_j) = D(x_j, x_i)$; $D(x_i, x_j) \geq 0$; $D(x_i, x_j) \leq D(x_i, x_k) + D(x_k, x_j)$ for all x_i, x_j, x_k; and $D(x_i, x_j) = 0$ if and only if $x_i = x_j$.

As a result, the density-sensitive distance metric can measure the geodesic distance along the manifold, which results in any two points in the same region of high density being connected by a lot of shorter edges while any two points in different regions of high density are connected by a longer edge through a region of low density, thus achieving the aim of elongating the distance among data points in different regions of high density and simultaneously shortening that in the same region of high density. Hence, this distance metric is data-dependent, and can reflect the data character of local density, namely, what is called density-sensitive.

3 Evolutionary Clustering Based on the Density-Sensitive Dissimilarity Measure

3.1 Representation and Operators

In this study, we consider the clustering problem from a combinatorial optimization viewpoint. Each individual is a sequence of real integer numbers representing the sequence number of K cluster representatives. The length of a chromosome is K words, where the first gene represents the first cluster, the second gene represents the second cluster, and so on. As an illustration, let us consider the following example.

Example 1. Let the size of the clustered data set be 100 and the number of clustering being considered be 5. Then the individual (6, 19, 91, 38, 64) represents that the 6-th, 19-th, 91-th, 38-th, and 64-th points are selected to represent the five clusters, respectively.

So this representation method does not mention the data dimension. If the size of the data set is N and the number of clustering is K, then the search space is N^K.

Crossover is a probabilistic process that exchanges information between two parent individuals for generating offspring. In this study, we choose the uniform crossover [11] because it is unbiased with respect to the ordering of genes and can generate any combination of alleles from the two parents.[12][5] An example of the operation of uniform crossover on the encoding employed is shown in example 2.

Example 2. Let the two parent individuals be (6, 19, 91, 38, 64) and (3, 29, 17, 61, 6), random generate the mask (1, 0, 0, 1, 0), then the two offspring after crossover are (6, 29, 17, 38, **64**) and (3, 19, 91, 61, 64). In this case, the first offspring is not (6, 29, 17, 38, **6**) because the 6 in bold is repeat, we keep it unchanged.

Each individual undergoes mutation with probability p_m as example 3.

Example 3. Let the size of the clustered data set be 100 and the number of clustering being considered be 5. Then the individual (6, 19, 91, 38, 64) can mutate to (6, 19+*floor*((100-19)**random*+1), 91, 38, 64) or (6, 19-floor((19-1)**random*+1), 91, 38, 64) equiprobably, where the second gene is selected to mutate, *random* denotes a uniformly distributed random number in the range [0,1), and *floor* denotes rounding towards minus infinity.

3.2 Objective Function

Each point is assigned to the cluster whose density-sensitive distance of its representative to the point is minimum. As an illustration, let us consider the following example.

Example 4. Let the 6-th, 19-th, 91-th, 38-th, and 64-th points represent the five clusters, respectively. For the first point, we compute the density-sensitive distance between it and the 6-th, 19-th, 91-th, 38-th, and 64-th points, respectively. If the density-sensitive distance between the first point and the 38-th point is the minimum one, then the first point is assigned to the cluster represented by the 38-th point. All the points are assigned in the same way.

Subsequently, the objective function is computed as follows:

$$Dev(C) = \sum_{C_k \in C} \sum_{i \in C_k} D(i, \mu_k) \tag{3}$$

where C is the set of all clusters, μ_k is the representative of cluster C_k, and $D(i, \mu_k)$ is the density-sensitive distance between the *i*-th data item of cluster C_k and μ_k.

3.3 Density-Sensitive Evolutionary Clustering Algorithm

The processes of fitness computation, roulette wheel selection with elitism [13], crossover, and mutation are executed for a maximum number of generations $G_{\max}$. The best individual in the last generation provides the solution to the clustering problem.

Algorithm 1. Density-Sensitive Evolutionary Clustering (DSEC)

```
Begin
1.  t=0
2.  random initialize population P(t)
3.  assign all points to clusters according to the density-
    sensitive dissimilarity measure and compute the objective
    function values of P(t)
4.  t=t+1
5.  if t< Gmax
6.     select P(t) from P(t-1)
7.     crossover P(t)
8.     mutate P(t)
9.     go to step 3
10. end if
11. output best and stop
end
```

Fig. 2. Density-Sensitive Evolutionary Clustering

The initial population in step 2 is initialized to *K* randomly generated real integer number in [1, *N*], where *N* is the size of the data set. This process is repeated for each of the *P* chromosomes in the population, where *P* is the size of the population.

4 Experimental Results

In order to validate the clustering performance of DSEC, here we give the experimental results on seven artificial data sets, named Line-blobs, Long1, Size5, Spiral, Square4, Sticks, and Three-circles, with different manifold structure. The distribution of data points in these data sets can be seen in Fig. 3. The results will be compared with the K-Means algorithm (KM)[1], a modified K-Means algorithm using the density-sensitive dissimilarity measure (DSKM)[8], and the genetic algorithm-based clustering technique (GAC) [3]. In all the algorithms, the desired clusters number is set to be known in advance. The parameter settings used for DSEC and GAC in our experimental study are given in Table 1. For DSKM and KM, the maximum iterative number is set to 500, and the stop threshold 1e-10.

Table 1. Parameter settings for DSEC and GAC

Parameter	DSEC	GAC
Maximum Number of generations	100	100
population size	50	50
Crossover probability	0.8	0.8
Mutation probability	0.1	0.1

Clustering quality is evaluated using two external measures, the Adjusted Rand Index [5] and the Clustering Error [8]. The adjusted rand Index returns values in the interval [0, 1] and is to be maximized. The clustering error also returns values in the interval [0, 1] and is to be minimized.

We perform 30 independent runs on each problem. The average results of the two metrics, clustering error and adjusted rand index, are shown in Table 2.

Table 2. Results of DSEC, GAC, DSKM and KM where the results in bold are the best ones

Problem	Clustering Error				Adjusted Rand Index			
	DSEC	GAC	DSKM	KM	DSEC	GAC	DSKM	KM
line-blobs	**0**	0.263	0.132	0.256	**1**	0.399	0.866	0.409
Long1	**0**	0.445	**0**	0.486	**1**	0.011	**1**	0.012
Size5	**0.010**	0.023	0.015	0.024	**0.970**	0.924	0.955	0.920
Spiral	**0**	0.406	**0**	0.408	**1**	0.034	**1**	0.033
Square4	0.065	**0.062**	0.073	0.073	0.835	**0.937**	0.816	0.816
Sticks	**0**	0.277	**0**	0.279	**1**	0.440	**1**	0.504
Three-circles	**0**	0.569	0.055	0.545	**1**	0.033	0.921	0.044

From Table 2, we can see clearly that DSEC did best on six out of the seven problems, while GAC did best only on the Square4 data set. DSKM also obtained the true clustering on three problems. KM and GAC only obtained desired clustering for the two spheroid data sets, i.e. Size5 and Square4. This is due to that the structure of the other five data sets does not satisfy convex distribution. On the other hand, DSEC and DSKM can successfully recognize these complex clusters, which indicate the density-sensitive distance metric are very suitable to measure complicated clustering structure. When comparisons are made between DSEC and DSKM, the two

algorithms can obtained the true clustering on the Long1, Spiral, Sticks in all the 30 runs, but DSKM can not do it on the Line-blobs and Three-circles. Furthermore, for the Size5 and Square4 problems, DSEC did a little better than DSKM in both the clustering error and the adjusted rand index. The main drawback of DSKM is that it has to recalculate the geometrical center of each cluster as the K-Means algorithm after cluster assignment which reducing the ability of reflecting the global consistency. DSEC made up this drawback by evolutionary searching the cluster representatives from a combinatorial optimization viewpoint. In order to show the performance visually, the typical simulation results on the eight data sets obtained from DSEC are shown in Fig. 3.

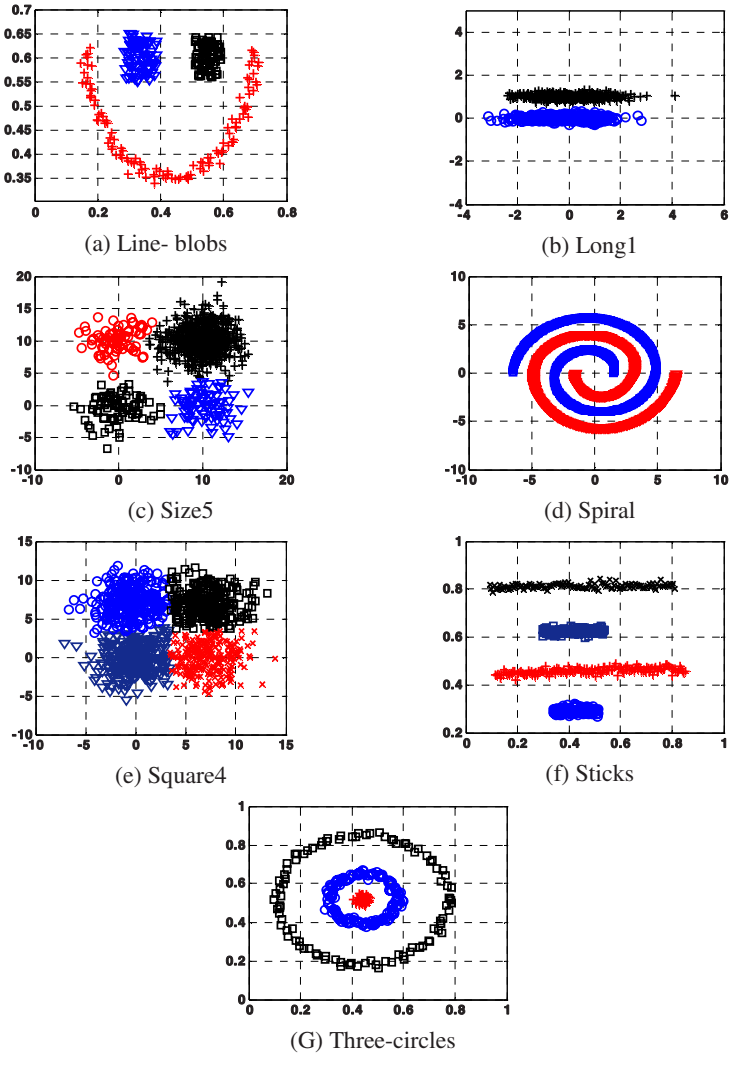


Fig. 3. The typical results on the artificial data sets obtained from DSEC

5 Concluding Remarks

In this paper, we proposed the density-sensitive evolutionary clustering by using a novel representation method and a density-sensitive dissimilarity measure. The experimental results on seven artificial data sets showed that in terms of cluster quality, DSEC outperformed GAC, DSKM and KM in partitioning most of the test problems.

The density-sensitive evolutionary clustering algorithm is a trade-off of flexibility in clustering data with computational complexity. The main computational cost for the flexibility in detecting clusters lies in searching for the shortest path between each pair of data points which makes it slower than KM and GAC.

Acknowledgements. This work was supported by the National High Technology Research and Development Program (863 Program) of China (No. 2006AA01Z107), the National Basic Research Program (973 Program) of China (No. 2006CB705700) and the Graduate Innovation Fund of Xidian University (No. 05004).

References

1. Hartigan, J.A., Wong, M.A.: A K-Means clustering algorithm. Applied Statistics, 28 (1979) 100-108
2. Hall, L.O., Ozyurt, I.B., Bezdek, J.C.: Clustering with a genetically optimized approach. IEEE Transactions on Evolutionary Computation, Vol. 3, No. 2 (1999) 103-112
3. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern Recognition, Vol. 33, No. 9 (2000) 1455-1465
4. Pan, H., Zhu, J., Han, D.: Genetic algorithms applied to multiclass clustering for gene expression data. Genomics, Proteomics & Bioinformatics, Vol. 1, No. 4 (2003) 279-287
5. Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation, Vol. 11 (2007)
6. Su, M.C., Chou, C.H.: A modified version of the K-Means algorithm with a distance based on cluster symmetry. IEEE Transactions on PAMI, Vol. 23, No. 6 (2001) 674-680
7. Charalampidis, D.: A Modified K-Means Algorithm for Circular Invariant Clustering. IEEE Transactions on PAMI, Vol. 27, No. 12 (2005) 1856-1865
8. Wang, L., Bo, L.F., Jiao, L.C.: A modified K-Means clustering with a density-sensitive distance metric. RSKT 2006, Lecture Notes in Computer Science, Vol. 4062. Springer-Verlag, Berlin Heidelberg New York (2006) 544-551
9. Bousquet, O., Chapelle,O., Hein, M.: Measure based regularization. Advances in Neural Information Processing Systems 16 (NIPS), MIT Press, Cambridge, MA (2004)
10. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. Proceedings of the Eighteenth International Conference on Machine Learning (ICML) 18, (2001) 19-26
11. Syswerda, G.: Uniform crossover in genetic algorithms. In: Proceedings of the Third International Conference on Genetic Algorithms. Morgan Kaufmann Publishers, San Francisco, CA (1989) 2-9
12. Whitley, D.: A genetic algorithm tutorial. Statistics and Computing, Vol. 4 (1994) 65-85
13. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, Massachusetts: Addison-Wesley (1989)

Reducing Overfitting in Predicting Intrinsically Unstructured Proteins

Pengfei Han[1], Xiuzhen Zhang[1], Raymond S. Norton[2], and Zhiping Feng[2]

[1] School of Computer Science and IT, RMIT University, Melbourne, VIC 3001, AUS
{phan, zhang}@cs.rmit.edu.au
[2] The Walter and Eliza Hall Institute of Medical Research, Parkville, VIC 3050, AUS
{ray.norton, feng}@wehi.edu.au

Abstract. Intrinsically unstructured or disordered proteins are proteins that lack fixed 3-D structure globally or contain long disordered regions. Predicting disordered regions has attracted significant research recently. In developing a decision tree based disordered region predictor, we note that many previous predictors applying 20 amino acid compositions as training parameter tend to overfit the data. In this paper we propose to alleviate overfitting in prediction of intrinsically unstructured proteins by reducing input parameters. We also compare this approach with the random forest model, which is inherently tolerant to overfitting. Our experiments suggest that reducing 20 amino acid compositions into 4 groups according to amino acid property can reduce the overfitting in decision tree model. Alternatively, ensemble-learning techniques like random forest is inherently more tolerant to this kind of overfitting and can be a promising candidate in disordered region prediction.

Keywords: overfitting, intrinsically unstructured proteins, disordered region, decision tree, random forest, amino acid composition.

1 Introduction

Proteins are linear chains composed of 20 amino acids (also called residues), linked together by polypeptide bonds and folded into complex three-dimensional (3D) structures. Disordered regions (DRs) in protein sequence are structurally flexible and usually have low sequence complexity [1,2,3,4]. Physicochemically, DRs are enriched in charged or polar amino acids, and depleted in hydrophobic amino acids [5,6,7]. Proteins containing DRs are intrinsically unstructured proteins (IUPs) and DR prediction can also be called IUP prediction.

Many computational studies of predicting DRs are based on the biased amino acid composition (AAC) of DRs, which is simple and effective [3,8,9]. However, due to the scarce of disordered training dataset and inevitable noise during physiological experiments, the issue of overfitting [9,10,11] has been raised. The common tackles against overfitting in IUP prediction include attribute selection [9,12] in which only the compositions with the best separating effect are selected; or deriving profile, which is a kind of combination of 20 AAC [11]; or choosing a less overfitting model [13] such as SVM.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 515–522, 2007.

However drawback still exists for some current approaches. Attribute selection may not always improve the IUP prediction accuracy [12]. Checking every combination of whole feature subset might help but it is prohibitively expensive. Less overfitting models like SVM are less accessible to domain scientists.

In our research we consider two approaches alleviating overfitting in IUP prediction. Approach one, we present a system predicts DRs using *reduced AAC* with the decision tree model. It achieves the same effect as pruning and a simplified tree structure is created by limiting the number of input attributes. With the help of domain knowledge, this solution works similar to attribute selection but is easier and computationally more efficient. Prediction accuracy improves and a limited set of rules are produced, which quantifies complex amino acid composition information that is previously unknown. In the second approach, we present a novel application of a recent model in the machine learning field called random forest (RF) [14] in IUP prediction. A special property of RF is that it does not overfit [15,16], which generally is not the case for numerous other machine learning algorithms. Our results demonstrate that random forest performs much more accurate than the decision tree and is able to stand overfitting impact.

2 Decision Tree Based IUP Prediction with AAC

We first describe how the training dataset is constructed in this section. Then we present our decision tree learning and prediction approach.

2.1 Training Data, 20-AAC and Windowing

Different from UCI [17] machine learning repository, IUP prediction has no standard training dataset. In this study, the training datasets come from DisProt (version 2.2) [18] and PDB-Select-25 (the Oct.2004 version) [19]. DisProt is a collection of disordered regions of proteins based on literature description. Only disordered segments of more than 30 residues are extracted, which includes 204 disordered segments and 28386 residues. This disordered training set is called *D-train* hereafter. The ordered training set is extracted from PDB-Select-25, a representative set of protein data bank (PDB) chains that shows less than 25% sequence homology. We selected 366 high-resolution (< 2Å) segments of stable structures which has no missing backbone or side chain coordinates and contains at least 80 residues. This training set includes a total of 80324 residues, and is referred to as *O-train* hereafter.

The windowing technique was introduced in [20], where a sequence of residues including the same number of residues on its both sides predicts for the residue at the center of the window. The AAC in a window is represented by 20 numbers (elements), denoted by n. When a window of w residues slides along a sequence i, the content of the sequence is represented by $n \times (L_i - w + 1)$ elements, where L_i is the length of sequence i. As a result the disordered training segments are represented by $\sum_{i=1}^{204} n \times (L_i - w + 1)$ elements, denoted as *D-M*, and the ordered training segments are represented by $\sum_{i=1}^{366} n \times (L_i - w + 1)$ elements, denoted as *O-M*.

Table 1. Different groups of amino acids

Reduced AAC groups	*Frequency*	*Residues*
Positively charged(P)	F_P	Lys, Arg
Negatively charged(N)	F_N	Asp, Glu
Hydrophobic(H)	F_H	Trp, Phe, Tyr, Leu, Ile, Val, Met
Others(E)	F_E	Ala, Cys, Gly, His, Asn, Pro, Gln, Ser, Thr

2.2 The C4.5 Decision Tree System

There are two different kinds of decision trees: classification and regression trees. C4.5 [21] is a popular classification tree learning system and employed as our tree based IUP predictor.

Given a set T of D(disorder) and O(order) fragments the information content (entropy) for T is $info(T)$. After T has been partitioned into T_1 and T_2 following a test F_i, the information needed to classify T is $info_{F_i}(T)$

The information gain $info(T) - info_{F_i}(T)$ measures the information that is gained by partitioning T with F_i. This gain is normalized by the information generated by the split of T ($split\ info(F_i)$) into ordered and disordered to rectify the bias towards attributes with a large number of values. Finally, the best test to divide a space is the one with the largest gain ratio $\frac{info(T)-info_{F_i}(T)}{split\ info(F_i)}$.

2.3 Overfitting

We found that decision tree and AAC based IUP predictor suffers from overfitting after comparing results of self-test and 10-fold cross validation. The predictor achieves a nearly perfect accuracy in self-test, 99.8%; however 10-fold cross validation decreases dramatically to 76.1%. Meanwhile, around 1100 rules are generated after the training procedure. Many of them involving complicated amino acid relationship have a fairly low usage according to the statistics. Some rules are contradictory to structure biology knowledge.

3 Reduced AAC Decision Trees for IUP Prediction

To tackle overfitting, we generate reduced AAC from training data D-M and O-M. 20 AAC in a window are grouped into four compositions, according to hydrophobicity and polarity properties of amino acid. They are positively charged (P), negatively charged (N), hydrophobic (H) and others (E), as shown in Table 1.

The training and prediction procedure of reduced AAC is the same as that of AAC. Decision tree is constructed from the training data with reduced AAC. After training, every path from the root of a tree to a leaf gives one if-then rule. To classify a query protein sequence, its corresponding reduced AAC is calculated for a given window equivalent to training. Then, starting from the

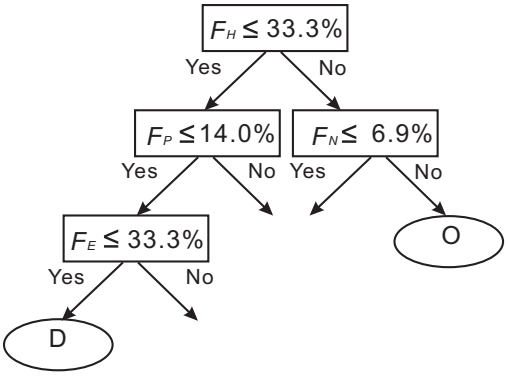


Fig. 1. A sample decision tree. First of all, the root node tests the frequency of hydrophobic residues(F_H) in a window. If it is higher than 33.3%, the frequency of negatively charged residues(F_N) is tested. If this frequency is higher than 6.9%, the central residue within that window is predicted in ordered state; otherwise a set of further tests is performed.

root, the tree node determines which composition has to be checked and what is the residue status. The Figure 1 is a sample decision tree.

As will be discussed in the result and discussion section, prediction accuracy of reduced AAC is significantly higher than that of AAC. Meanwhile, the number of rules has dropped dramatically to 150, which is much easier to be analyzed. The length of rules is shorten and these rules reveal more explicit AAC information.

4 Random Forest Based IUP Prediction

A random forest is an ensemble of unpruned decision trees, where each tree is grown using a subset (bootstrap) of the training dataset [14]. Bootstrap is the training set drawn randomly from original training sets with the same number of training samples. Each tree induced from bootstrap samples grows to full length without pruning and in different from information gain in C4.5, the splitting criterion of random forest is the Gini index.

If a data set T contains D and O fragments, Gini index is defined as $gini(T)$. After T is split into subsets T_1 and T_2 with a test F_i, the gini index of the split data is defined as $gini_{F_i}(T)$. The split provide the smallest $gini_{F_i}(T)$ is chosen to split.

In real implement, there can be a few tens even hundreds of trees. Figure 2 is our development approach based on random forest. The number of trees in the forest is adjustable. To classify a query sequence, each single tree in the forest works similar to the decision tree and gives a residue status either ordered or disordered. As a forest forms with a large number of trees, the final classification having the most votes is chosen.

Random forest provides a reliable estimate of error using the data that is randomly withheld from each iteration of tree development (the "out-of-bag" or

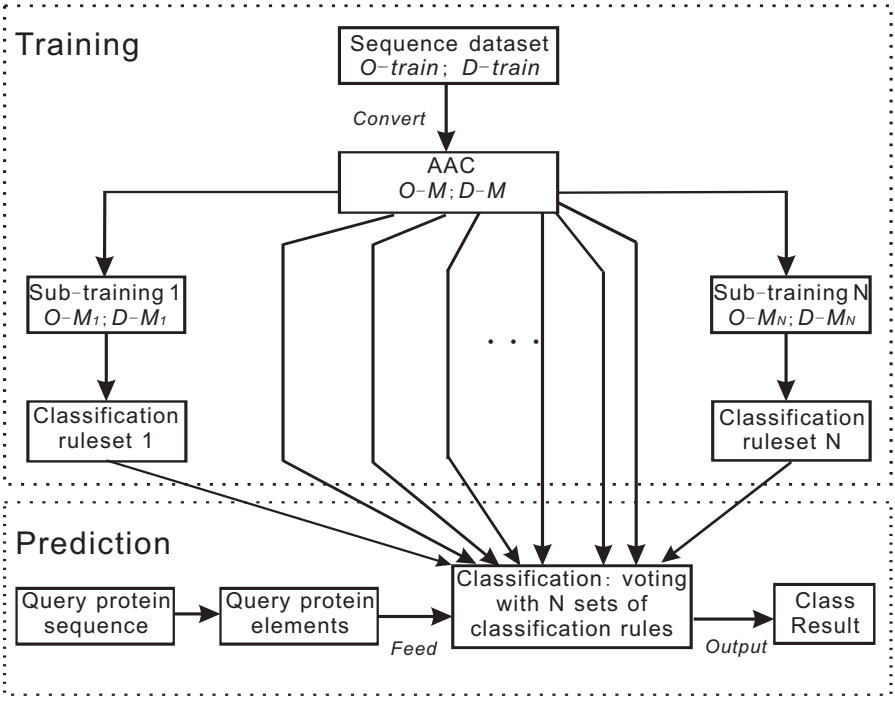


Fig. 2. The diagram for the development of our approach based on random forest

OOB portion). The error rate of a RF decreases as more trees are added until a certain point but will never get larger no matter how many more component predictors are added. Thus employing more trees will not lead to overfitting [15], which is a desirable feature for IUP prediction.

5 Result and Discussion

In this section we compare the overfitting influence on decision tree and random forest based IUP predictors.

5.1 Overfitting Comparison

Figure 3 are the ROC curves for the decision tree and random forest models. The bigger the area under the ROC curve the more precise the predictor is. The two curves on top are OOB test results of the random forest including 50 trees. The other two curves are 10-fold cross validation results of the decision tree. We keep 90% protein sequences to do the training then predict those 10% sequences left. After that we shift the training and predicting sequences until all protein sequences are predicted.

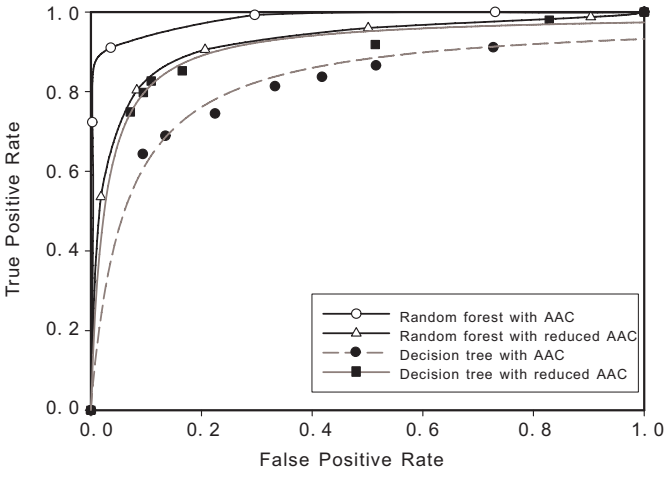


Fig. 3. ROC curves of our system for comparing overfitting in decision tree and random forest. The training window size of random forest and decision tree is 17 and 93 residues respectively.

For the decision tree, smaller windows generate less accuracy results, overall random forest has a much higher ROC curve than the decision tree. The random forest trained on AAC performs around 5% more accurate than that trained on reduced AAC. We also tested the relationship between prediction accuracy and different window size and the number of trees. Generally, with an increasing number of trees, the prediction accuracy improves. This improvement becomes marginal and tends to stabilize at 300 trees. Increasing window sizes also improves accuracy. However big windows have the problem that many residues in the beginning and at the end of the sequence are ignored. Given different window and number of trees, results of OOB test with AAC always perform superior than the RF model with reduced AAC. So there is no obvious overfitting caused by AAC in the RF model.

The result of C4.5 is less accurate compared to random forest. However, the decision tree with reduced AAC totally contains the curve of AAC. With reduced AAC, 10-fold cross validation has improved from 76.1% to around 80%. Given X axis is false positive rate and Y axis is true positive rate, it means reduced AAC makes less mistakes and finds more true DRs during IUP prediction. Grouping strategy makes some compositions originally vague well established and some compositions originally redundant simplified. So overfitting is alleviated by grouping.

Reduced AAC also significantly decreases the number of rules generated by the decision tree model. Our experiments showed that the rule number reduced from 1100 to around 150. Besides by reduced AAC, each rule gets more concise which improves the rule quality, and is much easier to be studied. As a summary, reducing the number of input parameters can be an approach complementing current techniques to avoid overfitting.

6 Conclusion

In this paper we focused on reducing overfitting in IUP prediction. We have demonstrated that overfitting can be reduced by simplifying the input parameters with domain knowledge, rather than complicating the model. Our initial attribute feature AAC demonstrates that overfitting happens which decreases the performance of the predictor. As a simple approach of grouping them according to amino acid physicochemical property, overfitting is assuaged in our decision tree model. Furthermore, with reduced AAC, decision tree based IUP predictor generates significantly less amount of rules, which are simpler and more precise.

Our second approach tackles overfitting by applying ensemble learning. Random forest as a model proven have no overfitting has outstanding accuracy in IUP prediction. With the simple AAC information, it performs much better than decision tree and does not suffer from the overfitting. Apart from the same drawback as other ensemble learning where output is less accessible to domain scientists, random forest is a very suitable tool to be used in comparative proteome studies and protein structure studies.

For future work, we will study if grouping amino acids can also help reducing overfitting in other models for IUP prediction, such as Hidden Markov Model and Support Vector Machine. In addition to AAC we have tested, grouping may also help reduce the overfitting in input like Markov Chain and amino acid replacement matrix [22]. They are all groupable with physicochemical properties.

Acknowledgments. The authors thank Dr Marc Cortese for his explanation of the DisProt database. Z.P. Feng is supported by an APD award from the Australian Research Council.

References

1. J. J. Ward, et al. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *J. Mol. Biol*, 337:635–645, 2004.
2. P. Romero, et al. Sequence complexity of disordered protein. *Proteins: Structure, Function, and Genetics*, 42:38–48, 2001.
3. K. Coeytaux and A. Poupon. Prediction of unfolded segments in a protein sequence based on amino acid composition. *Bioinformatics*, 21:1891–1900, 2005.
4. P. Radivojac, et al. Prediction of boundaries between intrinsically ordered and disordered protein regions. *Pacific Symposium on Biocomputing*, 216–227, 2003.
5. E. A. Weathers, et al. Reduced amino acid alphabet is sufficient to accurately recognize intrinsically disordered protein. *FEBS Lett*, 576:348–352, 2004.
6. J. C. Hansen, et al. Intrinsic protein disorder, amino acid composition, and histone terminal domains. *J. Biol. Chem.*, 281:1853–1856, 2006.
7. V. N. Uversky, et al. Showing your id. *J. Mol. Recognit*, 18:343–84, 2005.
8. Z. Dosztanyi, et al. The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins. *J. Mol. Biol*, 374:827–839, 2005.

9. A. Vullo, et al. Spritz: a server for the prediction of intrinsically disordered regions in protein sequences using kernel machines. *Nucleic Acids Res*, 34:164–168, 2006.
10. T. M. Mitchell. *Machine learning*. McGraw-Hill, 1997.
11. K. Peng, et al. Optimizing long intrinsic disorder predictors with protein evolutionary information. *J. Bioinform Comput Biol*, 3:35–60, 2005.
12. K. Peng, et al. Length-dependent prediction of protein intrinsic disorder. *BMC Bioinformatics*, 7:208, 2006.
13. J. A. Siepen, et al. Beta edge strands in protein structure prediction and aggregation. *Protein Sci*, 12:2348–2359, 2003.
14. L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.
15. J. Oh, et al. Estimating neuronal variable importance with random forest. *Bioengineering Conference, IEEE*, 29:33–34, 2003.
16. W. Bridewell, et al. Reducing overfitting in process model induction. *Twenty-Second International Conference on Machine Learning*, pages 81–88, 2005.
17. C.L. Blake, et al. UCI repository of machine learning databases, 1998.
18. Z. Obradovic, et al. Predicting intrinsic disorder from amino acid sequence. *Proteins: Structure, Function and Bioinformatics*, 53:566–572, 2003.
19. U. Hobohm and C. Sander. Enlarged representative set of protein structures. *Protein Sci*, 3:522, 1994.
20. P. Romero, et al. Sequence data analysis for long disordered regions prediction in the calcineurin family. *Genome Informatics*, 8:110–124, 1997.
21. J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufman Publishers, 1993.
22. M. S. Fornasari, et al. Site-specific amino acid replacement matrices from structurally constrained protein evolution simulations. *Molecular Biology and Evolution*, 19:352–356, 2002.

Temporal Relations Extraction in Mining Hepatitis Data

Tu Bao Ho[1], Canh Hao Nguyen[1], Saori Kawasaki[1],
and Katsuhiko Takabayashi[2]

[1] Japan Advanced Institute of Science and Technology, Ishikawa, 923-1292 Japan
[2] Chiba University Hospital, Chiba, 260-8677 Japan
{bao, canhhao, skawasa}@jaist.ac.jp, takaba@ho.chiba-u.ac.jp

Abstract. Various data mining methods have been developed last few years for hepatitis study using a large temporal and relational database given to the research community. In this work we introduce a novel temporal abstraction method to this study by detecting and exploiting temporal patterns and relations between events in viral hepatitis such as "event A slightly happened before event B and B simultaneously ended with event C". We developed algorithms to first detect significant temporal patterns in temporal sequences and then to identify temporal relations between these temporal patterns. Many findings by data mining methods show to be significant by physician evaluation and match with reported results in Medline.

1 Introduction

Recently, a precious source for hepatitis study has been given by Chiba university hospital to the data mining community [6]. The hepatitis temporal database collected during twenty years (1982-2001) containing results of 771 patients on 983 laboratory tests. It is a large temporal relational database consisting of six tables of which the biggest has 1.6 million records. In last few years, six problems P1-P6 posed by physicians in hepatitis study using the above database have attracted different research groups.

Temporal abstraction (TA) is an approach to temporal pattern detection that aims to derive an abstract description of temporal data by extracting their most relevant features over periods of time [2]. Different from the regular data processed by the other TA methods [3], the hepatitis data was collected irregularly in long periods, and none of the above methods can be applied to.

Our early work [4] developed a supervised TA technique called *abstraction pattern extraction* (APE) whose task is to map (to abstract) a given fixed length sequence into one of predefined abstraction patterns. In this work we develop a unsupervised TA technique called *temporal relation extraction* (TRE) whose task is to find temporal relations in terms of temporal logic [1] among detected temporal patterns, and use these relations together with abstraction patterns to solve problems P1-P2.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 523–530, 2007.

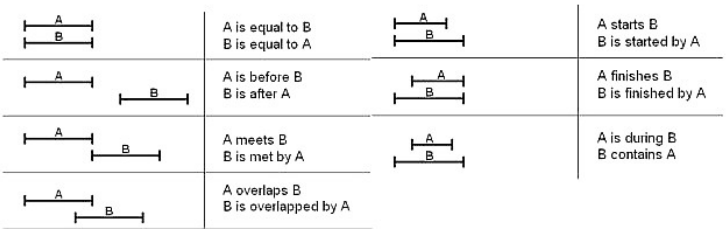


Fig. 1. Temporal relations in Allen's temporal logic

2 Hepatitis Data and Temporal Basic Patterns

Our focus in this work is on problems P1-P2 among six problems posed by physicians to challenge the KDD community [6]: (P1) Discover the differences in temporal patterns between hepatitis B and C (HBV and HCV); (P2) Evaluate whether laboratory tests can be used to estimate the stage of liver fibrosis (LC (liver cirrhosis) vs. nonLC (non liver cirrhosis)).

For each patient O_k the measured values e_i on a medical test A_j over time are an event sequence $S_{jk} = (e_1, t_1), (e_2, t_2), ..., (e_n, t_n)$. In case of the hepatitis data, sequences S_{jk} can be long as observed during twenty years. The starting point of our work is the view on *temporal patterns* that are a rather broad concept and defined differently in temporal data mining. We particularly view temporal patterns in terms of 13 kinds of temporal relations (Figure 1) between two events A and B summarized by Allen in the temporal logic [1]. In [5] a temporal pattern is considered as a set of states together with their interval relationships described in the Allen's interval logic [1]. We consider a temporal pattern as a conjunction/relation of temporal basic patterns (hereafter called basic patterns). In the hepatitis study, we selected 24 typical tests from 983 tests – based on the opinion of physicians and the preprocessing/analysis results of different research groups – divided into two types:

1. *Short-term changed tests (STCT)*: GOT, GPT, TTT, and ZTT that characterize liver inflammation and their values can highly increase in short terms (within several days or weeks) when liver cells are destroyed by inflammation.
2. *Long-term changed tests (LTCT)*: These characterize the liver reserve capacity and change smoothly their values in long terms (within months or years) when their reserve capacity becomes exhausted. They are divided into two subgroups: (a) Going down: T-CHO, CHE, ALB, TP, PLT, WBC, and HG; (b) Going up: D-BIL, I-BIL, T-BIL, and ICG-15.

Basic patterns in STCT sequences: The abstraction states of STCT include N (normal region), H (high), VH(very high), XH (extreme high), L (low), VL (very low), and XL (extreme low). We call a *peak* the event that has its value suddenly much higher than that of its neighbors. We define the *temporal basic patterns (BP)* of a STCT the subsequence characterizing a inflammation period

1. For each object O_k, from the event sequence Sj_k on each attribute A_j, find all possible significant abstracted temporal basic patterns BP on corresponding temporal intervals T.
2. Consider all temporal basic patterns found from all attributes for each object O_k and detect all significant temporal relations between those temporal basic patterns in terms of temporal logic. Represent each object O_k as a graph or a transaction of temporal relations.
3. Using data mining methods to find temporal rules from the collection of graphs or transactions.

Fig. 2. Framework of mining hepatitis data by temporal relation extraction (TRE)

where the sequence suddenly has the high or very high state and with/without peaks. These basic patterns have the form:

$$< state\ of\ test > = \ high\_value \quad \text{or}$$
$$< state\ of\ test > = \ high\_value\ \&\ peaks$$

where $< state\ of\ test >$ denotes the abstraction state of the test sequence and the test name, and $high\_value$ is one value in $\{H, VH, XH\}$. For example, "$GOT = XH\ \&\ peak$" means "GOT is in extremely high state with peaks".

Basic patterns in LTCT sequences: The abstraction states of STCT include N (normal), H (high), L(low). We define the *temporal basic patterns (BP)* of a LTCT the subsequence characterizing the change of states between two state regions. These basic patterns have the form:

$$< state\ of\ test > = \ state_1 \ > \ state_2$$

where $state_1$ and $state_2$ are two different values in {N, H, L} and ">" stands for "change the state to". For example, "$ALB = N > H$", or more informally "$ALB = NormalToHigh$" means "ALB changes from normal to high state".

Denote by (BP, T) a temporal basic pattern BP that occurs in a time interval $T = (t_s, t_e)$ where $(t_s, t_e) = t_1, t_2, ..., t_n$. Examples of temporal basic patterns are "ALB decreases from normal to low state", "GOT has many peaks in very high state". In the context of temporal data, we consider only temporal patterns happening in some period of time, and can implicitly write patterns BP instead of (BP, T). As defined above, temporal patterns viewed as temporal relations between temporal basic patterns are compound statements such as "Pattern A happened before pattern B and B happened during pattern C".

3 Finding Temporal Patterns

This section describes solutions to the problem of finding temporal basic patterns (step 1) and complex temporal patterns in form of temporal relations (step 2) in the framework. The key issue in these steps is that it is hard to determine exactly interval boundaries T in which temporal basic patterns BT occur while determining temporal relations between temporal basic patterns requires comparing their boundaries.

Algorithm 1. Detecting basic patterns in STCT sequences
Input: A sequence Sj_k of a test data from a STCT A_j
Output: Basic patterns characterizing the inflammation in the STCT.

1. Call a data point (e_i, t_i) a peak if $e_i > e_j + threshold$ where (e_j, t_j) is any neighbor of (e_i, t_i).
2. Find the most left peak (e_i, t_i) from the sequence. Set the $CurrentPeak = (e_i, t_i)$, the starting and ending boundaries of the period are $t_s = t_i - 1$ and $t_e = t_i + 1$.
3. Find the closest peak on the right of $CurrentPeak$.
4. If $(t_j < t_e)$ then set $t_e = t_j + 1, CurrentPeak = (e_j, t_j)$ and return to step 3.
5. If $(t_j \geq t_e)$ then
   (a) Calculate the base state BS (without considering peaks) of the interval (t_s, t_e),
   (b) Form the abstracted temporal event "$BS\&P$" in this interval,
   (c) Set a new period with the starting and ending boundaries: $t_s = t_i - 1$ and $t_e = t_j + 1$. Set $CurrentPeak = (e_j, t_j)$ and Return to step 3.

Fig. 3. Finding temporal basic patterns in a STCT sequences

3.1 Finding Basic Patterns

After smoothing data, we detect periods of state changing for both STCT and LTCT based on the following criteria: (a) The first point and last point belong to different states; (b) States of the first point and last point are stable for at least 6 months; (c) Intervals between consecutive crossing pairs must less than parameter θ_1 or intervals between two crossing pairs are less than θ_3 and there are at least $MinPoint$ crossing pairs between them; (d) The interval between two consecutive crossing pairs must be less than θ_2. By the statistics and visualization of the data, together with discussion with physicians, we choose $\theta_1 = 12 \times 4$ weeks, $\theta_2 = 3 \times 12 \times 4$ weeks and $\theta_3 = 5 \times 12 \times 4$ weeks. basic patterns for STCT and Algorithm 2 in Figure 4 is for LTCT.

3.2 Finding Temporal Relations

The step 2 in our framework aims to build a graph or a transaction of possible temporal relations from each object (patient) O_k starting from all of its detected events. A basic algorithm to do this task was originally given in [1] using constraint propagation technique (the transitive property of temporal events). In this work on hepatitis data, due to the specific features of the data, we develop an appropriate technique based on: (1) *Soft matching*: at the boundaries of intervals for relations "equal ", "meet", "start", "finish", and "overlap". The boundary points of two events are considered the same (time) if their absolute difference is smaller than a given threshold, or considered as different in "overlap" relation if

Algorithm 2. Detecting basic patterns in LTCT sequences
Input: A sequence Sj_k of a test data from a LTCT A_j
Output: Basic patterns characterizing the state change periods in the LTCT.

1. Detect crossing:
   If $state(f(t)) \neq state(f(t+1))$ then t is a crossing point.
2. Merging crossing points:
   - If $length(crossing\ point\ i,\ crossing\ point\ i+1) \leq \theta_1$ then merges i and $i+1$.
   - If $length(crossing\ point\ i,\ crossing\ point\ i+1) > \theta_2$ then separate i and $i+1$.
   - If $length(crossing\ point\ i,\ crossing\ point\ i+1) < \theta_3$ and $j - i > n$ then merge i and j.
3. Interval detecting: For each crossing point (an interval of merged crossing points), if it is stable for 6 months before and after, then this crossing point (the interval) is a change period.

Fig. 4. Detecting basic patterns in a LTCT sequences

their absolute difference is greater than a given threshold. (2) *"Slightly"* is a key constraint for the "before" relation, i.e., we consider only relations of the form "A slightly before B" viewed by some threshold.

4 Mining Abstracted Data and Evaluation

Our work follows four steps: (1) Created a transactional database for each hepatitis problem by proposed algorithms described in Section 3; (2) Used software CBA[1], our LUPC[2] and See5 to find rules from the transactional database with default parameters; (3) Filtered statistically significant rules by hypothesis testing; (4) Analyzed the findings with/by physicians.

Rules for hepatitis types HBV and HCV: Using CBA, we were able to generate a set of 238 rules, in which 20 rules for HBV and 218 rules for HCV. The overall accuracy of the prediction rule sets on the training data is 89.34%. Contingency table of the rule set on training data is as follows.

Predicted		HBV	HCV
Correct	HBV	208	30
	HCV	35	337

Table 1 shows the set of typical rules for describing HBV and HCV. We can observe the component test items in the temporal events exhibit different temporal patterns for each of HBV and HCV as follows:

[1] http://www.comp.nus.edu.sg/∼dm2
[2] http://www.jaist.ac.jp/ks/labs/ho/Projects.htm

Observation 1: Even when there are temporal relations between GOT and GPT, even both GOT and GPT have peaks in High region, the rules in which ALP oscillate between Normal and Low are for HBV while the ones in which ALP oscillate between High and Normal are for HCV. Some rules support this observation are the numbers: 145 (ALP changes from Low to Normal etc., class HBV), 206 (ALP changes from Normal to Low etc., class HBV), 20 (ALP changes from High to Normal, class HCV) and 202 (ALP changes from Normal to High etc., class HCV).

Observation 2: Among patients who have peaks on both GPT and TTT in High regions, T-BIL decreases from High to Normal in HBV patients, while T-BIL decreases Normal to Low in HCV patients. Some rules support this observation are the numbers: 196 (class HBV), 185 (class HCV), 203 (class HCV), 167 (class HCV) and 25 (class HCV).

Algorithm 3. Find a transaction or a graph of temporal relations

Input: The set of all associated events to one object O_k

Output: A transaction or graph of temporal relations.

1. To build a transaction
   - Initialize the transaction as an empty set.
   - Check all pairs of events for each temporal relation type. If a pair matches the relation, add this relation to the transaction.
2. To build a graph
   - Build the transaction of relations as in the previous step.
   - Build the graph by adding each existing temporal relation to the graph when considering the events as vertices and relations as edges.

Fig. 5. Finding a transaction/graph of temporal relations

Table 1. Some typical rules for HBV and HCV

RID	Class	Cov.	Conf.	Rule Conditions
145	B	3	100.0%	ALP=LowToNormal & GOT=Normal
206	B	20	80.0%	ALP=NormalToLow & GOT=High *Ends* GPT=High
20	C	13	100.0%	ALP=HighToNormal & GOT=High *Starts* GPT=High
196	B	12	83.3%	T-BIL=HighToNormal & GPT=High *Ends* TTT=High
185	C	7	85.7%	T-BIL=NormalToHigh & GPT=Normal
217	C	139	77.0%	GPT=High *Before* TTT=High & TTT=High *Before* ZTT=High
176	C	10	90.0%	GPT=Normal & TTT=High *Starts* ZTT=High
151	B	3	100.0%	TP=NormalToLow *Before* ZTT=High & TTT=High *Starts* ZTT=High
8	C	18	100.0%	TP=NormalToHigh & TTT=High *Before* ZTT=High
2	C	23	100.0%	TP=HighToNormal & TTT=High *Before* ZTT=High

Table 2. Some typical rules for (non-) liver cirrhosis

RID	Class	Cov.	Conf.	Rule Conditions
1	NonLC	10	100.0%	CRE=NormalToLow & TTT=Normal
2	NonLC	10	100.0%	T-BIL=NormalToLow & LDH=NormalToLow & GOT=High & TTT=High
3	NonLC	10	100.0%	T-BIL=NormalToLow & ZTT=High & LDH=NormalToLow & TTT=High
5	NonLC	8	100.0%	T-BIL=NormalToLow & ALP=NormalToHigh & GOT=High & TTT=High
9	NonLC	7	100.0%	ZTT=High *Before* GPT=High & ALP=NormalToHigh & TTT=High *Before* GPT=High
8	NonLC	7	100.0%	ALP=NormalToHigh *Before* TTT=High & ZTT=High
13	NonLC	6	100.0%	ZTT=High & T-BIL=HighToNormal & GOT=High & TTT=High
26	LC	4	100.0%	I-BIL=HighToNormal & ALB=LowToNormal
27	LC	4	100.0%	TTT=Normal & ALB=LowToNormal
37	LC	3	100.0%	ALB=NormalToLow & LDH=NormalToLow
38	LC	3	100.0%	T-BIL=LowToNormal & ALP=NormalToHigh & TTT=High *Before* GPT=High

Observation 3: Patients who have temporal relations of peaks in both TTT and ZTT have different state change on TP. In case of HCV , TP moves from High to Normal, meanwhile it changes from Normal to Low for HBV. Some rules support this observation are the numbers: 151 (class HBV), 8 (class HCV) and 2 (class HCV).

Matching with Medline abstracts: We looked for some reported results from medical researches to find evidences for and against our findings. We developed a simple search program integrating both keywords and synonyms in the query.

Murawaki et al [7] showed that the main difference between HBV and HCV is that the base state of TTT in HBV is normal, while that of HCV is high. We examined the rule sets and found that our rules are more complicated than that as they also include various temporal relations. However, there are many rules of very high coverage and high confidence, TTT appeared to be mostly in High state for HCV but in Normal state for HBV. We showed some rules support this finding in the table with numbers: 219, 227 226 and 193. This means that even though our rules are not exactly identical to reported knowledge of medical research, such knowledge is confirmed true in our rule set under certain condition.

Rules for liver cirrhosis: LC and non-LC: Using CBA, we were able to generate a set of 61 rules, in which 21 rules for LC and 40 rules for non-LC. The overall accuracy of the prediction rule sets on the training data is 96.30%. Contingency table of the rule set on training data is as follows.

Predicted		LC	non-LC
Correct	LC	37	0
	non-LC	4	67

Some rules in the set can be seen from the Table 2. Notions in the table are identical to that in Table 1. From the rule sets, we observed the following phenomena:

Observation 1: There are more rules for non-LC patients and most of them are of higher precision and coverage. This conforms to the common knowledge of experts that LC is harder to detect.

Observation 2: There were some long term changed test items that appeared mostly in LC patients, namely I-BIL and ALB. The following rules for LC patients support this observation: (Rule 15) I-BIL changes from normal to low (coverage: 5 patients, precision: 100%); (Rule 26) I-BIL changes from high to normal and ALB changes from low to normal (coverage: 4 patients, precision: 100%); (Rule 27) ALB changes from low to normal and TTT has peaks in normal state (coverage: 4 patients, precision 100%). From this, we may induce that I-BIL and ALP change their states mostly in LC patients, not in non-LC ones. They can be good indicators for predicting liver cirrhosis patients.

5 Conclusion

The main contribution of this work is a temporal relation extraction method that allows us to well abstract hepatitis data and discover interesting temporal patterns. It is believed that the temporal relation extraction method, when appropriately combined with numerical conditions or domain knowledge in other formalisms, can be well applied to other medical data mining tasks.

References

1. Allen, J., "Maintaining Knowledge About Temporal Intervals", *Communications of the ACM*, *26(11)*, 832–843, 1983.
2. Balaban, M., Boaz, D., and Shahar, Y., "Applying temporal abstraction in medical information systems", *Annals of mathematics, computing and teleinformatics 1(1)*, 56-64, 2003.
3. Bellazzi, R., Larizza, C., Magni, P., Monntani, S., and Stefanelli, M., "Intelligent Analysis of Clinic Time Series: An Application in the Diabetes Mellitus Domain", *Intelligence in Medicine, 20*, 37–57, 2000.
4. Ho, T.B., Nguyen, T.D., Kawasaki, S., Le, S.Q., Nguyen, D.D., Yokoi, H., Takabayashi, K., "Mining Hepatitis Data with Temporal Abstraction", *ACM International Conference on Knowledge Discovery and Data Mining KDD'03*, 369–377, 2003.
5. Hoppner, F., "Learning dependencies in multivariate times series", *the ECAI'02 Workshop on Knowledge Discovery in (Spatio)-Temporal Data*, 25–31, 2002.
6. http://lisp.vse.cz/challenge/ecmlpkdd2004/
7. Murawaki Y., Ikuta Y., Koda M., Kawasaki H., "Comparison of clinical laboratory liver tests between asymptomatic HBV and HCV carriers with persistently normal amino-transferase serum levels", *Hepatol Research 21(1)*, 67–75, 2001.
8. Sakai H., Horinouchi H., Masada Y., Takeoka S., Ikeda E., Takaori M., Kobayashi K., Tsuchida E., "Metabolism of hemoglobin-vesicles (artificial oxygen carriers) and their influence on organ functions in a rat model" *Biomaterials 25(18)*, 4317–4325, 2004.

Supervised Learning Approach to Optimize Ranking Function for Chinese FAQ-Finder

Guoping Hu[1], Dan Liu[2], Qingfeng Liu[2], and Ren-hua Wang[1]

[1] iFly Speech Lab, University of Science and Technology of China, Hefei, China,
[2] Research of iFlyTEK Co., Ltd., Hefei, China
applecore@ustc.edu

Abstract. In this paper, we address the optimization problem for huge Question-Answer (QA) pairs collection based Chinese FAQ-Finder system. Unlike most published researches which leaned to address word mismatching problem among questions, we focus on more fundamental problem: ranking function, which was always arbitrarily borrowed from traditional document retrieval directly. One unified ranking function with four embedded parameters is proposed and the characteristics of three different fields of QA pair and effects of two different Chinese word segmentation settings are investigated. Experiments on 1,000 question queries and 3.8 million QA pairs show that the unified ranking function can achieve 6.67% promotion beyond TFIDF baseline. One supervised learning approach is also proposed to optimize ranking function by employing 264 features, including part-of-speech, and bigram co-occurrence etc. Experiments show that 7.06% further improvement can be achieved.

Keywords: FAQ-Finder, Ranking Function, Supervised Learning.

1 Introduction

As one of the major approaches for question answering system construction, FAQ-Finder system answers new question by searching in a collection of previously-answered questions. Many researches have been carried out (e. g., [1, 2, 3, 4 and 5]). To build FAQ-Finder system needs consider two problems: 1) how to collect large scale and high quality Question-Answer (QA) pairs; and 2) how to retrieve relevant QA pairs given new question query.

To solve the collection problem, two resources come into researchers' sight: the huge accumulations of FAQ pages on internet (e.g., [2]) and the significantly increased community-based question and answer services on the web where people answer other people's questions (e.g., [3]). For example, the service on http://zhidao.baidu.com has accumulated more than 8 million QA pairs in Chinese with considerable quality, which is the start point of this paper.

For the second retrieval problem, lots of research work has been carried out, but most of them just focused on how to solve the word mismatching problem among questions, including utilizing lexical semantics dictionaries such as WordNet [1], conducting question type analysis [4] and employing machine translation technique [3] etc. Little research work paid attention to the more fundamental problem: ranking

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 531–538, 2007.

function. To our knowledge, only ranking functions such as TFIDF (e.g., [1]) are arbitrarily borrowed from traditional document retrieval without any customizations, which is obviously an unfortunate neglect. Therefore, based on the analysis of differences between FAQ-Finder and traditional document retrieval, we pursue to improve FAQ-Finder system by optimizing the ranking function in this paper.

The rest of the paper is organized as follows: section 2 describes our experimental collection. Unified ranking function approach and supervised learning approach are proposed in section 3 and section 4 separately. Finally, we conclude in section 5.

2 Collections

Baidu is one of the leading commercial search engines in China, and its question and answer service (http://zhidao.baidu.com) is also very popular. Over time, this service has built up a very large archive of QA pairs written in Chinese. The experiments in this paper are based on subsets of this archive (referred as Zhidao hereafter).

2.1 QA Pair Archive and Query Set

Table 1 shows a QA pair example from Zhidao archive. The question part has two fields - question title and question description. Question title is a brief description of user's question, and the question description is an optional field that describes the question title in more detail. The answer part always includes several answers, among which one and only one best answer is marked by the question original inquirer. In our experiment, we only consider the best answer and discard all the other answers. For brief, question title, question description and best answer are referred as *Q*, *D*, and *A* respectively hereafter. For more examples, please refer http://zhidao.baidu.com.

Table 1. A typical QA pair in the Zhidao archive (Translated from Chinese)

Question title	How long can PC keep running?
Question description	I always need download something all night.
Best answer	Now PC is robust enough to keep running for more than one week.
Other answer	PC can keep running but the speed will drop little by little.

We successfully download 3.9 million QA pairs, and split them into two parts according the posting time of each QA pair. The first part contains 3.8 million QA pairs which are posted before May 8, 2006, and the other part contains about 0.1 million QA pairs posted after that date. We utilize the first part as retrieval source in our experiments, referred as *BaiduSet* hereafter. The other part is employed to generate the experimental query set. The purpose of separation according posting time is to simulate searching in the existed archive with new question query. Here we simply employ some word frequency based approach to select 2000 common questions, and employ one assistant to pickup 1000 meaningful questions to build the query set, referred as *QuerySet* hereafter. For brief, the query is referred as *P* hereafter. The average length of *Q*, *D*, *A* and *P* is 31.9, 80.6, 255.2 and 24.7 bytes respectively.

2.2 Relevance Judgment

To verify the performance of each retrieval technique, we first construct one evaluation dataset according following steps:

1) Index *BaiduSet* on the *Q*, *D* and *A* field separately with Lucene.NET [6];
2) For each query *P* in *QuerySet*:
   a) Conduct thrice retrievals on each of above indices;
   b) Pool the top 100, 70 and 30 QA pairs from retrieval results on *Q*, *D* and *A* indices respectively. The result is referred as the *pool* of *P*;
   c) Do manual relevance judgment between *P* and each QA pair in its pool.

In manual relevance judgment, we consider two aspects: whether the question of the QA pair is semantically identical or similar to *P* and whether the *A* field of this pair is useful enough. Six grades of relevance scores are defined, as listed in Table 2.

Table 2. Specification of six grades of relevance scores for QA pair given *P*

Score	Specification
5	Semantically identical question and its answer is just what the inquirer thirsts for
4~3	Similar question and its answer is useful
2~1	Relevant question and its answer contains some relevant information
0	Not relevant question **or** its answer does not contain any useful information

We regard one QA pair is *useful* to *P* iff its relevance score is equal or higher than 3. According to this specification, we employed four assistants to annotate the relevance score for each of the QA pair in all the pools of 1000 queries. Figure 1 presents some distribution information about the relevance scores.

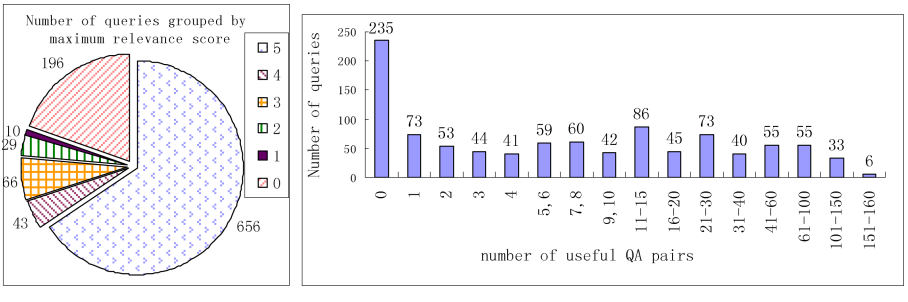


Fig. 1. Distribution of relevance scores. Left and right parts show the number of queries grouped by the maximum relevance score and by the number of useful QA pairs respectively.

2.3 Measure Criterion

Since multiple grades of relevance scores are defined, according to Kazuaki Kishida's work [7], we employ one multi-grades relevance indicator: the precision-oriented *modified sliding ratio* $v_{S'}$, which is defined as following formula:

$$v_{S'} = \sum_{k=1}^{n} \frac{1}{k} x_k \Big/ \sum_{k=1}^{n} \frac{1}{k} y_k \tag{1}$$

where x_k indicates relevance score of the k_{th} QA pair in a ranked pool, and y_k represents the relevance score of k_{th} QA pair in the *ideal* ranking. Average $v_{S'}$ score (referred as $v_{S'}$ too) on *QuerySet* is defined as the measure criterion for each FAQ-Finder system. Because only QA pairs in the pool are manually annotated, for justice, the $v_{S'}$ is only calculated on the QA pairs in each *P*'s pool after they are re-ranked according corresponding approach, regardless of all the other un-annotated QA pairs.

3 Unified Ranking Function Approach

3.1 Characteristics of FAQ-Finder

To retrieve relevant QA pair from millions of QA pairs given new question query is a similar but different task comparing with traditional document retrieval. The differences include:

1) Different query: the query for FAQ-Finder is a natural language question sentence, while for document retrieval the query is just one or several keywords. Therefore if TFIDF or BM25 is borrowed from document retrieval for FAQ-Finder, some extra processings are expected, such as distinguishing the actual keywords from syntactic connecting or auxiliary words;
2) Different corpus to be retrieved from: the corpus for traditionally document retrieval is article or html pages, which always contain at least hundreds of words. But in FAQ-Finder, the corpus is QA pair consisted of 2~3 fields. There are several statistical differences between traditional document and QA pair, such as length distribution and the chasm between query and corpus;
3) Different essential difficulties: for traditional document retrieval, the most difficulty lies in that too much documents which cover all queried keywords can be found and therefore they should be ranked according their popularities or other information. While in FAQ-Finder, query provided in whole sentence may lead to empty result in finding QA pair which contains all the words in query. Therefore how to evaluate the weight of each matched word and how to expand keywords to retrieve more candidates are some key problems for FAQ-Finder.

3.2 Unified Ranking Function

Based on the above analysis, initialized from traditional TFIDF ranking model, we design one unified ranking function for FAQ-Finder as follows:

$$weight(w,F) = \begin{cases} tf(w,F)^{\alpha} \times idf_w^{\beta}, \text{if } idf_w >= idf_{\max} \times \frac{\theta}{2} \\ 0, \text{else} \end{cases} \quad \text{and} \quad idf_w = \log \frac{N}{n(w)+1} + 1 \tag{2}$$

$$unified\_rank\_function(F,P) = \frac{\sum_{w \in F \cap P} weight(w,F) \times weight(w,P)}{|F|^{\gamma}} \tag{3}$$

where, P denotes the query and F denotes one field of QA pair, such as Q, D, and A. $tf(w, F)$ denotes the term frequency of word w in document F. N is the total number of QA pairs and $n(w)$ is the total number of QA pairs that contain w. idf_w is the inversed document frequency of w. $idf_{\max}$ denotes the maximum value of idf among all the words in F. α,β,γ and θ are four embedded parameters which are designed to control four different influences to the ranking function:

1) α controls in what degree we care about the repeated word in F;
2) β controls in what degree we emphasize the word with high idf value;
3) γ controls in what degree we care about the words not found in P but in F;
4) θ controls in what percentage words with lowest idf value are discarded;

The default values for these four parameters are 1.0, 1.0, 1.0 and 0.0 respectively. Note that the unified ranking function will reduce to TFIDF ranking function if all the four parameters are all set as default values. The influences of the four parameters will be investigated later.

3.3 Experimental Settings

To investigate the effect of word segmentation on Chinese FAQ-Finder, we perform two series of experiments: one splits all text (P, Q, D and A) into single Chinese characters, and the other splits all text by employing one word segmentation system consisted of a dictionary of about 60,000 words, word frequency based segmentation disambiguation algorithm, and automatic proper name recognition. The word segmentation system achieves about 97% word segmentation accuracy evaluated on traditional newspapers corpus. These two experimental settings are referred as *Dict_0K* and *Dict_60K* respectively hereafter. And to evaluate each contribution of Q, D, and A field, we carry out experiments on each of the three fields separately.

3.4 Experimental Results

First we investigated the influences of$\alpha,\beta,\gamma$and θ by setting each parameter value as 0.0 to 2.0 with 0.1 step. Six series of experiments were conducted under experimental setting combinations of three fields and two word segmentation settings. Figure 2 is the $v_{S'}$ curves against different values for each parameter. From Figure 2, we can see that it is possible to improve the performance by tuning the four parameters in unified ranking function. By employing *Hill-Climbing* algorithm [8], we optimized four parameters under each experimental setting and evaluated the improvements by 4-folds cross validation. Table 3 shows the improvements.

From the experimental results we can conclude:

1) Retrieval based on Q is significant better than D and A field;
2) Between the two word segmentation settings, different trends can be observed on the three fields of QA pair. Dict_0K contributes more on Q field than Dict_60K, while the trend reverses both on D and A fields;
3) Okapi (just BM25) is a little worse than TFIDF while the Language Model (LM) based retrieval model is the worst;
4) The optimized value of each parameter is quite rational and generalizable: β should be a little higher than 1.0, which means the word with high idf value should be

emphasized in FAQ-Finder;γ should be a little smaller than 1.0, which means we should care but can not care too much about those unmatched words between P and QA pair;θ should be 0.0 or a little higher than 0.0, which indicates that in most case all words in query contribute to FAQ-Finder;α should be less than 1.0 for Q and D, which denotes repeated words should not be considered too much. But in A, the trend is reversed, which means that the term frequency in answer field is useful;

5) Significant improvement (6.67% improvement with p <= 0.000539 under Q and Dict_0K setting) can be achieved.

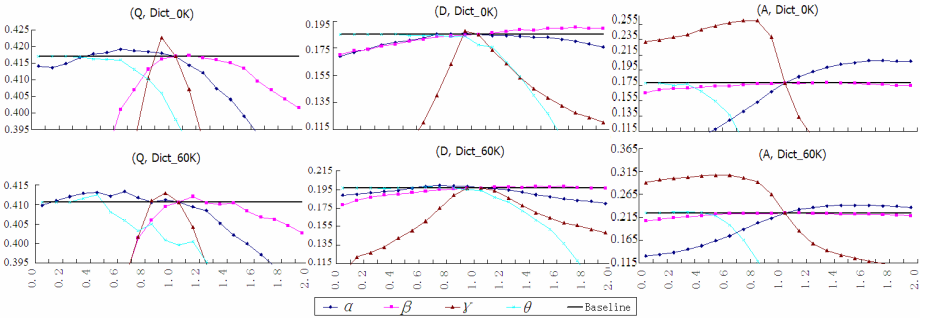


Fig. 2. The curves of $v_{S'}$ on six experimental settings with different α,β,γ and θ values. The title on each small figure denotes the experimental setting.

Table 3. $v_{S'}$ of each ranking function under six experimental settings specified by the first two columns. The "Baseline" column denotes unified ranking function with the four parameters set as their default values (just TFIDF), and the "Optimized" column contains the performances achieved by optimizing four parameters. The last four columns present the value setting for each parameter in formation of minimum~maximum obtained in the 4-folds optimization.

Field	Dict.	Okapi	LM	Baseline	Optimized	α	β	γ	θ
Q	0K	0.3310	0.0333	**0.4170**	**0.4448**	0.3~0.4	1.0~1.0	0.6~0.7	0.0~0.0
Q	60K	0.3589	0.0162	0.4105	0.4270	0.2~0.4	1.0~1.1	0.5~0.7	0.0~0.2
D	0K	0.1056	0.0156	0.1852	0.1377	0.3~0.8	0.9~1.3	0.6~0.9	0.0~0.2
D	60K	0.1574	0.0099	0.1966	0.2019	0.4~0.8	1.2~1.6	0.6~0.8	0.0~0.2
A	0K	0.2310	0.0117	0.2383	0.2658	1.1~1.4	1.0~1.4	0.6~0.8	0.0~0.2
A	60K	0.2958	0.0055	0.3237	0.3373	1.3~1.7	1.0~1.1	0.6~0.7	0.0~0.2

4 Supervised Learning Approach

4.1 Supervised Ranking Function

Intuitively, lots of features can be extracted from QA pair and query P to contribute the ranking procedure in FAQ-Finder, and here we just employ simple linear model to utilize various features. Assume a vector of features is extracted, noted as $\bar{X} = (x_1,...,x_n)^T$, where n is the dimension of feature vector. Then given a vector of feature weights $\bar{W} = (w_1,...,w_n)$, the final ranking function is simply defined as:

$$supervised\_rank\_function(P, QAPair) = \vec{W} \bullet \vec{X} \quad (4)$$

Given the linear model and an annotated training corpus, there are also various supervised learning approaches to optimize the weight vector. Similar to the work of Hu et al. on entity search [8], Hill-Climbing algorithm is employed for training here.

4.2 Features

We totally extracted 264 features for each *P* and QA pair, as shown in Table 4.

Table 4. Features extracted from each *P* and QA pair

Group	ID	Specification
P_Q	0	Unified_rank_score(*P*, *Q*) according formula (4)
Sc	1	The numerator of unified_rank_score(*P*, *Q*) according formula (4)
	2	Number of text string exactly matched word between *P* and *Q*
	3	Feature 2 normalized with the total length of *P* and *Q*
Tyc	4~7	Repeat features 0~3 but substituting exact string matching with semantically matching based on TongYiCiCiLin [9] (similar to [5])
Ed	8~11	Repeat features 0~3 but substituting exact string matching with loose matching based on edit distance
Top	12	Does the word with highest *idf* in *P* occur in *Q*?
	13	Does the content word with highest *idf* in *P* occur in *Q*?
	14~19	Does the word with highest *idf* in *P* occur in *Q*? The part-of-speech of the word is limited in time noun, common noun, and verb etc respectively.
	20	How many quoted words in *P* are found in *Q*?
2Gram	21~32	Repeat features 0~11 by substituting unigram matching with bigram matching
D	33~65	Repeat features 0~32 but substituting *Q* with *D*
A	66~98	Repeat features 0~32 but substituting *Q* with *A*
QDA	99~131	Repeat features 0~32 but substituting *Q* with combined text of *Q, D,* and *A*
Dict_60K	132~263	Repeat features 0~131 but substituting the word segmentation setting as Dict_0K. Features 0~131 are extracted under Dict_0K setting

4.3 Experimental Results

Experimental results (4-folds cross validation) of supervised learning approach are presented in Table 5 and Figure 3. From these experimental results, we can conclude that 1) supervised learning is an efficient approach to utilize various features in Chinese FAQ-Finder, and 2) there truly exist quite a lot of features that can contribute the retrieval performance.

Table 5. Experiment results of supervised learning approach

IDs of Included Features	$v_{S'}$ (improvement)	Sign Test
0 (baseline of retrieval on *Q* field only)	0.4448	
0~32 and 132~164 (all features related to *Q* only)	0.4704 (+5.76%)	p <= 2.82e-14
0,33,66, and 99 (baseline of retrieval on all fields)	0.4803	
0~263 (all features)	**0.5142 (+7.06%)**	p <= 4.39e-11

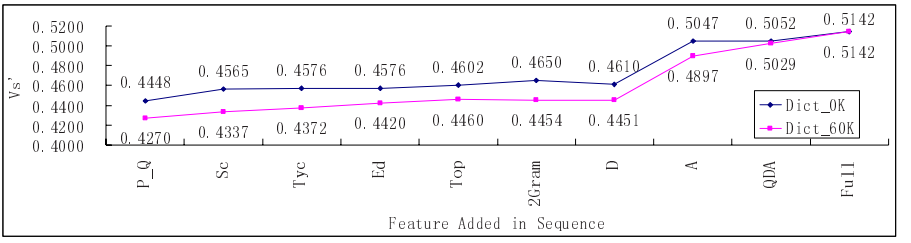


Fig. 3. The curves of $v_{S'}$ against sequently added features under Dict_0K and Dict_60K

5 Conclusion

We construct a Chinese FAQ-Finder system based on 3.8 million QA pairs in this paper. Unlike most published researches which lean to address word mismatching problem among questions, we focus on how to optimize the fundamental ranking function and two approaches are proposed. First we design a unified ranking function with four parameters for Chinese FAQ-Finder which achieves 6.67% (p <= 0.000539) improvement. Second, supervised learning approach together with 264 features extracted from the input query and QA pair are employed to further optimize ranking function, and 7.06% (p <= 4.39e-11) significant improvement is achieved again.

References

1. Robin D. Burke, Kristian J. Hammond, Vladimir A. Kulyukin, Steven L. Lytinen, Noriko Tomuro, Scott Schoenberg: Question Answering from Frequently Asked Question Files: Experiences with the FAQ FINDER System. *AI Magazine* 18(2): 57-66, 1997
2. Valentin Jijkoun, Maarten de Rijke, 2005. Retrieving Answers from Frequently Asked Questions Pages on the Web, *CIKM 2005*: 76-83
3. Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee: Finding similar questions in large question and answer archives. *Proc. of CIKM 2005*: 84-90, 2005
4. Lytinen, S..; Tomuro, N., The Use of Question Types to Match Questions in FAQFinder. *In AAAI 2002 Spring Symposium on Mining Answers From Text* (2002)
5. Che Wanxiang, Liu Ting, Qin Bing, Li Sheng, Chinese Sentence Similarity Computing for Bilingual Sentence Pair Retireval, *JSCL-2003*, (in Chinese), 2003.8
6. Lucene.NET, http://www.dotlucene.net/
7. Kazuaki Kishida, Property of Average Precision and its Generalization: An Examination of Evaluation Indicator for Informaiton Retrieval Experiments, *NII Technical Report*, NII-2005-014E, Oct. 2005
8. Guoping Hu, Jingjing Liu, Yunbo Cao, Hang Li, Jian-Yun Nie, and Jianfeng Gao, A Supervised Learning Approach to Entity Search, *Proc. of AIRS 2006*, 2006
9. Jia-Ju Mei. TongYiCiCiLin (The Thesaurus). *Shanghai Cishu Press*, China, 1983.

Combining Convolution Kernels Defined on Heterogeneous Sub-structures

Minlie Huang and Xiaoyan Zhu

Department of Computer Science and Technology,
Tsinghua University, Beijing, China, 100084
{aihuang, zxy-dcs}@tsinghua.edu.cn

Abstract. *Convolution kernels*, constructed by convolution of sub-kernels defined on sub-structures of composite objects, are widely used in classification, where one important issue is to choose adequate sub-structures, particularly for objects such as trees, graphs, and sequences. In this paper, we study the problem of sub-structure selection for constructing convolution kernels by combining heterogeneous kernels defined on different levels of sub-structures. Sub-kernels defined on different levels of sub-structures are combined together to incorporate their individual strengths because each level of sub-structure reflects its own angle to view the object. Two types of combination, linear and polynomial combination, are investigated. We analyze from the perspective of feature space why combined kernels exhibit potential advantages. Experiments indicate that the method will be helpful for combining kernels defined on arbitrary levels of sub-structures.

Keywords: SVM, convolution kernel, text mining, relation extraction.

1 Introduction

Most of machine learning methods require samples to be represented as feature vectors, with elaborate exploration of features. However, explicit extraction of features has very high computation cost in some cases, for instance, when samples are trees, graphs, or sequences [1]. Kernel functions can directly perform computation in the sample space. Kernel representation has released classifiers from the heavy burden of feature exploration.

For composite objects, the way of defining kernels is to decompose the object into its sub-structures or parts, then to define sub-kernels on their sub-structures, and finally to convolute the sub-kernels. Kernels constructed by this way are termed *convolution kernels* [2]. Most convolution kernels are problem-specific, depending on the means of decomposing of an object into its sub-structures. The key issue is to determine the decomposition and to choose adequate sub-structures of composite objects. The decomposition will be crucial to the performance of learning algorithms because different sub-structures have quite different powers of expressiveness. However, the optimal sub-structures are problem-specific and can only be validated experimentally.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 539–546, 2007.

In this paper, we propose a method to overcome the problem of sub-structure selection by combining heterogeneous kernels defined on different levels of sub-structures. Each level of sub-structure reflects its own angle to view the object and combined kernels can incorporate their individual strengths. Two types of combination, linear and polynomial combination, are studied. From the perspective of feature space, we analyze why combined kernels exhibit potential advantages. Experiments on extracting relations from bioscience texts indicate that the method is helpful for combining kernels defined on arbitrary levels of sub-structures.

The rest of the paper is organized as follows: in Section 2, the background of kernel functions and related work is presented; in Section 3, two means of combination (linear and polynomial) is introduced; in Section 4, the application of extracting relations from bioscience texts is described; in Section 5 comparative experiments are shown. Finally, conclusion is drawn in Section 6.

2 Kernel Function and Its Related Work

A function that calculates the inner product in a feature space is a kernel function. A kernel function defines a mapping function from the sample space X to a feature space F (φ: $X \rightarrow F$), which transforms implicitly a sample into an N-dimensional (N may be infinite) feature vector, as follows:

$$\phi(x) = (\phi_1(x), \cdots, \phi_N(x)) = (\phi_i(x)), \text{ for } i = 1, \cdots, N .$$

Recently, *kernel-based* methods for text mining tasks have been widely studied in machine learning communities. There has been *tree kernel* for parsing, tagging by [3], and relation extraction by [4], *string kernel* or *sequence kernel* for text categorization [5], *path kernel* for relation extraction [6]. These kernels each employ a single sub-structure such as sub-trees or sub-sequences.

However, few efforts have been attended on kernel combination. Joachims et al combined two kernels defined on content and hyper-link information respectively [7]. Lanckriet et al proposed a method for combining kernel representations from multiple data sources in an optimal fashion [8]. Zhao and Grishman presented a kernel-based approach by combining clues from different levels of syntactic processing [9]. However, previous work surveyed here is quite different from ours in that they combine different information sources in nature while our method combines different kernel representations on the same information source. Our combination is made by representing the same information at different granularities.

3 Kernel Combination

In the real world, the same composite object can usually be decomposed into different levels of sub-structures. High levels of sub-structures may be accurate for expressing the information, resulting in a high precision but low recall for classification, while low levels of sub-structures may lead to a high recall but low precision. However, it is generally difficult to determine the optimal sub-structure. If the characteristics of different levels of sub-structures can be integrated together, combined kernels will potentially exhibit advantages over single kernels. For simplicity, in this paper we

only consider the combination of two kernels. However, the methodology is also applicable for combining three or more kernels.

3.1 Linear Combination

Given two kernels K_1 and K_2 defined on object x and y, which are defined on different levels of sub-structures respectively, the linear combination of them is simply defined as follows:

$$\tilde{K}(x, y) = \beta * K_1(x, y) + (1-\beta) * K_2(x, y), \beta \geq 0 . \quad (1)$$

It is easy to prove that if K_1 and K_2 are kernel functions, the combined function is also a kernel function for any non-negative β.

From the definition of kernel function, we know that a kernel function can be represented as a generalized inner-product, as follows:

$$K(x, y) = \sum_i \phi_i(x) * \phi_i(y) . \quad (2)$$

Suppose we have $K_1(x, y) = \sum_i \phi_i(x) * \phi_i(y)$ and $K_2(x, y) = \sum_j \varphi_j(x) * \varphi_j(y)$, the combined kernel defined by Formula (1) can be formulated as below:

$$\tilde{K}(x, y) = \sum_i \sqrt{\beta}\phi_i(x) * \sqrt{\beta}\phi_i(y) + \sum_j \sqrt{1-\beta}\varphi_j(x) * \sqrt{1-\beta}\varphi_j(y) . \quad (3)$$

The feature space of the combined kernel is expanded to the following:

$$\Phi = (\sqrt{\beta}\phi_1, \sqrt{\beta}\phi_2, ..., \sqrt{1-\beta}\varphi_1, \sqrt{1-\beta}\varphi_2, ...) : X \mapsto F . \quad (4)$$

For the perspective of feature space, linear combination exploits a new feature space by a weighted union of the original two feature spaces. Therefore, the combined kernel is potentially superior to each single kernel. This analysis is very intuitive and there is a lack of theoretical proofs. Most convolution kernels are problem-specific and imply implicit feature spaces, making theoretical proof be very difficult. As we have mentioned before, Joachims et al [7] proved the upper bound of errors for different sources of information where they imposed an independence assumption, but this is not applicable for the same information source. The optimization method by Lanckriet et al [8] was also only suitable to different independent sources.

3.2 Polynomial Combination

The polynomial combination of two kernels can be defined as follows:

$$\tilde{K}(x, y) = K_1 + K_2 + \beta * (K_1 + K_2)^2, \beta \geq 0 . \quad (5)$$

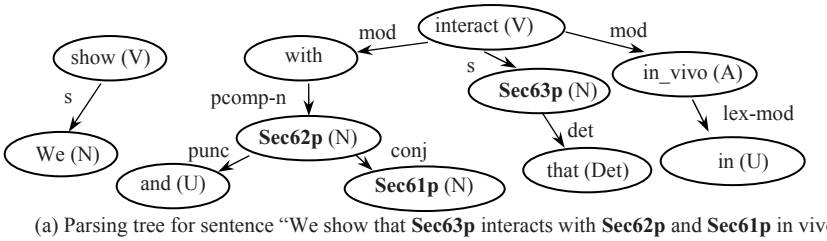
For brevity, we denote $K_1(x,y)$ by K_1, and $K_2(x,y)$ by K_2. Theoretically, more complex polynomial combination is admissible but whether higher power of polynomials will be helpful to improve the performance is yet to be verified by experiments. The combined kernel implies the following feature space:

$$\Phi=(\underbrace{\cdots,\phi_i,\cdots,\varphi_j,\cdots}_{\text{the first order}},\underbrace{\sqrt{\beta}\phi_m\phi_n,\cdots,\sqrt{\beta}\varphi_h\varphi_k\cdots,\sqrt{\beta}\phi_l\varphi_r,\cdots}_{\text{the second order}}):X\mapsto F,\ \forall i,j,m,n,h,k,l,r\cdot \tag{6}$$

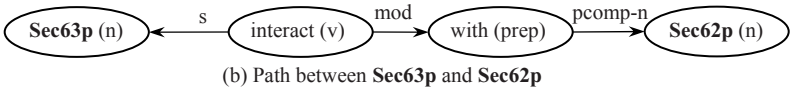
The expanded feature space includes two terms: the simple union of the original two feature spaces and the polynomial multiplication of the original ones. Hence the polynomial combination is potentially better than the linear combination.

4 Kernels for Relation Extraction

The application of the work is to predict whether a relation between two protein entities has been asserted by the dependency path that connects them. A dependency path connecting two entities E_1 and E_2, is a sequence of connected edges: $Path(E_1,E_2):=e_1e_2\ldots e_k$, where $e_i:=N_i \overset{Rel}{\sim} N_{i+1}$ is a dependency edge, and *Rel* is a dependency relation between node N_i and N_{i+1}. The first node (N_1) and last node (N_{k+1}) are the two entities E_1 and E_2 for which we want to extract a relation. Each node is denoted by a triplet (*word*, *base*, *pos*), where *word* is the original form of the word in the sentence, *base* the stemmed form, and *pos* the part-of-speech tag. Fig. 1 illustrates a parsing tree by *MINIPAR* [10], and a dependency path.



(a) Parsing tree for sentence "We show that **Sec63p** interacts with **Sec62p** and **Sec61p** in vivo."



(b) Path between **Sec63p** and **Sec62p**

Fig. 1. An example for dependency parsing tree and path (symbol in parenthesis is *pos*)

The task here is to predict whether a relation between E_1 and E_2 has been asserted by the path connecting them. The following two definitions are important for our later statement:

Def. 1. The *string* of a dependency path $e_1e_2\ldots e_k$ is $SDP:=w_2\ w_3\ \ldots\ w_k$, where words are concatenated by a space, and each w_i is the base form of the i-th node. For example, the string of the first path in Fig. 2 is "interact with".

Def. 2. The *document* of a dependency path $e_1e_2\ldots e_k$ is $DDP:=\{w_2,w_3,\ldots,w_k\}$, where each w_i is the base form of node N_i. For example, the document of the first path in Fig. 2 is {interact, with}.

4.1 Edge-Based Path Kernel (*EPK*)

As the previous definition shows, a dependency path can be viewed as a set of dependency edges. In other words, we may decompose a path into dependency edges. Note that an edge consists of two nodes and a relation. We first define a similarity function on nodes:

$$K_{node}(N_i, N_j) = \begin{cases} 1, & if\ N_i.\text{base} = N_j.\text{base} \\ 0, & others \end{cases}, \tag{7}$$

and then a function on dependency relations:

$$K_{rel}(\text{Rel}_i, \text{Rel}_j) = \begin{cases} 1, & if\ \text{Rel}_i = \text{Rel}_j \\ 0, & others \end{cases}. \tag{8}$$

Obviously, the two functions are both kernel functions. Then the following function can be the kernel for dependency edges $e_i := N_1^i \overset{\text{Rel}_i}{\sim} N_2^i$ and $e_j := N_1^j \overset{\text{Rel}_j}{\sim} N_2^j$:

$$K_E(e_i, e_j) = K_{node}(N_1^i, N_1^j) * K_{rel}(Rel_i, Rel_j) * K_{node}(N_2^i, N_2^j). \tag{9}$$

A path can be viewed as a discrete set of dependency edges. From the theorem presented in [2] (details omitted here), we can define a kernel function for $p_i=\{e_1^i, e_2^i, \cdots, e_m^i\}$ and $p_j=\{e_1^j, e_2^j, \cdots, e_n^j\}$ as follows:

$$K_{path}^E(p_i, p_j) = \sum_{e_j \in p_j} \sum_{e_k \in p_i} K_E(e_k, e_l). \tag{10}$$

This kernel is termed as *edge-based path kernel* (*EPK* for short). Apparently, paths are decomposed into dependency edges. This type of sub-structures is highly accurate, since it captures a large amount of information, including words and dependency link between words.

4.2 Node-Based Path Kernel (*NPK*)

A dependency path can be viewed as a set of nodes or a *document*, as described by Def. 2. Given two paths $p_i=\{N_1^i, N_2^i, \cdots, N_m^i\}$ and $p_j=\{N_1^j, N_2^j, \cdots, N_n^j\}$, we define a node-based path kernel as follows:

$$K_{path}^N(p_i, p_j) = \sum_{N_k^i \in p_i} \sum_{N_l^j \in p_j} K_{node}(N_k^i, N_l^j). \tag{11}$$

We term this kernel *node-based path kernel* (*NPK* for short).

The edge-based path kernel has a higher level of sub-structure than the node-based path kernel because dependency edges not only contain information about nodes, but also reveal dependency link between nodes. Hopefully the edge-based path kernel will offer a better precision since the sub-structure captures more dependency information. The edge-based and node-based path kernel reflects two possible angles viewing paths, and apparently, other types of sub-structures such as sub-strings are applicable.

5 Experimental Results

A benchmark corpus, GENIA corpus [11] is used to extract protein-protein interactions from bioscience literature in our method. Named entities have been previously identified. All relations (or interactions) between proteins are manually annotated. Sentences are firstly parsed by *MINIPAR* and then dependency paths are obtained. Totally there are 3,151 paths, and 1,461 of them are labeled to assert a relation. These paths are randomly partitioned into five parts for five-fold cross-validation. All kernels are incorporated into software package SVM^{light} [12].

5.1 Experiments on Individual Kernels

As defined by Def. 1, a path can be viewed as a *string*. If we take sub-strings as sub-structures of paths, we can define a string kernel as [5]. The best results are obtained when the length of substrings (n) is 3, and the decay factor (λ) is 0.5.

The results of individual kernels are shown in Table 1. The edge-based kernel achieves the best precision because the level of its substructures is the highest, while the string kernel is the worst since it has a low level of sub-structures, which conforms to our previous analysis.

Table 1. Experimental results for *SK*, *EPK* and *NPK*

Kernel	Precision (%)	Recall (%)	F_1 score (%)
SK (String Kernel)	68.30	29.38	41.09
NPK (Node-based Path Kernel)	74.68	35.75	48.35
EPK (Edge-based Path Kernel)	77.84	29.75	43.05

5.2 Experiments on Combined Kernels

We here validate the performance by combining string kernel, edge-based and node-based path kernel. Table 2 shows results for linear combination of the string kernel and edge-based path kernel when different weights are tuned. Table 3 shows results for polynomial combination when different weights are adjusted. In Table 4, we present results of several combined kernels with fixed weights (β=0.5, 0.25 respectively).

From these results, we observe that 1) combined kernels exhibit advantages over individual ones including all baseline kernels; and 2) polynomial combination contributes remarkable improvements over linear combination because the former can offer a more expressive feature space.

5.3 Comparing Combined Kernels with Standard SVM Kernels

In this part we compare combined kernels with traditional standard kernels. A dependency path here is treated as a *document*, and each path is represented as a feature vector such that standard SVM kernels can be calculated.

Table 5 shows the comparative results. Combined kernels outperform standard SVM kernels remarkably in terms of both precision and F_1 score. The comparative results show that our method is promising although standard SVM kernels partially suffer from the sparseness of features since paths may be very short.

Table 2. Linear combination for String Kernel and Edge-based Path Kernel

Weight	Precision (%)	Recall (%)	F_1 score (%)
β=0.0 (K_2=*Edge-based Path Kernel*)	77.84	29.75	43.05
β=0.1	**77.90**	**32.25**	**45.50**
β=0.3	**75.00**	**33.75**	**46.55**
β=0.5	**72.28**	**37.88**	**49.52**
β=0.7	**71.91**	**39.38**	**50.69**
β=0.9	**70.98**	**41.25**	**52.01**
β=1.0 (K_1=*String Kernel*)	68.30	29.38	41.09

Table 3. Polynomial combination for K_1=String Kernel and K_2=Edge-based Path Kernel

Weight	Precision (%)	Recall (%)	F_1 score (%)
β=0.10	68.15	46.86	55.53
β=0.25	**69.35**	**48.75**	**57.25**
β=0.50	72.30	46.64	56.70
β=0.75	74.44	44.86	55.98
β=1.00	76.48	42.35	54.51

Table 4. Comparative results for different kernels with linear and polynomial combination, where $LIN=0.5*K_1+0.5*K_2$ and $POL= K_1+K_2+0.25*(K_1+K_2)^2$

Kernel	Combination type	Precision (%)	Recall (%)	F_1 score (%)
SK	Baseline	68.30	29.38	41.09
NPK	Baseline	74.68	35.75	48.35
EPK	Baseline	77.84	29.75	43.05
K_1=*SK*	*LIN*	72.28	37.88	49.71
K_2=*EPK*	*POL*	**69.35**	**48.75**	**57.25**
K_1=*SK*	*LIN*	71.11	40.63	51.71
K_2=*NPK*	*POL*	**69.98**	**48.25**	**57.12**
K_1=*NPK*	*LIN*	76.23	35.88	48.79
K_2=*EPK*	*POL*	**70.81**	**45.63**	**55.50**

Table 5. Combined Kernels (polynomial) vs. Standard SVM kernels

Kernel	Precision (%)	Recall (%)	F_1 score (%)
Linear Kernel	62.11	44.25	51.68
RBF kernel	75.42	21.72	33.73
Polynomial Kernel	70.65	34.88	46.70
Sigmoid Kernel	55.66	46.50	50.67
SK+EPK	**69.35**	**48.75**	**57.25**
SK+NPK	**69.98**	**48.25**	**57.12**
NPK+EPK	**70.81**	**45.63**	**55.50**

6 Conclusion

In this paper, we have presented a method to construct convolution kernels by combining heterogeneous sub-kernels defined on different levels of sub-structures. Strengths of single kernels are incorporated together by linear or polynomial combination. The problem of substructure selection is avoided because different levels of sub-structures can be integrated together. We also analyze why combined kernels can offer improvements from the perspective of feature space. Our experiments have shown very promising results.

Our future work will be to validate the idea of kernel combination for solving other types of text mining tasks. Also, we will experiment the proposed combined kernels on other corpora designed for bio-text mining.

Acknowledgments. The work was supported by Natural Science Foundation of China under grant No. 60572084, and China 863 Program under No. 2006AA02Z321.

References

1. Scholkopf, B.: Support vector learning. R. Oldenbourg Verlag, 1997.
2. Haussler, D.: Convolution kernels on discrete structures. Technical report, UC Santa Cruz.
3. Collins, M.: New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. ACL 2002: 263-270.
4. Zelenko, D., Aone, C., Richardella, A.: Kernel Methods for Relation Extraction. Journal of Machine Learning Research, 3(2003):1083-1106.
5. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text Classification Using String Kernels. Journal of Machine Learning Research, 2002, 2(2):419-444.
6. Bunescu, R. and Mooney, R.: A Shortest Path Dependency Kernel for Relation Extraction. EMNLP, Vancouver, B.C., pp. 724-731.
7. Joachims, T., Cristianini, N., Shawe-Taylor, J.: *Composite Kernels for Hypertext Categorisation*. In Proceedings of ICML-01, 18th ICML., 250-257.
8. Lanckriet, G., Deng, M., Cristianini, N., Jordan, M.I., Noble, W.S.: Kernel-based Data Fusion and its Application to Protein Function Prediction in Yeast. PSB, 300-311, 2004.
9. Zhao, S. and Grishman R.: Extracting Relations with Integrated Information Using Kernel Methods. ACL 2005, pages 419–426, Ann Arbor, June 2005.
10. Lin, D.: A Dependency-based Method for Evaluating Broad-Coverage Parsers. IJCAI 1995: 1420-1427.
11. Ohta, Y., Tateisi, Y., Kim, J., Mima, H., Tsujii, J.: The GENIA Corpus: An Annotated Research Abstract Corpus in the Molecular Biology Domain. In Human Language Technologies Conference 2002, pp 73-77.
12. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proc. of European Conference on Machine Learning (ECML '98)*, pages 137–142.

Privacy-Preserving Sequential Pattern Release

Huidong Jin[1,⋆], Jie Chen[1], Hongxing He[1], and Christine M. O'Keefe[2]

[1] CSIRO Mathematical and Information Sciences, GPO Box 664,
Canberra ACT 2601, Australia
Huidong.Jin@nicta.com.au, Jiechen@ieee.org, Hongxing.He@csiro.au
[2] CSIRO Preventative Health National Research Flagship,
Canberra ACT 2601, Australia
Christine.OKeefe@csiro.au

Abstract. We investigate situations where releasing frequent *sequential patterns* can compromise individual's privacy. We propose two concrete objectives for privacy protection: k-*anonymity* and α-*dissociation*. The first addresses the problem of inferring patterns with very low support, say, in $[1, k)$. These inferred patterns can become quasi-identifiers in linking attacks. We show that, for all but one definition of support, it is impossible to reliably infer support values for patterns with two or more *negative items* (items which do not occur in a pattern) solely based on frequent sequential patterns. For the remaining definition, we formulate privacy inference channels. α-dissociation handles the problem of high certainty of inferring sensitive attribute values. In order to remove privacy threats w.r.t. the two objectives, we show that we only need to examine pairs of sequential patterns with length difference of 1. We then establish a Privacy Inference Channels Sanitisation (PICS) algorithm. It can, as illustrated by experiments, reduce the privacy disclosure risk carried by frequent sequential patterns with a small computation overhead.

1 Introduction

Data mining poses the dilemma of discovering useful knowledge from databases while avoiding privacy disclosure. There have been various research efforts on privacy-preserving data mining [1,2] from different perspectives such as identification [3], secure computation [1] and sensitive rules [4]. However, little work has been concentrated on removing privacy threats carried by data mining results [5], e.g., sequential patterns. In this work we study how released sequential patterns represent threats to privacy. We will cover sensitive *attribute values disclosure* and *identification disclosure* that focuses on the anonymity of individuals.

Our research motivation is from the healthcare domain where protecting the patients' privacy, such as anonymity and health status, is crucial. In Australia,

⋆ Huidong Jin is currently with National ICT Australia(NICTA), Canberra Lab, Australia. NICTA is funded by the Australian Governments Department of Communications, Information Technology, and the Arts and the Australian Research Council through Backing Australias Ability and the ICT Research Centre of Excellence programs. The authors thank D. Lovell, W. Müller, D. McAullay and anonymous reviewers for their comments and suggestions.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 547–554, 2007.

e.g., the government agency Medicare Australia holds data on drug prescriptions, while each state government holds local hospitalisation data including diagnoses [6]. To enhance healthcare, government agencies could analyse the health events and release knowledge discovered, e.g., frequent sequential patterns.

Example 1. Bob gets 3 sequential patterns from the above healthcare databases:

1. $[a, b, c, d]$ with support 1000, i.e., 1000 patients having a, later on, b, and then c and d, where b (or c, d) may not necessarily be immediately after a (or b, c). a, b, d indicate, say, drugs while c one condition;
2. $[a, b, d]$ with support 1000;
3. $[a, d]$ with support 1001.

These frequent sequential patterns represent a number of individuals as required by the minimum support threshold [7], and seemingly do not compromise privacy. However, these released sequential patterns alone can indirectly divulge privacy including sensitive values and re-identification. (1) From the first two patterns, Bob easily infers that if a patient took Drugs a, b and then d, he/she certainly suffered Condition c, which can be sensitive like HIV. This is risky if one party, say, Medicare Australia or a third commercial insurance company, holds prescriptions only. (2) Based on Patterns 2 and 3, Bob knows that *one and only one* patient has a and then d but without b in between. Through linkage with other data sources, this patient can be re-identified. This results in privacy leakage via linking attacks [8,2].

To protect privacy while releasing sequential patterns and their frequency information, in Section 2, we will propose two new concrete privacy-preserving objectives: (1) *k-anonymous sequential patterns* from which one impossibly infers the existence of patterns with very low support; (2) *α-dissociative sequential patterns* from which one impossibly infers an attribute value with very high certainty. They can serve as a standard of releasing frequent sequential patterns without undue privacy divulgence. We analyse and formulate privacy disclosure inference channels for the two objectives in Section 3. In Section 4, we develop an algorithm PICS (Privacy Inference Channel Sanitisation) to detect and remove these possible privacy threats by deliberately incrementing support values of released frequent sequential patterns. With small distortion, these frequent sequential patterns can be released without undue privacy divulgence w.r.t. the two objectives. We conclude the work in Section 5.

2 k-Anonymous and α-Dissociative Sequential Patterns

We first brief some definitions related to sequential patterns. Let $\mathcal{E}=\{e_1, e_2, \cdots, e_d\}$ be a set of d items. We call a subset $A \subseteq \mathcal{E}$ an ***itemset*** and $|A|$ the ***size*** of A. A ***sequence*** $S=\langle A_1, A_2, \cdots, A_m\rangle$ is an ordered list of itemsets, where $A_i \subseteq \mathcal{E}$, $i \in \{1, \cdots, m\}$. The ***size***, m, of a sequence is the number of itemsets in the sequence, i.e., $|S|=m$. The ***length*** of a sequence $S=\langle A_1, \cdots, A_m\rangle$ is defined as $\mathsf{L}(S)=\sum_{i=1}^{m} |A_i|$. A sequence $S_a=\langle A_1, \cdots, A_n\rangle$ is ***contained*** in another sequence $S_b=\langle B_1, ..., B_m\rangle$

if there exist integers $1 \leq i_1 < i_2 < \cdots < i_n \leq m$ such that $A_1 \subseteq B_{i_1}, \cdots, A_n \subseteq B_{i_n}$. We denote $S_a \Subset S_b$, e.g., $\langle a(bc)a \rangle \Subset \langle ab(abc)(ad) \rangle$. For simplicity, we use $\langle (ab) \rangle$ to indicate a transaction where Items a and b occur at the same time, and $\langle ab \rangle$ to indicate that Item b is preceded by a. A *sequence database* $\mathcal{D}$ is a set of sequences with transactions from $\mathcal{E}$.

A ***pattern*** is an ordered list of itemsets from $\mathcal{E}$, where items can be negative $\overline{e}_i$. A negative item $\overline{e}_i$ means that Item e_i surely does not occur. In Pattern $\left[a\overline{b}(cd)\right]$, e.g., b doesn't occur between a and (cd). A ***sequential pattern*** is an ordered list of itemsets from $\mathcal{E}$ without negative one. Patterns have similar operations with sequences, though they will be placed between '[' and ']', instead of '⟨' and '⟩', e.g., Sequence $\langle ab(cdf)g \rangle$ contains Pattern $[a(cd)]$ but not $\left[a\overline{b}(cd)\right]$.

The ***support*** of a sequential pattern P in $\mathcal{D}$ is defined as the number of sequences that contain P, i.e., $\mathsf{supp}_{\mathcal{D}}(P) = |\{S \,|P \Subset S, S \in \mathcal{D}\}|$. If $P_1 \Subset P_2$, we have $\mathsf{supp}_{\mathcal{D}}(P_1) \geq \mathsf{supp}_{\mathcal{D}}(P_2)$. Given a support threshold θ_s, a sequential pattern is called a ***frequent sequential pattern*** if its support $\mathsf{supp}_{\mathcal{D}}(P)$ is not less than θ_s, i.e., $\mathsf{supp}_{\mathcal{D}}(P) \geq \theta_s$. The problem of mining sequential patterns is to find all *frequent sequential patterns* for a sequence database $\mathcal{D}$, given θ_s. We denote all the frequent sequential patterns as $\mathsf{FSP}(\mathcal{D}, \theta_s)$, and all frequent sequential patterns with length l as $\mathsf{FSP}^{(l)}(\mathcal{D}, \theta_s)$. We omit $\mathcal{D}$ if it is clear from the context.

We now define concrete privacy protection objectives for releasing frequent sequential patterns to end users. To simplify the discussion, we assume these patterns are generated without privacy disclosure, say, in a secure environment. A frequent sequential pattern with its support value can be regarded as a select query that returns the size of a set of sequences containing the pattern. From this viewpoint, we can adapt the concept of k-anonymity [8] from data to patterns in a straightforward way.

Definition 1. Given a small integer threshold $k(> 1)$, a set of frequent sequential patterns from $\mathcal{D}$ are called **k-anonymous sequential patterns** if it is impossible to identify a pattern P such that $0 < \mathsf{supp}_{\mathcal{D}}(P) < k$. Pattern P is **non-k-anonymous** if $0 < \mathsf{supp}_{\mathcal{D}}(P) < k$.

A non-k-anonymous pattern may be used to identify a set of sequences of cardinality greater than 0 and less than k. It may serve as a quasi-identifier for linking attack [2], e.g., $\mathsf{supp}_{\mathcal{D}}\left(\left[a\overline{b}d\right]\right) = 1$ in Example 1. We assume the support threshold $\theta_s > k$, i.e., a single frequent sequential pattern is not non-k-anonymous.

Besides violating anonymity, there is another possibility of releasing sensitive information. As in Example 1, if one patient contains $[abd]$, it is 100% sure that he/she suffers Condition c between taking b and d. Condition c could be sensitive.

Definition 2. Give a rational α (slightly smaller than 1.0), a set of frequent sequential patterns are ***α-dissociative sequential patterns*** if it is impossible to identify a pair of patterns P_1 and P_2 such that

$$P_1 \neq P_2,\ P_1 \Subset P_2,\ \frac{\mathsf{supp}(P_2)}{\mathsf{supp}(P_1)} \geq \alpha. \tag{1}$$

Using non-α-dissociative patterns, for some individuals, we can use the existence of sub-pattern P_1 to infer the existence of super-pattern P_2 that may contain sensitive information with high certainty, say, $\geq \alpha$.

The parameters k and α can be set at any level, depending on the amount of protection that is desired. Thus, if a set of frequent sequential patterns are k-anonymous and α-dissociative, it is acceptable for releasing them from these two concrete privacy-preserving perspectives.

3 Privacy Inference Channels

We now study the possibility of inferring non-k-anonymous or non-α-dissociative patterns from the set of frequent sequential patterns $\mathsf{FSP}(\mathcal{D}, \theta_s)$. A *privacy inference channel* indicates a subset of frequent sequential patterns from which it is probable to infer sensitive information such as non-k-anonymous or non-α-dissociative patterns. Based on the anti-monotonicity of frequent sequential patterns [7], we have the following theorem.

Theorem 1. If $\mathsf{FSP}(\mathcal{D}, \theta_s)$ is not α-dissociative, there must exist a pair of patterns P_t and P_s such that $\mathsf{L}(P_s) = \mathsf{L}(P_t) - 1$, $P_s \Subset P_t$, $\frac{\mathsf{supp}(P_t)}{\mathsf{supp}(P_s)} \geq \alpha$.

Proof: If it is incorrect, i.e., for each pair of P_t and P_s, if $\mathsf{L}(P_s)=\mathsf{L}(P_t)-1$ and $P_s \Subset P_t$, then $\frac{\mathsf{supp}(P_t)}{\mathsf{supp}(P_s)}<\alpha$. Since $\mathsf{FSP}(\mathcal{D}, \theta_s)$ is not α-dissociative, there exists a pair of patterns P_1 and P_2 satisfying Equation 1. For any $P_1 \Subset P_2$, $P_2 \in \mathsf{FSP}(\mathcal{D}, \theta_s)$, there exists a list of frequent sequential patterns, $\{P_{t_1}, P_{t_2}, \cdots, P_{t_j}\}$ ($j \triangleq \mathsf{L}(P_2) - \mathsf{L}(P_1)+1$, i.e., j is defined to be $\mathsf{L}(P_2)-\mathsf{L}(P_1)+1$) such that $P_{t_1}=P_1$, $P_{t_j}=P_2$, $\mathsf{L}(P_{t_i})=i-1+\mathsf{L}(P_1)$, and $P_{t_i} \Subset P_{t_{i+1}}$ for $i=1,2,\cdots,j-1$. We get a contradiction.

$$\frac{\mathsf{supp}(P_2)}{\mathsf{supp}(P_1)} = \frac{\mathsf{supp}(P_{t_j})}{\mathsf{supp}(P_{t_{j-1}})} \frac{\mathsf{supp}(P_{t_{j-1}})}{\mathsf{supp}(P_{t_{j-2}})} \cdots \frac{\mathsf{supp}(P_{t_2})}{\mathsf{supp}(P_{t_1})} < \alpha \times \alpha \cdots \times \alpha < \alpha. \qquad \blacksquare$$

Theorem 1 implies that, to detect whether there exist privacy inference channels for α-dissociation, we only need to compare the support values of pairs of frequent sequential patterns with length difference of 1.

As for support values of patterns with negative items, it is intuitive to define one for a pattern with one negative item. For example, $\mathsf{supp}_{\mathcal{D}}([e_{i_1}\overline{e}_{i_2}e_{i_3}]) \triangleq \mathsf{supp}_{\mathcal{D}}([e_{i_1}e_{i_3}]) - \mathsf{supp}_{\mathcal{D}}([e_{i_1}e_{i_2}e_{i_3}])$, where $1 \leq i_1, i_2, i_3 \leq d$. However, it is not correct to extend this inference channel to patterns with two or more negative items based on the inclusion-exclusion principle [3]. We will further show that there are not reliable inference channels for all but one definition (i.e., Definition 4) of support for patterns with more than one negative item.

We first illustrate this on Pattern $[\overline{e}_{i_1}e_{i_2}\overline{e}_{i_3}]$. To define whether a sequence supports it, we may take account of *every* e_{i_2} or *at least one* e_{i_2} satisfying the pattern. We may consider *both* 'no Item e_{i_1} preceding Item e_{i_2}' and 'no Item e_{i_3} following Item e_{i_2}' are valid or *either* of them. Thus,

Definition 3. There are four possible ways to define support of $[\overline{e}_{i_1}e_{i_2}\overline{e}_{i_3}]$:

3.1 A sequence S is defined to **support** the pattern $[\overline{e}_{i_1}e_{i_2}\overline{e}_{i_3}]$, i.e., $\mathsf{supp}_S([\overline{e}_{i_1}e_{i_2}\overline{e}_{i_3}]) = 1$, if there exists such an item e_{i_2} in S that it is *not preceded by* e_{i_1} or *not followed by* e_{i_3}.

3.2 A sequence S is defined to **support** the pattern $[\overline{e}_{i_1} e_{i_2} \overline{e}_{i_3}]$ if in S there exists e_{i_2} that is *not preceded by* e_{i_1} and *not followed by* e_{i_3}.

3.3 A sequence S is defined to **support** the pattern $[\overline{e}_{i_1} e_{i_2} \overline{e}_{i_3}]$ if any e_{i_2} in S is *not preceded by* e_{i_1} and *not followed by* e_{i_3}.

3.4 A sequence S is defined to **support** the pattern $[\overline{e}_{i_1} e_{i_2} \overline{e}_{i_3}]$ if any e_{i_2} in S is *not preceded by* e_{i_1} or *not followed by* e_{i_3}.

We will extend Definition 3.4 to Definition 4 for any pattern P with multiple negative items to discuss its privacy inference channels. The follow theorem indicates there are no reliable inference channels w.r.t. Definitions 3.1-3.3.

Theorem 2. There are no reliable inference channels based on the frequent sequential patterns in $\mathsf{FSP}(\mathcal{D}, \theta_s)$ to evaluate whether the pattern $[\overline{e}_{i_1} e_{i_2} \overline{e}_{i_3}]$ is k-anonymous or not w.r.t. Definition 3.1 (or Definition 3.2 or 3.3).

The basic idea of the proof is that we can create two sequence sets that have same frequent sequential patterns but much difference on infrequent sequential patterns. One underlying observation is that the ordering is crucial in sequential patterns. a sequence $\langle e_{i_2} e_{i_1} e_{i_3} e_{i_2} \rangle$, e.g., containing $[e_{i_1} e_{i_2}]$ and $[e_{i_2} e_{i_3}]$, does not necessarily contain $[e_{i_1} e_{i_2} e_{i_3}]$. For other patterns with two or more negative items, we can similarly show there are no privacy inference channels for all but one definition of support. The proof is omitted due to space limitation.

We introduce two patterns related to P. Let the lower bound sequential pattern $P^{(l)} \Subset P$ consist of only positive items, and the upper bound one $P^{(u)}$ is generated by transferring all negative items in P into positive ones. If $P_1 = [e_{i_1} \overline{e}_{i_2} (e_{i_3} \overline{e}_{i_4} e_{i_5})]$, e.g., $P_1^{(l)} = [e_{i_1} (e_{i_3} e_{i_5})]$ and $P_1^{(u)} = [e_{i_1} e_{i_2} (e_{i_3} e_{i_4} e_{i_5})]$.

Definition 4. Sequence S is defined to **support** Pattern P with at least a negative item, i.e., $\mathsf{supp}_S(P) = 1$, if it supports its lower bound sequential pattern $P^{(l)}$ while does not support its upper bound one $P^{(u)}$. The support of any pattern with negative item(s) is then defined as.

$$\mathsf{supp}_{\mathcal{D}}(P) = \mathsf{supp}_{\mathcal{D}}(P^{(l)}) - \mathsf{supp}_{\mathcal{D}}(P^{(u)}). \tag{2}$$

So, if both $P^{(u)}$ and $P^{(l)}$ are frequent, there may be a privacy inference channel for P, say, violating k-anonymity. Similar to Theorem 1, to detect and then remove non-k-anonymity privacy inference channels of a whole set of frequent sequential patterns, we only need to examine the support values of those patterns with one and only one negative item due to the following property.

Theorem 3. If there are two patterns P_1 and P_2 such that $P_1^{(l)} \Subset P_2^{(l)}$ and $P_2^{(u)} \Subset P_1^{(u)}$, then, w.r.t. Definition 4,

$$\mathsf{supp}_{\mathcal{D}}(P_1) \geq \mathsf{supp}_{\mathcal{D}}(P_2). \tag{3}$$

The proof is easy because $\mathsf{supp}_{\mathcal{D}}(P_1^{(l)}) \geq \mathsf{supp}_{\mathcal{D}}(P_2^{(l)})$ and $\mathsf{supp}_{\mathcal{D}}(P_2^{(u)}) \geq \mathsf{supp}_{\mathcal{D}}(P_1^{(u)})$. Then, simply based on Equation 2, we get Equation 3. Therefore, the search space for channel detection is reduced immensely, and our detection algorithm for privacy inference channels can be very efficient.

Algorithm 1. Privacy Inference Channel Sanitisation (**PICS**)

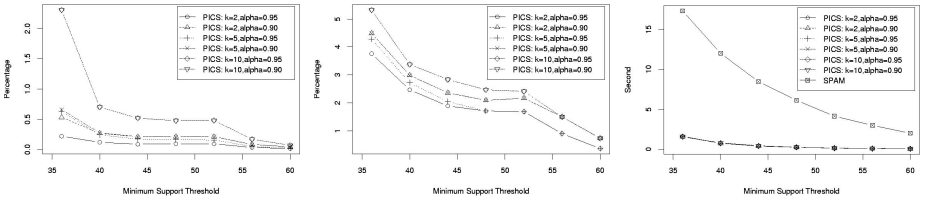
Input: $k(1 < k < \theta_s)$, $\alpha(0 < \alpha < 1)$, $\mathsf{FSP}(\mathcal{D}, \theta_s)$.

1. **for** each $P_i \in \mathsf{FSP}(\mathcal{D}, \theta_s)$ **do** $P_i.\delta = 0$; /*Initialise, say, increments as 0*/
2. $i = l \triangleq \max_{P \in \mathsf{FSP}(\mathcal{D}, \theta_s)} \{\mathsf{L}(P)\}$; /*Start with longest sequential patterns*/
3. **while** $(i{-}{-}) > 0$ **do** /*Go through patterns with length decreased 1 by 1*/
4. **for** each $P_i \in \mathsf{FSP}^{(i)}(\mathcal{D}, \theta_s)$ **do**
5. **for** each $P_j \in \mathsf{FSP}^{(i+1)}(\mathcal{D}, \theta_s)$ such that $P_i \Subset P_j$ **do**
6. $P_i.\delta = P_i.\delta + P_j.\delta$; /*Maintain database-compatibility*/
7. **for** each $P_j \in \mathsf{FSP}^{(i+1)}(\mathcal{D}, \theta_s)$ such that $P_i \Subset P_j$ **do**
8. $K = \max\left\{k, \left\lfloor \frac{1-\alpha}{\alpha}(P_j.\delta + P_j.\mathsf{supp}) \right\rfloor + 1\right\}$;
9. $\delta = (P_i.\delta + P_i.\mathsf{supp}) - (P_j.\delta + P_j.\mathsf{supp})$;
10. **if** $\delta < K$ then $P_i.\delta = P_i.\delta + K - \delta$; /*Detect & remove inference channels*/
11. **Output** $\mathcal{O}_{k,\alpha}$ consists of P_i with support $(P_i.\mathsf{supp} + P_i.\delta)$ for each $P_i \in \mathsf{FSP}(\mathcal{D}, \theta_s)$.

4 Blocking Inference and Experimental Results

Our privacy-preserving released frequent sequential patterns $\mathsf{FSP}(\mathcal{D}, \theta_s)$ are achieved by adjusting their support values to remove privacy inference channels discussed in Section 3 and meanwhile maintain database compatibility. The basic idea is to create a pseudo sequence database $\mathcal{D}_p$ by inserting some sequences into $\mathcal{D}$ such that (1) $\mathsf{FSP}(\mathcal{D}_p, \theta_s)$ contains the same frequent patterns as $\mathsf{FSP}(\mathcal{D}, \theta_s)$ but with a bit different support values and (2) there are no privacy inference channels from $\mathsf{FSP}(\mathcal{D}_p, \theta_s)$. Then we release $\mathsf{FSP}(\mathcal{D}_p, \theta_s)$ instead of $\mathsf{FSP}(\mathcal{D}, \theta_s)$. To keep the result accurate, we insert as few sequences as possible. For any inference channel related to non-α-dissociation as in Equation 1, we simply increment the support of the sub-pattern P_1 by K_1. To maintain database compatibility, the support values of all its sub-patterns are increased by K_1 accordingly. It change $\mathsf{FSP}(\mathcal{D}, \theta_s)$ into $\mathsf{FSP}(\mathcal{D}_1, \theta_s)$ where $\mathcal{D}_1$ is $\mathcal{D}$ plus K_1 copies of sequences only containing P_1. $K_1{=}\max\left\{0, \left\lfloor \frac{\mathsf{supp}(P_2)}{\alpha} - \mathsf{supp}(P_1) \right\rfloor + 1\right\}$ is minimal to ensure that the new support value of P_1 is greater than $\frac{\mathsf{supp}(P_2)}{\alpha}$. For any non-$k$-anonymity inference channel as in Equation 2, we increment the support values of the lower bound frequent pattern $P^{(l)}$ and all its sub-patterns by K_2. It looks like inserting K_2 copies of $P^{(l)}$s. $K_2{=}\max\left\{0, k - (\mathsf{supp}(P^{(l)}) - \mathsf{supp}(P^{(u)}))\right\}$ is minimal to ensure that the new support value of $P^{(l)}$ is not less than $\mathsf{supp}(P^{(u)}) + k$. Thus, incrementing the support values in this way is equivalent to inserting sequences into the original $\mathcal{D}$, and thus database-compatibility is maintained. Moreover, it will not create new inference channels.

According to Theorems 1 and 3, we only need to examine and remove the inference channels caused by pairs of frequent sequential patterns with length difference of 1. Based on these, we propose our PICS (Privacy Inference Channel Sanitisation) algorithm for sanitising frequent sequential patterns as in Algorithm 1. Here $P_i.\mathsf{supp} \triangleq \mathsf{supp}_{\mathcal{D}}(P_i)$ is the original support value for the pattern P_i in $\mathcal{D}$, and $P_i.\delta$ is the support increment introduced by the sanitisation procedure. Basically,



(a) Average distortion ratio. (b) Fraction of support adjusted. (c) Execution time of PICS and SPAM.

Fig. 1. Performance of PICS on the first sequence database with different settings

we start from frequent sequential patterns P_i with the maximal length one by one until with the shortest patterns. For given parameters k and α, we check whether there exists $P_j \in \mathsf{FSP}^{(|P_i|+1)}(\mathcal{D}, \theta_s)$ such that $P_i \Subset P_j$, $\delta < K$ where $\delta \triangleq (P_i.\mathsf{supp}+P_i.\delta)-(P_j.\mathsf{supp}+P_j.\delta)$ and $K \triangleq \max\{k, \lfloor \frac{1-\alpha}{\alpha}(P_j.\delta+P_j.\mathsf{supp}\rfloor+1\}$. We then increase $P_i.\delta$ by $K-\delta$ (i.e., $\max\{K_1, K_2\}$ mentioned above) to ensure the new support difference between P_i and P_j is K. This is embedded in Line 10. In addition, we increase all the sub-patterns of P_i by $K-\delta$. This is implemented in Line 6. The adjusted support for each P_i is $P_i.\mathsf{supp} + P_i.\delta$ as in Line 6. Lines 4-6 ensure the sanitised support values satisfy anti-monotonicity, i.e., if $P_i \Subset P_j$, then $P_i.\mathsf{supp} + P_i.\delta \geq P_j.\mathsf{supp} + P_j.\delta$. There is a sequence database $\mathcal{D}_p$ whose frequent sequential patterns are exactly the output from PICS, i.e., $\mathsf{FSP}(\mathcal{D}_p, \theta_s) = \mathcal{O}_{k,\alpha}$ for given k and α.

We chose freeware SPAM [7] to generate frequent sequential patterns for a sequence database. We implemented PICS in Python. Two sequence databases were generated by IBM Quest Market-Basket Synthetic Data Generator [7]. The first has 20,000 sequences, and 10 different items. The second has 8,000 different sequences and 12 different items. Since PICS ensures the resulting frequent sequential patterns are k-anonymous and α-dissociative, we only measure how much the sanitised support values differ from the original ones. Three metrics are used to evaluate the distortion to support values. (1) **Average distortion ratio** $\frac{\sum_{P \in \mathsf{FSP}(\mathcal{D},\theta_s)} \frac{P.\delta}{P.\mathsf{supp}}}{|\mathsf{FSP}(\mathcal{D},\theta_s)|} \times 100\%$ is the average increment to the original support of sequential patterns. (2) **Fraction of support adjusted** $\frac{|\{P | P \in \mathsf{FSP}(\mathcal{D},\theta_s), P.\delta > 0\}|}{|\mathsf{FSP}(\mathcal{D},\theta_s)|} \times 100\%$ is how often support values are incremented. (3) **The execution time** of PICS is compared with that of SPAM.

A series of experimental results of PICS with different k and α on the first sequence database are illustrated in Fig. 1. The support threshold θ_s is 36, 40, $\cdots$, or 60; k is 2, 5 or 10; and α is 0.90% or 0.95%. Clearly, the execution overhead of PICS is quite small in comparison with SPAM as in Fig. 1(c). When $\theta_s = 40$, e.g., SPAM takes 12.11 seconds while PICS takes 1.08 seconds, only 8.92% of SPAM. PICS is quite conservative for low value of k and $(1-\alpha)$. Typically, when $\theta_s = 40$, SPAM generates 770 frequent sequential patterns. PICS adjusts less than 4% support values as in Fig. 1(b), and the average distortion ratio is less than 0.72% as in Fig. 1(a). The average distortion ratio and fraction of support adjusted appear

to decrease with θ_s with all the settings of k and α. In addition, when k is quite small, say, 2, the average distortion ratio (or the fraction of support adjusted) with $\alpha = 0.95$ is much smaller than that with $\alpha = 0.90$. When k is quite large, say, 10, α values have little influence during the sanitisation procedure. This clearly illustrates that k-anonymity and α-dissociation are two complementary privacy-preserving objectives. Similar performance is observed for the second sequence database. Thus, with a small computation overhead, PICS maintains reasonably good accuracy w.r.t. the original sequential patterns while resulting in k-anonymous and α-dissociative frequent sequential patterns.

5 Conclusions

In this paper, to reduce the privacy disclosure risk caused by releasing frequent sequential patterns, we have introduced two complementary privacy-preserving objectives: k-anonymity and α-dissociation. They address identification and attribute value disclosure respectively. We have established a practical algorithm PICS to detect and remove all the privacy inference channels with respect to both the objectives. After incrementing support values of a small proportion of frequent sequential patterns, PICS can effectively and efficiently sanitise frequent sequential patterns for privacy-preserving release, as substantiated by a series of experimental results. We are studying possible privacy disclosure caused by releasing different types of data mining results but from the same database.

References

1. Vaidya, J., Clifton, C.: Privacy-preserving data mining: Why, how, and when. IEEE Security & Privacy **2** (2004) 19–27
2. Wong, R., Li, J., Fu, A., Wang, K.: (alpha,k)-anonymity: An enhanced k-anonymity model for privacy-preserving data publishing. In: KDD'06. (2006) 754–759
3. Atzori, M., Bonchi, F., Giannotti, F., Pedreschi, D.: Blocking anonymity threats raised by frequent itemset mining. In: ICDM'05. (2005) 561–564
4. Oliveira, S.R.M., Zaïane, O.R., Saygin, Y.: Secure association rule sharing. In: PAKDD. (2004) 74–85
5. Kantarcioglu, M., Jin, J., Clifton, C.: When do data mining results violate privacy? In: KDD'04, ACM Press (2004) 599–604
6. Jin, H., Chen, J., Kelman, C., He, H., McAullay, D., O'Keefe, C.M.: Mining unexpected associations for signalling potential adverse drug reactions from administrative health databases. In: PAKDD'06. (2006) 867–876
7. Ayres, J., Gehrke, J., Yiu, T., Flannick, J.: Sequential PAttern Mining using a bitmap representation. In: KDD'02. (2002) 215–224
8. Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems **10** (2002) 557–570

Mining Concept Associations for Knowledge Discovery Through Concept Chain Queries

Wei Jin[1], Rohini K. Srihari[1], and Xin Wu[2]

[1] Department of Computer Science & Engineering, University at Buffalo, State University of New York, USA
[2] Department of Computer Science and Technology, University of Science and Technology of China, China
wjin2@buffalo.edu, rohini@cedar.buffalo.edu, xinwu@mail.ustc.com.cn

Abstract. The availability of large volumes of text documents has created the potential of a vast amount of valuable information buried in those texts. This in turn has created the need for automated methods of discovering relevant information without having to read it all. This paper focuses on detecting links between two concepts across text documents. We interpret such a query as finding the most meaningful evidence trail across documents that connect these two concepts. In this paper we propose to use link-analysis techniques over the extracted features provided by Information Extraction Engine for finding new knowledge. We compare two approaches to perform this task. One is the concept-profile approach based on traditional bag-of-words model, and the other is the graph-based approach which combines text mining, graph mining and link analysis techniques. Counterterrorism corpus is used to evaluate the performance of each model and demonstrates that the graph-based approach is preferable for finding focused information. For greater coverage of information we should use the concept-profile based approach.

Keywords: Knowledge Discovery; Text Mining; Link Analysis.

1 Introduction

It is recognized that text information is growing at an astounding pace. These vast collections of publications offer an excellent opportunity for text mining, i.e., the automatic discovery of knowledge. The main theme of our paper is based on the hypotheses that "The wealth of recorded knowledge is greater than the sum of its parts", which means a document collection often, reveals interesting information other than what is explicitly stated, interesting links and hidden information that connect facts, propositions or hypotheses can be formed by using some techniques to discover previously unknown logic connections among the existing information we have.

The goal of this paper is to sift through these extensive document collections and find such links. The problem addressed here focuses on detecting links between two concepts across documents. A traditional search involving, for example, two person

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 555–562, 2007.

names will attempt to find documents mentioning both of these individuals. Failing this, the search results in documents containing one of the names. This research focuses on a different interpretation of such a query: what is the best evidence trail across documents that connect these two individuals? For example, both may be involved with educational institutions, although not necessarily the same one; this information is gleaned from multiple documents. A generalization of this task involves query terms representing general concepts (e.g. airplane crash, criminal prosecution). We refer to this type of query as a concept chain query, a special case of text mining. Fig.1illustrates an example of a concept chain connecting *Amir Abdelgani* and *Mohammed Saleh* in the corpus. The connection is through the concept *fuel*; the model presented here was able to pick up this connection in spite of the textual distance between the two concepts in question.

> *Mohammed Saleh*, who provided fuel from his Yonkers gas station to make bombs, obtained legal permanent residency by marrying an American. Ibrahim Ilgabrowny passed messages between conspirators and obtained five fraudulent Nicaraguan passports for his cousin, El Sayyid Nosair, and his family. Nosair, convicted of conspiracy, married an American in 1982 and became a citizen in 1989. He was also convicted of a gun charge in the killing of Rabbi Meir Kahane in 1990. *Amir Abdelgani* picked up fuel and helped determine targets; he, too, was married to an American.

Fig. 1. Sample concept chain and evidence

The remainder of the paper is organized as follows. Section 2 discusses related work. Section 3 presents concept-profile text mining approach, and section 4 introduces semantic approach using graph-based retrieval model. Section 5 describes the experiments based on processing the 9-11 corpus and is followed by conclusions.

2 Related Work

Much of the work in hypotheses generation makes use of an idea originated by Swanson in the 1980s called the "complementary structures in disjointed literatures" (CSD). Swanson realized that large databases of scientific literature could allow discoveries to be made by connecting concepts using logical inference. He proposed a simple "A influences B, and B influences C, therefore A may influence C" model for detecting instances of CSD that is commonly referred as Swanson's ABC model [6]. Using this technique, he found a connection implying patient benefit between fish oil and Raynaud's syndrome, two years before clinical trials established that the benefit was real. Since their pioneering contributions this kind of knowledge discovery work has attracted the attention of other researchers. Gordon and Lindsay were among the first to use this approach, followed a few years later by Weeber [7]. More recently, Srinivasan has used this approach to demonstrate the feasibility of her approach based on MeSH terms and UMLS semantic types and presented open and closed text mining

algorithm that are built within the discovery framework established by Swanson and Smalheiser [5]. Our approach is motivated by Srinivsan's closed text mining algorithm and we extended this technique [2, 3, 4].

There has been work on discovering connections between concepts across documents using social network graphs, where nodes represent documents and links represent connections between documents. However much of the work on social network analysis has focused on special problems, such as detecting communities [8]. [9] is the work which is close to the problem being presented here. The authors model the problem of detecting associations between people as finding a connection subgraph and present a solution based on electricity analogues. However there are several differences which should be noted. The most notable difference is the reliance on URL links to establish connections between documents. Our approach extracts associations based on content (textual) analysis. Second, the connection subgraph approach presents all paths together, while our approach presents the paths individually. This allows greater user input in determining the best paths, including recency, novelty, semantic coherence, etc. Third, the approach presented here attempts to generate an explanation of the chains, whereas the connection subgraph approach does not. Finally, the connection subgraph solution only addresses named entities whereas this approach extends to general concepts.

3 Concept-Profile Text Mining Approach

Concept extraction involves running an information extraction engine, Semantex [1] on the corpus. Semantex tags named entities, common relationships associated with person and organization, as well as providing subject-verb-object (SVO) relationships. We extract as concepts all named entities, as well as any noun phrases participating in SVO relationships. All named entity instances are retained as instances of their respective named entity concept category such as *Organization*, *Country* and *Human Action*. In the following sections, we use *semantic type* to denote named entity concept category.

3.1 Concept Profiles

A profile is essentially a set of concepts that together represent the corresponding *topic*. We build topic profiles by first identifying a relevant subset of documents from the text collection, then identify characteristic concepts (single words and/or phrases) from this subset and assess their relative importance as descriptors of the topic. Concepts are extracted from the free-text portions of the documents which co-occur with the topic in the sentence level. The profiles are weighted vectors of concepts as shown below for a topic T_j:

$$\text{Profile } (T_j) = \{w_{j,1}m_1,\ w_{j,2}m_2, \cdots, w_{j,n}m_n\} . \quad (1)$$

Where m_k represents a concept, $w_{j,k}$ its weight and there are totally n concepts in the concept dictionary.

3.2 Employing Semantic Types in Profiles

Up to now our profiles are simply vectors of weighted concepts. Now we describe how to further differentiate between the concepts using semantic types. Basically concepts are separated by semantic type and concept weights are computed within the context of a semantic type. This result is a vector of concept vectors, one for each of semantic types. Thus,

$$\text{Profile } (T_j) = \{w_{j,1,1}m_{1,1},\cdots,\},\cdots,\{w_{j,n,1}m_{n,1,}\cdots,\}\} \ . \quad (2)$$

Where $m_{x,y}$ represents the concept m_y that belongs to the semantic type x, $w_{j,x,y}$ is the computed weight for $m_{x,y}$. To calculate weight, we use a variation of TF*IDF weighting scheme and then normalize the weights:

$$w_{j,x,y} = u_{j,x,y} / \text{highest}(u_{j,x,d}). \quad (3)$$

Where $d=1, \cdots, l$ and $u_{j,x,y} = n_{j,x,y} \times log(N/n_{x,y})$.

Here N is the number of documents in the collection, $n_{x,y}$ is the number of documents in which $m_{x,y}$ occurs and $n_{j,x,y}$ is the number of retrieved documents for T_j in which $m_{x,y}$ co-occurs with T_j in the same sentence. Then we normalize by highest ($u_{j,x,d}$), the highest value for $u_{j,x,y}$ observed for the concepts with semantic type x, produces weights that are in [0, 1] within each semantic type. (Note that there are l concepts in the domain for semantic type x).

Table 1 illustrates a portion of the concept profile that is constructed for *Bush*; the best concepts are shown.

Table 1. Portion of profile for concept 'Bush'

Semantic Type	Concept	Weight
Title	President	1.00
Country	North Korea	1.00
	Iraq	0.882826
Person	Bin Ladin	1.00
	Woodward	0.891192
Army	Central Command	1.00
	Defense	0.802759
Province	Texas	1.00
	Washington	0.663360

3.3 Generating Paths Between Concepts

This stage finds potential conceptual connections in different levels, creates the concept chain and ranks them according to the weight of the corresponding selected concept. The basic algorithm is based on the method proposed in [5] but we adapted it to meet our needs and also extended the technique to generate concept chains. The algorithm is composed of the following steps where the user input is two topics/concepts of interest designated, *A* and *C*.

1. Conduct independent searches for *A* and *C*. Build the *A* and *C* profiles. Call these profiles *AP* and *CP* respectively.

2. According to the semantic types for intermediate concepts specified by the user, compute a *B* profile (*BP*) composed of terms in common between *AP* and *CP* within the specified semantic types. The weight of a concept in *BP* is the sum of its weights in *AP* and *CP*, respectively. Concepts are retained and ranked by estimated potential for each specified semantic type. This is the first level of intermediate potential concepts.
3. Expand the concept chain using the created *BP* profile together with the topics to build additional levels of intermediate concept lists which (i) connect the topics to each concept in *BP* profile in the sentence level within specified semantic types, and (ii) also normalize and rank them.

Output: Levels of potential concepts ranked by their weights within specified semantic types. A potential conceptual connection between *A* and *C* is a path starting from topic *A*, proceeding through sequential levels of intermediate concepts until reaching the ending topic *C*.

4 Concept-Graph Based Text Mining Approach

The Concept Graphs, which are an extension of concept vectors, are made automatically by calculating the significance of concept-concept associations. The formal definition is as follows:

4.1 Graph Construction

Definition 1: A Graph G (N, E) representing a document collection (subset) is a weighted label graph where N is a set of nodes; E is a collection of weighted edges.

- *Node*: Concept in the document collection. Multiple instances of a single concept are treated as one unique node in the graph.
- *Edge*: Constructed based on proximity and co-occurrence relationship between concepts. If the two concepts co-occur within a window unit (i.e. paragraph, sentence), then there is an edge connecting them.
- *Weight*: Represents the strength of such relationship.

Weight $\omega_{A,B}$ can be calculated as the co-occurrence frequency of concept A and B within the window, in our experiment, we use the following formula analogous to Dice Coefficient to measure this relationship:

$$\omega_{A,B} = \log (1+ F (A,B)) . \quad (4)$$

Where $F (A, B) = 2\times N_{A, B} / (N_A+N_B)$, N_A (N_B) is the frequency of concept A (B) in the document collection. $N_{A, B}$ is the co-occurrence frequency of concept A and B within the window. Based on this model, each document can be represented as follows:

$$D=[C_1, C_2, \cdots , C_n] \quad A=[a_{ij}]. \quad (5)$$

Where *D*: concept list. C_i: the i^{th} concept in the concept dictionary. *A*: the diagonal association matrix among concepts. a_{ij}: the association strength between concept C_i and C_j ($1 \le i < j \le n$). To normalize *A*, we get a matrix $W = [\omega_{i,j}]$.

$$\omega_{ij} = \frac{a_{ij}}{\sum_k \sum_l a_{kl}} \cdot \tag{6}$$

4.2 Generating and Ranking Concept Chains

This stage finds potential conceptual connections, creates concept chains and ranks them according to the weight of the corresponding selected chains.

Firstly the graph undergoes cleaning phase. The user may adjust the size of the constructed graph using parameter *Edge Support*. This parameter is to filter the edges whose weights are below such threshold designated by the user. Next, the graph searching algorithm of finding the top n maximal-weight paths connecting these two concepts in different length levels, is employed. For instance, the existence of the chain of length 1 means there is a direct link between these two concepts within the window; The chains of length more than 1 indicate there exists an unapparent association, and the intermediate concepts are suggested by the retrieved chains and ranked by their estimated potential. Due to computational consideration, the algorithm combines the depth-first search and width-first search together. The users may specify *path length* (*searching depth*) and *path number* according to their needs through setting the appropriate parameters.

The ranking of concept chains takes a total order defined as follows:

Definition: Given two concept chains, r_i and r_j, $r_i \succ r_j$ (also called r_i precedes r_j or r_i has a higher precedence than r_j) if

1. the length of r_i is less than that of r_j or
2. their lengths are the same, but the total weight of chain r_i (sum of the weights of traversed edges in the chain r_i) is greater than that of r_j.

5 Experiments on Counterterrorism Corpus

For the experiments we used the 9/11 commission report as the corpus. This involves processing a large open source document collection pertaining the 9/11 attack, including the publicly available 9/11 commission report. The whole collection was preprocessed using Semantex [1]. At the end of preprocessing, a concept dictionary including 9131 concepts is created and each concept is assigned to one or more ontology category.

5.1 Evaluation set

The objectives of the evaluation were to measure precision and recall of the concept chains that the system generated. Precision was judged by manual inspection of the top n chains. For recall, we synthesized an evaluation set as follows. We selected chains of lengths ranging from 1 to 4. The chains were selected as follows:

- We ran queries with various pairs of named entities, that is, the end points of the chains were named entities (although intervening concepts were not required to be named entities). This was done to facilitate manual judgment of the goodness of chains.

- The textual windows relevant to each query pair were then manually inspected: we selected those where there was a logical connection between the two concepts.
- We then generated the concept chains for these concept pairs (and documents) as evaluation data.

5.2 Experiment Result

This section presents the results on the evaluation set. The experiment processes 30 query pairs in the evaluation set and generates concept chains of lengths ranging from 1 to 4 for each query pair. We make the comparison by calculating the average precision and recall of the chains each technique created for all the query pairs. Table 2 summarizes the results we obtain on executing concept chain queries from the evaluation set. The comparison will be used to emphasis the strength and weakness of each technique compared to each other.

Table 2. Comparison of average precision and recall

Model	Average Precision	Average Recall
Concept-profile Model	76.19%	89.81%
Graph Model	83.77%	81.50%

As a post analysis, the concept-profile model handles a majority of the queries with a recall of 89.81%. The recall parameter shows the strength of the concept-profile model presented here. Through combining concept profiles and ontology information, we got good coverage of the links we were looking for. However, the precision in the semantic links represented by the graph model was better than that of concept-profile model, which performs 83.77% comparing with 76.19% achieved by concept-profile model. Our main conclusion is that if we need much focused information then the best results will be obtained by using the semantic links represented by the graph model. When we look for greater coverage of information we should use the concept-profile model.

6 Conclusion

This paper focuses on detecting links between two concepts across text documents (e.g. two persons). We interpret such a query as finding the most meaningful evidence trail across documents that connect these two concepts. We proposed to use link-analysis techniques over the extracted features provided by Information Extraction Engine for finding new knowledge and compared two approaches to perform this task. One is the concept-profile approach integrating ontology information, and the other is the graph-based approach which combines text mining, graph mining and link analysis techniques. Counterterrorism corpus is used to evaluate the performance of each model and demonstrates that the semantic links represented by the graph model is preferable for finding focused information. For greater coverage of information we should use the concept-profile based approach.

Future directions include the development of more sophisticated retrieval models that can combine various evidence sources (concept order, occurrence, context, novelty etc.) in one model. We are also researching extensions of concept chains to concept graph queries. This will enable users to quickly generate hypotheses graphs which are specific to a corpus. These matched instances can then be used to look for other, similar scenarios.

References

1. Srihari, R. K., Li, W., Niu, C. and Cornell, T. InfoXtract: A Customizable Intermediate Level Information Extraction Engine. Journal of Natural Language Engineering. Cambridge, U. Press (2006) 1–26
2. Jin, W. and Srihari, R. K. Graph-based Text Representation and Knowledge Discovery. Proc. of 22-th ACM Symposium on Applied Computing, Seoul, Korea (2007)
3. Jin, W. and Srihari, R. K. Knowledge Discovery across Documents through Concept Chain Queries. Proc. Workshop of 2006 IEEE International Conf. on Data Mining: Foundation of Data Mining and Novel Techniques in High Dimensional Structural and Unstructured Data (2006) 448-452
4. Srihari, R. K., Lamkhede, S. and Bhasin, A. Unapparent Information Revelation: A Concept Chain Graph Approach. Proc. of ACM Conf. Information and Knowledge Management, ACM Press (2005) 329-330
5. Srinivasan, P. Text Mining: Generating Hypothesis from Medline. JASIST, Vol. 55 (2004) 396-413
6. Swason, D. R. Complementary Structures in Disjoint Science Literatures. Proc. of 14-th ACM SIGIR Conf. on Research and Development in Information Retrieval, ACM Press (1991) 280-289
7. Weeber, M., Vos, R. and Klein, H. Generating Hypothesis by Discovering implicit Associations in the Literature. J. Amer. Med. Inform. Assoc. Vol. 10. (2003) 252-259
8. Gibson, D., Kleinberg, J. and Raghavan, P. Inferring Web Communities from Link Topology. Proc. of 9-th ACM Conf. on Hypertext and Hypermedia (1998) 225-234
9. Faloutsos, C. McCurley, K. S. and Tomkins, A. Fast Discovery of Connection Subgraphs. Proc. of 10-th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (2004) 118-127.

Capability Enhancement of Probabilistic Neural Network for the Design of Breakwater Armor Blocks

Doo Kie Kim[1], Dong Hyawn Kim[2], Seong Kyu Chang[1], and Sang Kil Chang[1]

[1] Department of Civil and Environmental Engineering, Kunsan National University, Kunsan, Jeonbuk 573-701, South Korea
kim2kie@chol.com, s9752033@kunsan.ac.kr, sangkil-jjang@hanmail.net
[2] Department of Ocean System Engineering, Kunsan National University, Kunsan, Jeonbuk 573-701, South Korea
eastlite@nate.com

Abstract. In this study, the capability of probabilistic neural network (PNN) is enhanced. The proposed PNN is capable of reflecting the global probability density function (PDF) by summing the heterogeneous local PDF automatically determined in the individual standard deviation of variables. The present PNN is applied to predict the stability number of armor blocks of breakwaters using the experimental data of van der Meer, and the estimated results of PNN are compared with those of empirical formula and previous artificial neural network (ANN) model. The PNN showed better results in predicting the stability number of armor blocks of breakwaters and provided the promising probabilistic viewpoints by using the individual standard deviation in a variable.

Keywords: breakwater; armor block; stability number; multivariate Gaussian distribution; classification; artificial neural network (ANN); probabilistic neural network (PNN).

1 Introduction

Armor units are designed to defend breakwaters from repeated wave loads. Because armor units are decided by the stability numbers, these numbers are very important to design rubble mound breakwaters. The stability of rubble mound breakwaters is usually analyzed by the well-known empirical formulae by Hudson [1] and van der Meer [2]. Those formulae are used to determine the individual weight of armor blocks of breakwaters. Although those formulae were derived from a number of experimental data, they show too much disagreement between the measured stability numbers and the predicted ones. The uncertainties in the empirical formulae inevitably increase the factor of safety and eventually, the construction cost. Therefore, a number of studies have been carried out to develop an advanced empirical formula for breakwater stability.

Kaku [3] and Kaku et al. [4] proposed an empirical formula for the damage level prediction based on the van der Meer's experimental data. Smith et al. [5] compared

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 563–570, 2007.

their own test results with the prediction by Kaku et al. [4]. Hanzawa et al. [6] proposed an empirical formula for stability number based on their own test data. Although several empirical formulae have been proposed for decades, remarkable improvement has not been seen. Recently, Mase et al. [7] examined the applicability of artificial neural network (ANN) to analyzing the stability of rubble-mound breakwater and compared the predicted stability numbers by neural network with the measured ones of van der Meer and Smith et al. [5]. The ANN technique seems to make a breakthrough in the design of rubble mound breakwaters. Actually, the stability numbers predicted by the ANN agree better than those by van der Meer's ([7]). The stability number, however, still needs to be improved. Kim, D. H. et al. [8] presented several network models to predict the stability number of armor blocks of breakwaters. The same training data set is used for ANN but the structures of the ANN and the number of nodes at input and hidden layer differ from those of Mase's ANN. Even if the ANN technique shows better performance than the empirical model based approach in breakwater design, it can be adapted to new data through a re-training process and needs more efforts to determine the architecture of network and more computational time in training the network. Moreover, the estimated results from the ANN are not probabilistic but deterministic. The probabilistic neural network (PNN), therefore, could be an effective and reasonable alternative, because PNN needs less time to determine the architecture of the network and to train the network. Moreover the PNN provides the probabilistic viewpoints as well as deterministic classification results.

In this paper, the PNN technique is enhanced to reflect the global probability density function (PDF) by summing the heterogeneous local PDF. The heterogeneous local PDF of the PNN is automatically determined to use the individual standard deviation of variables. Training and test patterns for the PNN are prepared using the data sets from the experimental data of van der Meer [9]. The predicted stability numbers are compared with those measured by laboratory. The results show that the PNN can effectively predict the stability numbers in spite of data complexity, incompleteness, and incoherence, and it can be an effective tool for designers of rubble mound breakwaters to support their decision process and to improve design efficiency.

2 Capability Enhancement of PNN

PNN is basically a pattern classifier that combines the well-known Bayesian decision strategy with the Parzen non-parametric estimator of the PDFs of different classes [10]. PNN has gained interest because it offers a way of interpreting the network's structure in the form of a probability density function and it is easy to implement. An accepted norm for decision rules or strategies used to classify patterns is that they do so in a way that minimizes the "expected risk." Such strategies are called Bayesian strategies" and can be applied to problems containing any number of classes.

Parzen showed how one may construct a family of estimates of $f(\mathbf{X})$, [11], and Cacoullos has also extended Parzen's results to estimates in the special case that the multivariate kernel is a product of univariate kernels [12]. In the particular case of the Gaussian kernel, the multivariate estimates can be expressed as

$$f_A(\mathbf{X}) = \frac{1}{(2\pi)^{p/2}\sigma^p}\frac{1}{m}\sum_{i=1}^{m}\exp\left[-\frac{(\mathbf{X}-\mathbf{X}_{Ai})^T(\mathbf{X}-\mathbf{X}_{Ai})}{2\sigma^2}\right] \tag{1}$$

where $\mathbf{X}$ is the test vector to be classified; $f_A(\mathbf{X})$ is the value of the PDF of category A at input $\mathbf{X}$; m is the number of training vectors in category A; p is the dimensionality of the training vectors; $\mathbf{X}_{Ai}$ is the i^{th} training vector for category A; and σ is the smoothing parameter. Note that $f_A(\mathbf{X})$ is the simple sum of small multivariate Gaussian distributions centered at each training sample because only one global smoothing parameter is used.

The pattern layer of PNN consists of a number of pattern units. Each pattern unit (shown in more detail in Fig. 1) forms a dot product of the input pattern vector $\mathbf{X}$ with a weight vector $\mathbf{W}_i$, $Z_i = \mathbf{X}\cdot\mathbf{W}_i$, and then performs a nonlinear operation on Z_i before outputting its activation level to the summation unit. Instead of the sigmoid activation function commonly used for back propagation neural network (BPNN), the nonlinear operation used here is $\exp[(Z_i - 1)/\sigma^2]$. Assuming that both $\mathbf{X}$ and $\mathbf{W}_i$ are normalized to unit length, this is equivalent to using

$$\exp[-\frac{(\mathbf{X}-\mathbf{W}_i)^T(\mathbf{X}-\mathbf{W}_i)}{2\sigma^2}] \tag{2}$$

which is the same form as Equation (1).

To complement the defect of the conventional PNN using one global smoothing parameter, Berthold and Diamond [13] suggested a constructive probabilistic neural network (CPNN) by taking different smoothing parameters for different patterns. Jin et al. [14] applied the CPNN to classify the freeway traffic patterns for incident detection. However, CPNN needed to consider a different probabilistic property for each variable.

Each variable, such as the permeability of breakwater (P), the damage level (S_d), the surf similarity parameter (ξ_m), the dimensionless water depth (h/H_s), and the spectral shape (SS), has an individual standard deviation and a different probabilistic property. However, the PDF did not consider the individual probabilistic property of variables in PNN because only one global smoothing parameter was used. Therefore, in this paper, the PNN enhanced to reflect the global probability density function by summing the heterogeneous local PDFs automatically determined to use the individual standard deviation of variables. The basic idea is to individually use the heterogeneous local PDF in a variable because the probabilistic property of variables is not homogenous but heterogeneous. The individual PDF was derived from the standard deviation of variables. The PDF for i^{th} sample is determined to sum different standard deviations of the training vector with j^{th} variables (Fig. 2). Therefore, the nonlinear operation of enhanced PNN can be expressed as

$$\exp\left\{-\sum_{j=1}^{p}\left(\frac{(X_j - W_{i,j})^2}{2\sigma_j^2}\right)\right\} \tag{3}$$

where i and j are indices for the i^{th} training pattern and j^{th} variable; p is the number of variables; X_j is the j^{th} variable of input data; $W_{i,j}$ is the j^{th} variable of the i^{th} training vector; σ_j is the standard deviation with the j^{th} variable.

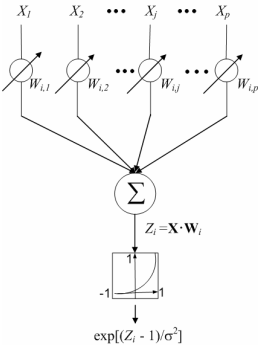


Fig. 1. Pattern layer of conventional PNN

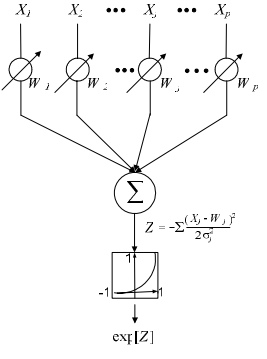


Fig. 2. Pattern layer of enhanced PNN

3 Prediction of Stability Numbers Using PNN

In order to apply the enhanced PNN to the prediction of stability numbers, the rule base which implicitly tells the input (design condition) output (stability number) relationship should first be composed by using the so-called training patterns. About two thirds of the experimental data by van der Meer's ([9]) were used as training patterns, and the others as test patterns to evaluate the performance of the constructed PNN. In the van der Meer's 641data, there are only two cases for the number of wave attack; 1000 and 3000 wave attacks. In general, it is not easy to have nonlinear mapping function using only two cases of data in function mapping problems. Therefore, two PNNs were separately constructed; PNN1 is for 1000 attacks and composed of 326 experimental data sets. PNN2 is for 3000 attacks and composed of 315 experimental data sets. The measured stability numbers that were defined as output whose definitions are respectively set to 207 and 196 classes (the ranges of the measured stability numbers was from 0.7907 to 4.3848). In order to make the training pattern an adequate representation of the class distinctions, PNN1 and PNN2 were constructed using 207 and 196 training patterns out of 326 and 315 experimental data sets, respectively, which correspond to 1000 and 3000 wave attacks. Five design parameters including the permeability of breakwater (P), the damage level (S_d), the surf similarity parameter (ξ_m), the dimensionless water depth (h/H_s), and the spectral shape (SS) were used as the input set for PNN and all the input data are

normalized to 0.1~0.9 to give an equal weighting factor before implementing the data to the network. In the network, in cases of impermeability core, permeability core, and homogeneity structure, the permeability of breakwater (P) was assumed to be respectively 0.1, 0.5, and 0.6. In cases of Pierson Moskowitz, narrow, and wide spectrum, the spectral shapes (SS) were used to be respectively 1, 2, and 3. Table 1 shows the samples for construction of PNN using the van der Meer [9]'s data. The global PDF of the PNN was derived from Equation (3). Table 2 shows the standard deviation values and means of the normalized variables.

Table 1. The samples of training patterns for construction of PNN

	Input(normalized data)						Output(class)
	N_w	P	S_d	ξ_m	h/H_s	SS	N_s
	1000	0.1(0.1)	0.76(0.11)	7.15(0.85)	14.47(0.76)	1(0.1)	2(0.94)
	1000	0.1(0.1)	2.14(0.15)	4.85(0.58)	10.06(0.53)	1(0.1)	23(1.35)
	1000	0.1(0.1)	4.04(0.20)	2.13(0.27)	8.82 (0.47)	1(0.1)	43(1.55)
P	1000	0.1(0.1)	4.55(0.22)	2.19(0.28)	7.14 (0.39)	1(0.1)	84(1.96)
N	1000	0.1(0.1)	7.31(0.30)	2.19(0.28)	6.37 (0.35)	1(0.1)	104(2.14)
N	1000	0.1(0.1)	3.73(0.20)	1.41(0.19)	5.22 (0.29)	1(0.1)	147(2.64)
1	1000	0.1(0.1)	4.07(0.21)	0.76(0.11)	4.15 (0.24)	1(0.1)	191(3.29)
	1000	0.1(0.1)	11.66(0.42)	0.78(0.11)	3.17 (0.19)	1(0.1)	206(4.30)
	3000	0.1(0.10)	0.65(0.10)	5.37(0.68)	17.24(0.90)	1(0.1)	1(0.79)
	3000	0.1(0.10)	3.43(0.15)	6.84(0.86)	9.78(0.52)	1(0.1)	27(1.39)
	3000	0.1(0.10)	5.26(0.18)	4.00(0.51)	8.21(0.44)	1(0.1)	52(1.68)
	3000	0.6(0.90)	1.29(0.11)	5.57(0.70)	7.20(0.39)	1(0.1)	75(1.91)
	3000	0.5(0.74)	7.29(0.22)	3.71(0.48)	5.11(0.29)	1(0.1)	97(2.13)
P	3000	0.1(0.10)	25.96(0.55)	1.72(0.23)	5.09(0.29)	1(0.1)	149(2.71)
N	3000	0.1(0.10)	7.89(0.23)	0.76(0.11)	4.15(0.24)	1(0.1)	182(3.29)
N	3000	0.1(0.10)	20.81(0.46)	0.78(0.12)	3.17(0.19)	1(0.1)	196(4.30)
2	3000	0.1(0.1)	2.65(0.139	4.84(0.615)	12.66(0.664)	3(0.9)	1.077(7)
	3000	0.1(0.1)	1.49(0.119	5.34(0.677)	12.17(0.640)	1(0.1)	1.119(8)
	3000	0.1(0.1)	0.86(0.107)	2.24(0.294)	11.92(0.627)	3(0.9)	1.144(9)
	3000	0.1(0.1)	2.53(0.137	2.57(0.335)	11.68(0.615)	1(0.1)	1.167 (10)

Table 2. Standard deviation and mean of normalized variables

Variables	PNN1		PNN2	
	σ	Mean	σ	mean
P	0.3291	0.4004	0.3296	0.4027
S_d	0.1206	0.2362	0.1334	0.2502
ξ_m	0.1583	0.3727	0.1565	0.3709
h/H_s	0.1295	0.3790	0.1305	0.3806
SS	0.2082	0.1724	0.2033	0.1698

To compare the performance of each model in a more reasonable way, the agreement index (I_a) and the correlation coefficient (CC) are used as follows [15]

$$I_a = 1 - \frac{\sum_{i=1}^{n} (e_i - m_i)^2}{\sum_{i=1}^{n} [|e_i - \overline{m}| + |m_i - \overline{m}|]^2} \tag{4}$$

where e_i and m_i denote the estimated and the measured stability numbers respectively; $\overline{m}$ is the average of measured stability numbers; T is the transpose matrix. If I_a is close to one, the predicted set agrees well to the measured set.

In case where all the experimental data including trained patterns are used as testing patterns, the results are shown in Table 3. PNN models seem to be the best predictor in this example.

To evaluate the generalized capability of the ANN and PNN, they were tested only by untrained patterns. The results are shown in Table 4. In which, the results by ANN and PNN models show slight deterioration compared to those in Table 3. However, the comparison results show that PNN can effectively predict the stability numbers.

Table 3. Performance of stability models with all patterns including training patterns

	VM1	VM2	ANN1	ANN2	PNN1	PNN2
Ia	0.926	0.927	0.959	0.962	0.991	0.989
CC	0.875	0.877	0.925	0.929	0.982	0.977

Table 4. Performance of stability models with only untrained patterns

	ANN1	ANN2	PNN1	PNN2
Ia	0.928	0.931	0.955	0.949
CC	0.904	0.897	0.949	0.902

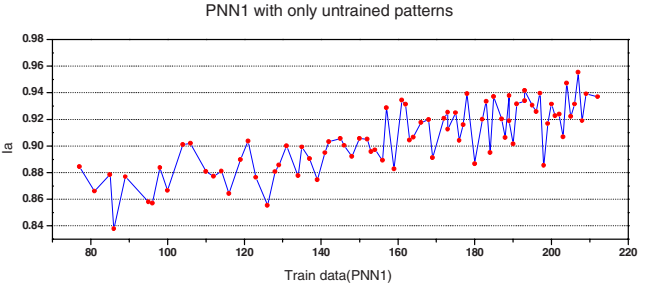


Fig. 3. I_a of PNN1 according to the number of trained patterns

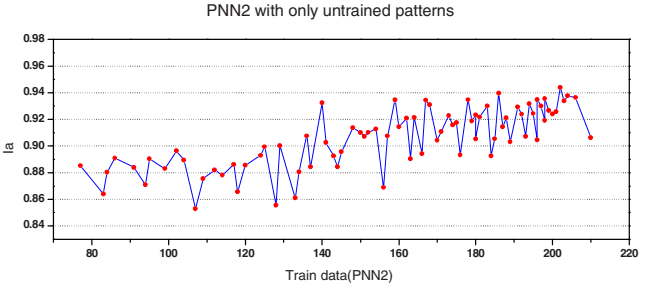


Fig. 4. I_a of PNN2 according to the number of trained patterns

To optimize the construction of the PNN, I_a of models were compared according to the number of training patterns. Figs. 3 and 4 show respectively the trends of I_a of PNN1 and PNN2 according to the number of trained patterns.

4 Conclusions

An enhanced PNN method was proposed and incorporated to predict the stability number of breakwater. The permeability of breakwater (P), damage level (S_d), surf similarity parameter (ξ_m), dimensionless water depth (h/H_s), and the spectral shape (SS) are used as inputs to the PNN, and the stability number of breakwater is defined as classes to be predicted by the proposed method. From the results, it has been found that the estimation performance of the proposed method is more effective than those of the empirical model and ANN. We can find the optimum condition of the construction of the PNN through the trend according to the number of training patterns. Also, the proposed technique has following merits as:

(1) It can provide the probabilistic viewpoint as well as deterministic classification results in considering the uncertainties in the design of rubble mound breakwaters.
(2) The heterogeneous local PDF of the PNN is automatically determined to use the individual standard deviation of variables.

Acknowledgments. This work is a part of a research project supported by Korea Ministry of Construction & Transportation (MOCT) through Korea Bridge Design & Engineering Research Center at Seoul National University. The authors wish to express their gratitude for the financial support.

References

1. Hudson, R.Y.: Design of Quarry Stone Cover Layer For Rubble Mound Breakwaters, Research Report No. 2-2. Waterways Experiment Station, Coastal Engineering Research Centre, Vicksburg, Miss (1958)
2. van der Meer, J.W.: Deterministic and probabilistic design of breakwater armor layers. Vol.114. No.1. Ocean Engineering (1988a) 66-80

3. Kaku, S.: Hydraulic stability of rock slopes under irregular wave attack, Master Thesis, University of Delaware, Newark, Del. (1990)
4. KaKu, S., Kobayashi, N., and Ryu, C.R.: Design formulas for hydraulic stability of rook slopes under irregular wave attack, Proceedings of 38th Japanese Conference Coastal Engineering (1991) 661-665
5. Smith, W.G., Kobayashi, N., and KaKu, S.: Profile changes of rock slopes by irregular waves, Proceedings of 23th International Conference Coast Engineering ASCE (1992) 1559-1572
6. Hanzawa, M., Sato, H., Takahashi, S., Shimosako, K., Takayama, T., and Tanimoto, K.: New stability formula for wave-dissipating concrete blocks covering horizontally composite breakwaters, Proceedings of 25th Coastal Engineering Conference, ASCE (1996) 1665-1678
7. Mase, H., Sakamoto, M. and Sakai, T.: Neural network for stability analysis of rubble-mound breakwater, Vol.121. No.6. ASCE Journal of waterway, port, coastal, and ocean Engineering (1995) 294-299
8. Kim, D.H. and Park, W.S.: Neural network for design and reliability analysis of rubble mound breakwaters, Vol.32. No.11/12. Ocean engineering (2005) 1332-1349
9. van der Meer, J.W.: Rock slopes and gravel beaches under wave attack. PhD Thesis, Delft Univ. of Technology., Delft, The Netherlands (1988b)
10. Specht, D. F.: Probabilistic Neural Networks. Vol.3. Neural Networks (1990) 109-118
11. Parzen, E.: On estimation of a probability density function and mode. Vol.33. Annals of Mathematical Statistics (1962) 1065-1076
12. Cacoullos, T.: Estimation of a multivariate density. Vol.18. No.2. Annals of the Institute of Statistical Mathematics (Tokyo) (1966) 179-189
13. Berthold, M.R. and Diamond, J.: Constructive training of probabilistic neural networks. Neurocomputing (1998) 167-183
14. Jin, X., Cheu, R.L., and Srinivasan, D.: Development and adaptation of constructive probabilistic neural network in freeway incident detection. Vol.10. Transportation Research Part C (2002) 121-147
15. Willmott, C.J.: On the validation of models, Vol.2. No.2. Phys. Geogr. (1981) 184-194

Named Entity Recognition Using Acyclic Weighted Digraphs: A Semi-supervised Statistical Method

Kono Kim[1], Yeohoon Yoon[2], Harksoo Kim[3,*], and Jungyun Seo[4]

[1] Natural Language Processing Laboratory, Department of Computer Science, Sogang University, 1 Sinsu-dong, Mapo-gu, Seoul, 121-742, Korea
kono@sogang.ac.kr
[2] NHN corporation, Venture Town Bldg., 25-2 Jeongja-dong, Bundang-gu, Seongname-City, Gyeonggi-do, 463-844, Korea
yhyoon@nhncorp.com
[3] Program of Computer and Communications Engineering, College of Information Technology, Kangwon National University, 192-1, Hyoja 2(i)-dong, Chuncheon-si, Gangwon-do, 200-701, Korea
nlpdrkim@kangwon.ac.kr
[4] Department of Computer Science and Interdisciplinary Program of Integrated Biotechnology, Sogang University, 1 Sinsu-dong, Mapo-gu, Seoul, 121-742, Korea
seojy@sogang.ac.kr

Abstract. We propose a NE (Named Entity) recognition system using a semi-supervised statistical method. In training time, the NE recognition system builds error-prone training data only using a conventional POS (Part-Of-Speech) tagger and a NE dictionary that semi-automatically is constructed. Then, the NE recognition system generates a co-occurrence similarity matrix from the error-prone training corpus. In running time, the NE recognition system constructs AWDs (Acyclic Weighted Digraphs) based on the co-occurrence similarity matrix. Then, the NE recognition system detects NE candidates and assigns categories to the NE candidates using Viterbi searching on the AWDs. In the preliminary experiments on PLO (Person, Location and Organization) recognition, the proposed system showed 81.32% on average F1-measure.

Keywords: named entity recognition, semi-supervised statistical method, acyclic weighted digraph.

1 Introduction

As NEs (Named Entities) such as organization's names, person's names and location's names contain more informative information, NE recognition is the fundamental for efficient information access. Generally, the NE recognition methods are divided into two kinds; rule-based methods and statistical methods. The rule-based methods use regular-expression-like patterns and NE dictionaries [6]. If the NE

* Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 571–578, 2007.

dictionaries are so much massive and the patterns are generated by referring to a large corpus, the performances of the rule-based methods will be good. However, it is well known that managing a lot of rules is very difficult and the cost for the initial implementation is high. Meanwhile, the statistical methods collect statistical knowledge from corpus and determine NE categories based on the statistical knowledge. The statistical methods can be divided into two kinds according to their learning methods; supervised learning methods and unsupervised learning methods. The supervised learning methods perform well, but the performances of the supervised learning methods depend on the size of NE tagged training data. If the size of NE tagged training data is small, most of supervised learning methods will raise the sparse data problems. On the other hand, the unsupervised learning methods do not require NE tagged training data, but the performances of the unsupervised learning methods are much less than those of the supervised learning methods. Recent researches have been focused on improving the accuracy of NE recognition based on some supervised learning models such as DT (Decision Tree) [7], MEM (Maximum Entropy Model) [2], HMM (Hidden Markov Model) [1], and CRF (Conditional Random Field) [3]. However, these approaches still need a large amount of NE tagged training corpus. To reduce the time-consuming tasks of training data construction, we propose a semi-supervised statistical method which combines a supervised factor (i.e. looking up a NE dictionary) with an unsupervised factor (i.e. training based on a large raw corpus).

This paper is organized as follows. In Section 2, we propose a NE recognition system based on a semi-supervised learning method. In Section 3, we explain experimental results. Finally, we draw some conclusions in Section 4.

2 NE Recognition Using Acyclic Weighted Digraphs

The proposed system consists of a knowledge acquisition module and a NE recognition module. Using a conventional POS (Part-Of-Speech) tagger and a NE dictionary, the knowledge acquisition module, first, naively extracts NE candidates from a raw corpus and assigns all possible categories to the NE candidates. To construct the NE dictionary, we collected PLO entities (i.e. person's names, location's names, organization's names) from an on-line yellow page and semi-automatically classified the PLO entities into 50 subcategories by using an on-line encyclopedia. As a result, the NE dictionary includes 53 kinds of named entities that are annotated with their categories. Then, the knowledge acquisition module calculates all possible co-occurrence similarities between NE candidates and adjacent content words although the NE tagged corpus includes many errors. When sentences are input, the NE recognition module simply finds all possible NE candidates from the input sentences by using the same method with the knowledge acquisition module. Then, the NE recognition module filters out inadequate NE candidates and classifies each unfiltered NE candidate into one among 53 categories using the co-occurrence similarities that are already calculated by the knowledge acquisition module.

2.1 NE Dictionary Construction

To construct the NE dictionary, we collect 489,212 PLO entities (400,438 person's names, 46,776 location's names and 41,998 organization's names) from an on-line yellow page. Then, we automatically assign subcategories to the PLO entities by using the genera of lemmas in an on-line encyclopedia (http://100.naver.com). Most encyclopedias describe lemmas with genera and specific differences. A genus is a kind of a class to which each lemma belongs, and a specific difference is a kind of difference from other members of its category. For example, the lemma, 'Woo island', has its description like 'an island which belongs to one of districts in Jeju island'. In such case, the genus of 'Woo island' is 'island', and the specific difference is 'which belongs to one of districts in Jeju island'. Based on these characteristics of encyclopedias, we manually construct a mapping table with genus words and their categories, as shown in Table 1.

Table 1. A part of the mapping table

Genus (in Korean)	Category
연구소, 연구원, 연구센터	Laboratory
시	City
국가	Country
학교	School
산, 봉, 산맥, 능선	Mountain
섬	Island
바위	Rock
강, 천	River
의사, 변호사, 박사	Expert
관공서, 공사, 공단	Government
국회의사당	Assembly

By looking up the mapping table, we automatically assigned subcategories to each PLO entity. If a PLO entity does not exist in the encyclopedia or the genus of a PLO entity does not exist in the mapping table, we do not assign a subcategory to the PLO entity. Then, we manually correct misclassified entities.

2.2 Co-occurrence Similarity Acquisition

To calculate co-occurrence similarities between NEs and adjacent content words, the knowledge acquisition module extracts all NE candidates from a raw corpus and assigns all possible categories to the NE candidates by using a POS tagger and the NE dictionary, as shown in Fig. 1. In this paper, we use articles of Chosun-Ilbo (One of Korean daily newspapers; http://www.chosun.com, 2,698,196 raw sentences from 1996 to 1997) as a training corpus.

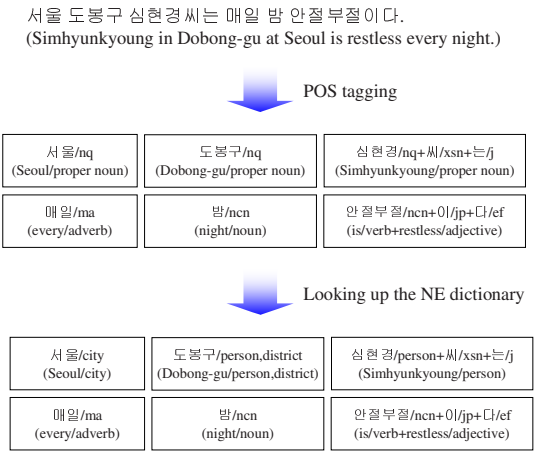


Fig. 1. The conceptual image of the training corpus

Then, the knowledge acquisition module calculates all possible co-occurrence similarities between NE candidates and content words, as shown in Equation (1).

$$Sim(X,Y) = \frac{f(X,Y)}{f(X)+f(Y)-f(X,Y)} \tag{1}$$

In Equation (1), X can be a content word or a NE category, and Y can also be a content word or a NE category. $f(X)$ is the frequency of X in the training corpus, and $f(X,Y)$ is the frequency of X occurring with Y in a sentence. If X and Y are content words, Equation (1) will return the co-occurrence similarity between the two content words. If X is a content word and Y is a NE category, Equation (1) will return the co-occurrence similarity between the content word and the NE category. Finally, the knowledge acquisition module constructs a co-occurrence similarity matrix by gathering all possible co-occurrence similarities, as shown in Fig. 2.

In Fig. 2, n and m are the number of content words and the number of NE categories in the training corpus, respectively. w_i is the ith content word, where content words are arranged in descending order of frequencies $f(w_1) \geq \ldots \geq f(w_i) \geq \ldots \geq f(w_n)$. c_j is the jth NE category, where NE categories are arranged in descending order of frequencies $f(c_1) \geq \ldots \geq f(c_j) \geq \ldots \geq f(c_m)$.

	w_1	…	w_n	c_1	…	c_m
W_1	1.0	…	0.1	0.7	…	0.4
…	…	…	…	…	…	…
w_n	0.1	…	1.0	0.1	…	0.1
C_1	0.7	…	0.1	1.0	…	0.2
…	…	…	…	…	…	…
c_m	0.4	…	0.1	0.2	…	1.0

Fig. 2. The conceptual image of the co-occurrence similarity matrix

2.3 NE Detection

When a sentence is input, the NE recognition module first extracts noun phrases[1] from the sentence by using the POS tagger and some heuristics. Then, the NE recognition module naively assigns all possible NE categories to the noun phrases by looking up the NE dictionary. Finally, the NE recognition system determines if the NE candidates are not fake but real by searching the optimal path on an AWD (Acyclic Weighted Digraph). In this paper, an AWD consists of two sets, V and E, where V is a finite nonempty set of vertices which represent NE categories or content words, and E is a set of vertex pairs that are connected with directed arcs from the ith vertex to the $i+1$th vertex in a sequence of NE categories or content words. Fig. 3 shows an example of the AWD for detecting real NEs.

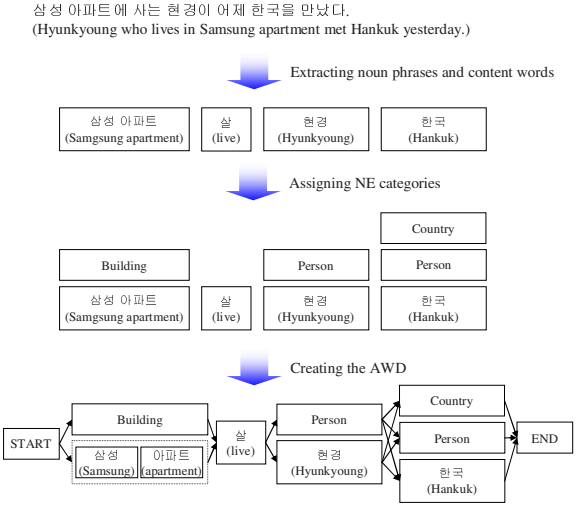


Fig. 3. An example of the AWD for detecting real NEs

To find the optimal pass on the AWD, we modify the Viterbi algorithm [8] that is well-known as the best searching algorithm using dynamic programming. Actually, the Viterbi algorithm uses the transition probabilities multiplied by observation probabilities in HMM. However, the NE recognition module cannot directly calculate the observation probabilities and transition probabilities because the training corpus is unlabeled. Therefore, we assume that the observation probabilities of all vertices are 1.0 and the transition probabilities can be approximated to the co-occurrence similarities between adjacent vertices. Although our assumptions include some flaws, we believe that the transition probabilities will be similar to co-occurrence similarities obtained from a large training corpus. Based on these assumptions, the NE recognition module obtains the co-occurrence similarities between adjacent vertices

[1] In this paper, we consider a sequence of nouns as a noun phrase.

by looking up the co-occurrence similarity matrix and uses the co-occurrence similarities as the transition probabilities. Then, the NE recognition module determines the optimal pass on the AWD by using the modified Viterbi algorithm. If a NE candidate consists of a multi-word phrase like '삼성 아파트 (Samsung apartment)' in Fig. 3, the NE recognition module calculates co-occurrence similarities of each word in the multi-word phrase and selects the maximum value as the co-occurrence similarity of the multi-word phrase, as shown in Equation (2). In other words, the co-occurrence similarity of 'START & 삼성 아파트(Samsung apartment)' is the maximum one among the co-occurrence similarities, 'START & 삼성 (Samsung)' and 'START & 아파트 (apartment)'.

$$tr_{multi}(v_i, v_{i+1}) = \underset{1 \le j \le n}{MAX}\{tr(v_i, v_{i+1}^j)\} \tag{2}$$

In Equation (2), v_i is the ith vertex on the AWD, and v_{i+1}^j the jth word in the $i+1$th vertex that consists of n words.

2.4 NE Categorization

After detecting real NEs, the NE recognition module assigns categories to the real NEs. The NE recognition module first removes unnecessary vertices for categorization. For example, if '삼성 아파트(Samsung Apartment)' is detected as a real NE in Fig. 3, the NE recognition module will remove the two lexical forms of vertices '삼성 (Samsung)' and '아파트 (apartment)' on the AWD, as shown in Fig. 4.

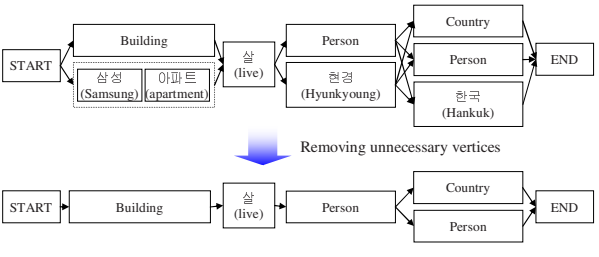


Fig. 4. An example of the AWD for assigning NE categories

Then, the NE recognition module finds the optimal pass on the modified AWD by using the same method with the NE detection. Finally, the NE recognition module assigns proper categories to real NEs by tracing the optimal pass.

3 Preliminary Experiments

3.1 Data Sets

MUC (Message Understanding Conference) has provided on ongoing forum for assessing the state of the art in text analysis technology and for exchanging

information on innovative computational techniques in realistic tasks [4], [5]. However, MUC does not provide any test collections to evaluate a Korean NE recognition system. Therefore, to evaluate the performance of the proposed system, we built a Korean test collection in a telebanking domain. The test collection consists of 1,769 sentences that include 380 person's names, 173 location's names, and 238 organization's names. In this paper, we defined 53 NE categories (PLO categories and 50 their subcategories), but we could not evaluate performances on classification of the 53 NE categories because we did not completely construct the test collection yet. Therefore, we evaluated the performances on classification of only the PLO categories (i.e. person's names, location's names, and organization's names). Although the preliminary experiments are incomplete and coarse, we think that the preliminary experiments have some meaning because our goal is to recognize NE categories without fully-annotated training corpus. We are trying to supplement and expand the Korean test collection and will completely evaluate performances of the proposed system in the future.

3.2 Performance Evaluation

To evaluate performance of the proposed system, we used the F1-measure, as shown in Equation (3).

$$F1 = \frac{2rp}{r + p} \tag{3}$$

In Equation (3), p is the precision that means proportion of correct ones out of returned NE categories, and r is the recall rate that means proportion of returned NE categories out of classification targets.

Table 2 shows the performance of the proposed system. As shown in Table 2, the performance is 81.32% on average F1-measure which is 9.28% higher than the baseline system that used only NE dictionary.

Table 2. The performance of the proposed system

	The proposed system			The baseline system		
Category	Precision	Recall	F1	Precision	Recall	F1
Person	85.75	87.11	86.43	76.27	90.53	83.40
Location	71.38	68.39	69.88	46.53	51.94	49.23
Organization	83.27	92.02	87.64	74.16	92.86	83.51
Average	80.13	82.51	81.32	65.65	78.44	72.04

The performance for Organization is relatively good, while the performance for Location is poor. The difference is caused by biased training; the proposed system is actually tuned to Organization categories because the training data, the articles of Chosun-Ilbo, holds probably more organization's names than location's names.

4 Conclusion

We proposed a NE recognition system using a semi-supervised statistical method. In training time, the proposed system extracts all possible NE candidates by looking up the NE dictionary that is semi-automatically constructed. Then, the proposed system constructs a matrix that includes co-occurrence similarities between NE candidates and adjacent content words. In running time, the proposed system generates AWDs based on the co-occurrence similarity matrix. Then, the NE recognition system detects NE candidates and assigns proper categories to the NE candidates using Viterbi searching on the AWDs. In the preliminary experiments on PLO (Person, Location, and Organization) recognition, the proposed system showed 81.32% on average F1-measure. Based on the experimental results, we think that the proposed system may be a good solution to reduce the time-consuming tasks of training data construction because it does not require a large amount of NE tagged training corpus.

Acknowledgement. This research (paper) was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Commerce, Industry and Energy of Korea. It was also partially supported by Kangwon Institute of Telecommunications and Information (KITI).

References

1. Bikel, D. M., Miller, S., Schwartz, R.: Nymble: a High-performance Learning Name-finder. In: Proceedings of the Fifth Conference on Applied Natural Language Processing, (1997) 194-201
2. Borthwick, A., Sterling, J., Agichtein, E., Grishman, R.: NYU: Description of the MENE Named Entity System as Used in MUC-7. In: Proceedings of the Seventh Message Understanding Conference, (1997)
3. Cohen, W. W., Sarawagi, S.: Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extraction Processes and Data Integration Methods. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (2004)
4. MUC-6: http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html, (1995)
5. MUC-7: http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_proceedings/o- verview.html, (1997)
6. Seon, C. N., Ko, Y., Kim, J., Seo, J.: Named Entity Recognition Using Machine Learning Methods and Pattern-Selection Rules. In: Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium, (2001)
7. Sekine, S., Grishman, R., Shinnou, H.: A Decision Tree Method for Finding and Classifying Names in Japanese Texts. In: Proceedings of 6th Workshop on Vary Large Corpora, (1998)
8. Viterbi, A. J.: Error Bounds for Convolution Codes and an Asymptotically Optimal Decoding Algorithm. IEEE Transactions on Information Theory, Vol. 13, (1967), 260-269

Contrast Set Mining Through Subgroup Discovery Applied to Brain Ischaemina Data

Petra Kralj[1], Nada Lavrač[1,2], Dragan Gamberger[3], and Antonija Krstačić[4,⋆]

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] Nova Gorica Polytechnic, Vipavska 13, 5000 Nova Gorica, Slovenia
[3] Rudjer Bošković Institute, Bijenička 54, 10000 Zagreb, Croatia
[4] University Hospital of Traumatology, Draškovićeva 19, 10000 Zagreb, Croatia

Abstract. Contrast set mining aims at finding differences between different groups. This paper shows that a contrast set mining task can be transformed to a subgroup discovery task whose goal is to find descriptions of groups of individuals with unusual distributional characteristics with respect to the given property of interest. The proposed approach to contrast set mining through subgroup discovery was successfully applied to the analysis of records of patients with brain stroke (confirmed by a positive CT test), in contrast with patients with other neurological symptoms and disorders (having normal CT test results). Detection of coexisting risk factors, as well as description of characteristic patient subpopulations are important outcomes of the analysis.

1 Introduction

Data analysis in medical applications is characterized by the ambitious goal of extracting potentially new relationships from the data, and providing insightful representations of detected relationships. Medical data analysis is frequently performed by applying rule learning, as the induced rules are easy to be interpreted by human experts.

The goal of standard classification rule learners [5] is to induce classification/prediction models from labeled examples. Opposed to these *predictive induction* algorithms which induce a model in the form of a set of rules, *descriptive induction* algorithms aim to discover individual patterns in the data, described in the form of individual rules. Descriptive induction algorithms include association rule learners [1], and subgroup discovery systems [2,6,8,11].

This paper addresses a data analysis task where groups of labeled examples are given and the goal is to find differences between the groups. This data analysis task, named *contrast set mining*, was first presented in [3]. In this paper we

⋆ This work was supported by Slovenian Ministry of Higher Education, Science and Technology project "Knowledge Technologies", Croatian Ministry of Science, Education and Sport project "Machine Learning Algorithms and Applications", and EU FP6 project "Heartfaid: A knowledge based platform of services for supporting medical-clinical management of the heart failure within the elderly population".

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 579–586, 2007.

propose to solve this task by transforming the contrast set mining task to a subgroup discovery task and to apply the subgroup discovery methodology to solve the task. This approach solves some open issues of existing contrast set mining approaches, like dealing with continuous valued attributes, choosing an appropriate search heuristic, selecting the level of generality of induced rules, avoiding of overlapping rules, and presenting the results to the end-users.

Although the goals of contrast set mining, which aims at finding differences between contrasting groups, and subgroup discovery, which aims at finding descriptions of population subgroups, seem different, this paper proves that the goals are the same and the results can be interpreted in both ways. The proposed approach of contrast set mining through subgroup discovery (presented in Section 4) was applied to a real-life problem of analyzing patients with brain ischaemia (presented in Section 2), where we provide insightful data analysis helping to answer questions about the severity of the brain damage based on risk factors obtained from physical examination data, laboratory test data, ECG data and anamnestic data. The usefulness of the approach is shown by the achieved results (Section 5) interpreted by medical specialists.

2 Brain Ischaemia Data

The brain ischaemia dataset consists of records of patients who were treated at the Intensive Care Unit of the Department of Neurology, University Hospital Center "Zagreb", Zagreb, Croatia, in year 2003. In total, 300 patients are included in the dataset: 209 with the computed tomography (CT) confirmed diagnosis of brain stroke, and 91 patients who entered the same hospital department with adequate neurological symptoms and disorders, but were diagnosed as patients with transition ischaemic brain attack (TIA, 33 patients), reversible ischaemic neurological deficit (RIND, 12 patients), and severe headache or cervical spine syndrome (46 patients). In this paper, the goal of the experiments is to characterize brain stroke patients confirmed by a positive CT test in contrast with the patients with a normal CT test.

Patients are described with 26 descriptors representing anamnestic, physical examination, laboratory test and ECG data, and their diagnosis. Anamnestic data: aspirin therapy *(asp)*, anticoagulant therapy *(acoag)*, antihypertensive therapy *(ahyp)*, antiarrhytmic therapy *(aarrh)*, antihyperlipoproteinaemic therapy - statin *(stat)*, hypoglycemic therapy *(hypo)*, sex *(sex)*, age *(age)*, present smoking *(smok)*, stress *(str)*, alcohol consumption *(alcoh)*, family anamnesis *(fhis)*. Physical examination data are: body mass index *(bmi)*, systolic blood pressure *(sys)*, diastolic blood pressure *(dya)*, fundus ocular *(fo)*. Laboratory test data: uric acid *(ua)*, fibrinogen *(fibr)*, glucose *(gluc)*, total cholesterol *(chol)*, triglyceride *(trig)*, platelets *(plat)*, prothrombin time *(pt)*. ECG data: heart rate *(ecgfr)*, atrial fibrillation *(af)*, left ventricular hypertrophy *(ecghlv)*.[1]

The diagnosis of patients is based on the physical examination confirmed by the CT test. All the patients in the control group have a normal brain CT test

[1] Details can be found on http://lis.irb.hr/PAKDD2007paper/.

in contrast with the positive CT test of patients with a confirmed brain attack. It should be noted that the group of patients with brain stroke and the control group do not consist of healthy persons but of patients with suspected severe neurological symptoms and disorders. In this sense, the available dataset is particularly appropriate for studying the specific characteristics and subtle differences that distinguish the two groups. While the detected relationships can be accepted as the actual characteristics for these patients, the computed evaluation measures—including probability, specificity and sensitivity of induced rules—only reflect characteristics specific to the available data set, not necessarily holding for the general population or other medical institutions.

3 Methodological Background

A common question of exploratory data analysis is "What is the difference between the given groups?" where the groups are defined by a selected property of individuals that distinguishes one group from the others. For example, the distinguishing property that we want to investigate could be the gender of patients and a question to be explored can be "What is the difference between males and females affected by a certain disease?" or, if the property of interest was the response to a treatment, the question can be "What is the difference between patients reacting well to a selected drug and those that are not?" Searching for differences is not limited to any special type of individuals: we can search for differences between molecules, patients, organizations, etc.

Data analysis tasks that try to find differences between contrasting groups are very common and the approach presented here can be applied in many of these tasks. When the end-users ask for differences characterizing different groups, they are usually not interested in all the differences; they may prefer a small set of representative and interpretable patterns. Finding all the patterns that discriminate one group of individuals from the other contrasting groups is not appropriate for human interpretation. Therefore, as is the case in other descriptive induction tasks, the goal is to find descriptions that are unexpected and interesting to the end-user.

The approach presented in this paper offers this kind of analysis. From a dataset of class labeled instances (the class label being the property of interest) by means of subgroup discovery [7] we can find interpretable rules that offer a good starting point for human analysis of contrasting groups.

Contrast set mining. The problem of mining contrast sets was first defined in [3] as finding "conjunctions of attributes and values that differ meaningfully in their distributions across groups." They proposed the STUCCO algorithm [3], which is based on Bayardo's Max-Miner [4] rule discovery algorithm. In the level-wise search for contrast sets, formed of conjunctions of attribute-value pairs of length i, the interestingness of the conjunct is estimated by its statistical significance, assessed using a χ^2 test with a Bonferroni correction. Domain specific parameters need to be set, like the minimum support difference between groups. The algorithm works only on domains with nominal attributes.

It was shown in [10] that contrast set mining is a special case of a more general rule learning task, and that a contrast set can be interpreted as an antecedent of a rule and $Group_i$, for which it is characteristic, as the rule consequent: $ContrastSet \rightarrow Group_i$.

When using rule learners (OPUS-AR and C4.5 rules) for contrast set mining [10], the user needs to select a quality measure (choosing between support, confidence, lift, coverage and leverage). In this setting the number of generated rules largely exceeds the number of rules generated by STUCCO, unless pruned by the user-defined maximum number of rules parameter. Expert interpretation of rules is difficult due to a large amount of rules and sometimes also their specificity.

Subgroup discovery. A subgroup discovery task is defined as follows: "Given a population of individuals and a property of those individuals that we are interested in, find population subgroups that are statistically 'most interesting', e.g. are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest" [11]. The result of subgroup discovery is a relatively small set of *subgroup descriptions* formed of conjunctions of features. Members of a subgroup are examples from the dataset that correspond to the subgroup description. Good subgroups are large (descriptions covering many examples with the given property of interest), and have a significantly different distribution of examples with the given property compared to its distribution in the entire population.

Subgroup discovery algorithms include adaptations of rule learning algorithms to perform subgroup discovery [7,8] algorithms for relational subgroup discovery [9,11] and algorithms for exploiting background knowledge for discovering non-trivial subgroups [2], among others.

Since subgroup descriptions are conjunctions of features that are characteristic for a selected class of individuals (property of interest), a subgroup description can be seen as a condition of a rule $SubgroupDescription \rightarrow Class$ and therefore subgroup discovery can be seen as a special case of a more general rule learning task.

4 Contrast Set Mining Through Subgroup Discovery

We present an approach to contrast set mining by means of subgroup discovery. Even though the definitions of subgroup discovery and contrast set mining seem different, we here provide a proof of the compatibility of the tasks. Furthermore, by subgroup discovery means, we solve the following open issues in contrast set mining [10]: proposing appropriate heuristics for identifying interesting contrast sets, appropriate measures of quality of contrast sets, and appropriate methods for presenting contrast sets to the end-users. The issue of dealing with continuous attributes is also solved by subgroup discovery algorithm SD [7].

Translating contrast set mining tasks to subgroup discovery tasks. Contrast set mining and subgroup discovery were developed in different communities, each developing its own terminology that needs to be clarified before

Table 1. Table of synonyms from different communities

Contrast Set Mining (CSM)	Subgroup Discovery (SD)	Rule Learning (RL)
contrast set	subgroup description	rule condition
group	class (property of interest)	class
attribute value pair	feature	condition
examples in groups G_1, G_2 ($G_3 \ldots G_n$)	examples of $Class$ and $\overline{Class}$	examples of C_1, C_2 ($C_3 \ldots, C_n$)
examples for which contrast set is true	subgroup	covered examples
support of contrast set on G_1	true positive rate	true positive rate
support of contrast set on G_2	false positive rate	false positive rate

proceeding. In order to show the compatibility of contrast set mining and subgroup discovery tasks, we first define the *compatibility* of terms used in different communities as follows: terms are compatible if they can be translated into equivalent logical expressions and if they bare the same meaning, i.e., if terms from one community can replace terms used in another community.

To show that terms used in contrast set mining (CSM) can be translated to terms used in subgroup discovery (SD), Table 1 provides a term dictionary through which we translate the terms used in CSM and SD into a unifying terminology of classification rule learning.

We now wish to show that every contrast set mining task (CSM) can be translated into a subgroup discovery task (SD). The definitions of contrast set mining and subgroup discovery appear different: contrast set mining searches for discriminating characteristics of groups called contrast sets, while subgroup discovery searches for subgroup descriptions.

A contrast set is formally defined as follows: Let $A_1, A_2, ..., A_k$ be a set of k variables called attributes. Each A_i can take values from the set $\{v_{i1}, v_{i2}, ..., v_{im}\}$. A contrast set is a conjunction of attribute value pairs defined on user defined groups $G_1, G_2, ..., G_n$ of data instances, whose characteristics we wish to uncover through contrast set mining [3]. A special case of contrast set mining considers only two contrasting groups G_1 and G_2. In such cases, we wish to find characteristics of one group discriminating it from the other and vice versa.

In subgroup discovery, subgroups are described as conjunctions of features of the form $A_i = v_{ij}$ for nominal attributes, and $A_l > value$ or $A_l \leq value$ for continuous attributes. The subgroup discovery task aims at finding population subgroups that are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest [11].

Using the dictionary of Table 1 it is trivial to show that a two-group contrast set mining task $CSM(G_1, G_2)$ can be directly translated into the following two subgroup discovery tasks: $SD(Class = G_1$ vs. $\overline{Class} = G_2)$ and $SD(Class = G_2$ vs. $\overline{Class} = G_1)$. Since this translation is possible for two-group contrast set mining, it is—by induction—also possible for a general contrast set mining task.

Solving open issues of CSM with SD

In this paper, contrast set mining is performed by subgroup discovery algorithm SD [7], an iterative heuristic beam search rule learner.

Handling continuous attributes: SD uses a feature-based data representation, where attribute values needed for the construction of features are generated automatically from the data. In this way, the SD algorithm overcomes a deficiency of CSM: handling of continuous attributes.

Rule quality heuristic: At each run, the SD algorithm finds subgroups for a selected property of interest and a selected generalization parameter g. The output of the SD algorithm is a set of rules with good covering properties on the given example set, which is obtained by using rule quality heuristic $q_g(R) = \frac{TP}{FP+g}$, where TP (true positives) denotes the number of covered examples from the positive class, FP (false positives) the number covered negative examples, and generalization parameter g offers the user the opportunity to influence the degree of specificity of rules, since with large g general rules are preferred by the q_g heuristic, while with small g each covered negative example is severely punished thus generating specific rules.[2]

Rule diversity: To obtain diverse rules in different iterations, the algorithm implements weighting of covered positive examples after selecting a rule. Instead of the unweighted $q_g(R)$ measure, the weighted rule quality measure replaces TP with the sum of weights of covered positive examples. Although this approach can not guarantee the statistical independence of generated rules, it aims at ensuring good diversity of induced rules. This can be verified also from the results presented in the following section.

Presenting the results to end-users: In the next section we present some visualization methods with the results of our experiments. The visualizations proved to be intuitive and useful to the domain experts, and can help estimating the quality of the results.

5 Results of Brain Ischaemia Data Analysis

In this section we illustrate the usage of the presented approach of contrast set mining through subgroup discovery including the visualizations of the results.

There are several questions that medical doctors find interesting and that can be investigated by using the presented method and dataset. Due to space restrictions of this paper, we concentrate only on the question *"What is the difference between patients with confirmed stroke and patients with other severe neurological disorders?"* Other questions that could be addressed in a similar manner are: "What is the difference between patients with TIA and RIND and the confirmed stroke patients?", "What is the difference between patients with thrombolic ischaemia and embolic ischaemia", and others.

For each of the two classes, Figure 1 shows three best rules induced by selecting $g = 10$ and $g = 50$, visualized with the bar visualization along with their TP and

[2] Generalization parameter values are usually selected in the range between 1 and 100; in our experiments values 10 and 50 were used.

g	rule	stroke 209	normal 91
g=10	(fibr > 4.45) and (age > 64) → stroke	41 %	0 %
	(af = yes) and (ahyp = yes) → stroke	28 %	5 %
	(str = no) and (alcoh = yes) → stroke	28 %	5 %
	(ahyp = no) and (fibr ≤ 4.55) and (dya ≤ 95.5) → normal	6 %	36 %
	(fibr ≤ 4.55) and (af = no) and (stat = no) and (dya ≤ 95.5) and (age ≤ 70) → normal	8 %	42 %
	(age ≤ 61) and (Fhis = no) and (asp = yes) → normal	0 %	12 %
g=50	(ahyp = yes) → stroke	74 %	46 %
	(fibr > 3.35) and (age > 58) → stroke	79 %	37 %
	(age > 52) and (asp = no) → stroke	64 %	37 %
	(fibr ≤ 4.55) and (af = no) and (fo ≤ 1) and (RRsys ≤ 190) → normal	25 %	71 %
	(fibr ≤ 4.55) and (fo ≤ 1) and (acoag = no) and (age ≤ 75) → normal	37 %	82 %
	(age ≤ 70) and (str = yes) and (stat = no) and (RRsys ≤ 190) → normal	19 %	54 %

Fig. 1. Contrast sets for groups (classes) brain stroke and normal, induced for g-values 10 and 50, visualized with the bar visualization

FP values. The order of rules is selected by the iterative SD algorithm and is determined by the q_g rule quality value that takes into account the covering relations between the current rule and other rules previously selected for the same g-value.

An interesting subgroup description is rule (age>52.00) *and* (asp=no), which stimulated the analysis presented in Figure 2. This analysis provides an excellent motivation for patients to accept prevention based on aspirin therapy, as the rule explicitly recognizes the importance of the aspirin therapy for persons older than 52 years.

In addition, the moderately sensitive and specific rules are relevant also for the selection of appropriate boundary values for numeric descriptors included into rule conditions. Examples are age over 58 years, and fibrinogen over 3.35. In the case of fibrinogen, reference values above 3.7 are treated as positive while rules induced for brain stroke domain suggest 4.45 in combination with age over 64 years, and 3.35 in combination with age over 58 years for more sensitive detection of stroke. These values, if significantly different from generally accepted reference

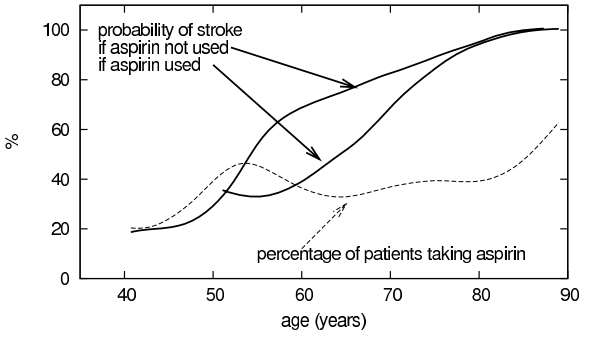


Fig. 2. The probability of brain stroke, estimated by the proportion of stroke patients, shown in dependence of patient age presented for patients taking aspirin as the prevention therapy, and the probability of stroke for patients without this therapy. The percentage of patients with the aspirin therapy is presented by a dashed line.

values, can initialize research in the direction of possibly accepting them as new decision points in medical decision making practice. Even more importantly, the fact that various boundary points can be suggested in combinations with different conditions is better than the existing medical practice which tends to define unique reference values irrespective of the disease that has to be described and irrespective of other patient characteristics.

6 Conclusions

This work demonstrates that subgroup discovery methodology is appropriate for solving contrast set mining tasks. It shows the results of contrast set mining through subgroup discovery applied to the problem of distinguishing between patients with and without brain stroke. Attention was devoted also to the selection of appropriate visualizations, enabling effective presentations of obtained results. The presented theory and experimental results show that using subgroup discovery for contrast set mining solves many open issues of contrast set mining.

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, AAAI Press:307–328, 1996.
2. M. Atzmueller, F. Puppe, and H.P. Buscher. Exploiting background knowledge for knowledge-intensive subgroup discovery. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)*, pages 647–652, 2005.
3. S. D. Bay and M. J. Pazzani. Detecting group differences: Mining contrast sets. *Data Min. Knowl. Discov.*, 5(3):213–246, 2001.
4. R. J. Bayardo. Efficiently mining long patterns from databases. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 85–93. ACM Press, 1998.
5. s P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
6. D. Gamberger and N. Lavrač. Descriptive induction through subgroup discovery: A case study in a medical domain. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 163–170, Morgan Kaufmann, 2002.
7. D. Gramberger and N. Lavrač. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, (17):501–527, 2002.
8. N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.
9. F. Železný and N. Lavrač. Propositionalization-based relational subgroup discovery with RSD. *Machine Learning*, 62:33–63, 2006.
10. G. I. Webb, S. Butler, and D. Newlands. On detecting differences between groups. In *KDD '03: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 256–265, New York, NY, USA, 2003. ACM Press.
11. S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the First European Conference on Principles of Data Mining and Knowledge Discovery*, pages 78–87, Springer, 1997.

Intelligent Sequential Mining Via Alignment: Optimization Techniques for Very Large DB

Hye-Chung Kum[1], Joong Hyuk Chang[2], and Wei Wang[1]

[1] University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA
[2] University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{kum, weiwang}@cs.unc.edu, jhchang@uiuc.edu

Abstract. The shear volume of the results in traditional support based frequent sequential pattern mining methods has led to increasing interest in new intelligent mining methods to find more meaningful and compact results. One such approach is the *consensus sequential pattern mining* method based on sequence alignment, which has been successfully applied to various areas. However, the current approach to consensus sequential pattern mining has quadratic run time with respect to the database size limiting its application to very large databases. In this paper, we introduce two optimization techniques to reduce the running time significantly. First, we determine the theoretical bound for precision of the proximity matrix and reduce the time spent on calculating the full matrix. Second, we use a sample based iterative clustering method which allows us to use a much faster k-means clustering method with only a minor increase in memory consumption with negligible loss in accuracy.

1 Introduction

The goal of sequential pattern mining is to detect patterns in a database comprised of sequences of *itemsets*. For example, retail stores often collect customer purchase records in sequence databases in which a sequential pattern indicates a customer's buying habit. In such a database, each purchase is represented as a set of items, *itemsets*, purchased together, and a customer sequence would be a sequence of such itemsets.

Sequential pattern mining is commonly defined as finding the complete set of frequent subsequences [1]. Much research has been devoted to efficient discovery of such frequent sequential patterns [1] [7] [8]. However, such problem formulation of sequential patterns has some inherent limitations. First, the result set is huge and difficult to use without more post processing. Even the number of maximal or closed sequential patterns are huge, and many of the patterns are redundant and not useful. Second, the exact match based paradigm is vulnerable to noise and variations in the data. Many customers may share similar buying habits, but few follow exactly the same buying patterns. Finally, frequency alone cannot detect statistically significant patterns [4].

To overcome these limitations, recently there is an increasing interest in new intelligent mining methods to find more meaningful and compact results. The new methods abandon the traditional paradigm and take a fundamentally different

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 587–597, 2007.

Table 1. Representing the underlying pattern

seq_1	⟨()	()	**(BC)**	**(DE)**⟩
seq_2	⟨**(A)**	()	**(BC**X**)**	**(D)**⟩
seq_3	⟨**(A**E**)**	(B)	**(BC)**	**(D)**⟩
seq_4	⟨**(A)**	()	**(B)**	**(DE)**⟩
consensus_pat	⟨(A)		(BC)	(D)⟩

approach. One such approach to intelligent sequential pattern mining is the *consensus sequential pattern mining* based on sequence alignment. Consensus sequential patterns can detect general trends in a group of similar sequences, and may be more useful in finding non-trivial and interesting long patterns. It can be used to detect general trends in the sequence database for natural customer groups, which is more useful than finding all frequent subsequences.

Formally, *consensus sequential patterns* are patterns shared by many sequences in the database but not necessarily exactly contained in any one of them. Table 1 shows a group of sequences and a pattern that is approximately similar to them. In each sequence, the bold items are those that are shared with the consensus pattern. seq_1 has all items in *consensus_pat*, in the same order and grouping, except it is missing item A and has an additional item E. Similarly, seq_4 is missing C and has an extra E. In comparison, seq_2 and seq_3 have all items but each has a couple of extra items. These evidences strongly indicate that *consensus_pat* is the hidden underlying pattern behind the sequences. Such pattern mining of *consensus patterns* can effectively summarize the database into common customer groups and identify their buying patterns.

An effective algorithm for consensus sequential pattern mining has been proposed in [5]. The alignment based method, ApproxMAP(APPROXimate Multiple Alignment Pattern mining), has been applied to many areas such as multi-database mining [3], temporal streaming data mining [6], and policy analysis. Moreover, a detailed comparison study of the alignment based and support base methods has shown the effectiveness of ApproxMAP [4].

However, ApproxMAP has quadratic time complexity with respect to the size of the database limiting its application to very large databases. The time complexity is dominated by the clustering step which has to calculate the proximity matrix and build the clusters. In this paper, we introduce two effective optimization techniques. First, ApproxMAP can be optimized by calculating the proximity matrix to only the needed precision. In this paper, we introduce and prove the theoretical bound of the required precision reducing the running time considerably. Second, the clustering step can be improved by adapting the well known k-means method to ApproxMAP. Here, we introduce modifications to the typical algorithm that address the issues with calculating the mean, cluster initialization, and determining the number of clusters required. We further investigate the tradeoff between time and space empirically to determine the appropriate sample size and the utility of the optimization technique.

The remainder of the paper is organized as follows. Section 2 illustrates the basic ApproxMAP algorithm. The details and theoretical basis for the optimization are given in Section 3. Finally, Section 4 presents the experimental results.

Table 2. Sequence database $\mathcal{D}$ lexically sorted

ID	Sequences	ID	Sequences
seq_4	⟨(A) (B) (DE)⟩	seq_6	⟨(AY) (BD) (B) (EY)⟩
seq_2	⟨(A) (BCX) (D)⟩	seq_1	⟨(BC) (DE)⟩
seq_3	⟨(AE) (B) (BC) (D)⟩	seq_9	⟨(I) (LM)⟩
seq_7	⟨(AJ) (P) (K) (LM)⟩	seq_8	⟨(IJ) (KQ) (M)⟩
seq_5	⟨(AX) (B) (BC) (Z) (AE)⟩	seq_{10}	⟨(V) (PW) (E)⟩

Table 3. cluster 1 ($min\_strenth = 50\% \wedge w \geq 4$)

seq_2	⟨(A)	()	(BCX)	()	(D)⟩	
seq_3	⟨(AE)	(B)	(BC)	()	(D)⟩	
seq_4	⟨(A)	()	(B)	()	(DE)⟩	
seq_1	⟨()	()	(BC)	()	(DE)⟩	
seq_5	⟨(AX)	(B)	(BC)	(Z)	(AE)⟩	
seq_6	⟨(AY)	(BD)	(B)	()	(EY)⟩	
seq_{10}	⟨(V)	()	()	(PW)	(E)⟩	
Weighted Seq $wseq_1$	(A:5, E:1,V:1, X:1,Y:1):6	(B:3, D:1):3	(B:6, C:4,X:1):6	(P:1,W:1,Z:1):2	(A:1,D:4, E:5,Y:1):7	7
Consensus Pattern	⟨(A)		(BC)		(DE)⟩	

Table 4. cluster 2 ($min\_strength = 50\% \wedge w \geq 2$)

seq_8	⟨(IJ)	()	(KQ)	(M)⟩	
seq_7	⟨(AJ)	(P)	(K)	(LM)⟩	
seq_9	⟨(I)	()	()	(LM)⟩	
Weighted Sequence $wseq_2$	⟨(A:1,I:2,J:2):3	(P:1):1	(K:2,Q:1):2	(L:2,M:3):3	3
Consensus Pattern	⟨(IJ)		(K)	(LM)⟩	

2 Consensus Sequential Pattern Mining: ApproxMAP

We presented sequential pattern mining based on sequence alignment in [5]. Extending research on string analysis, we generalized string multiple alignment to find consensus sequential patterns in ordered lists of sets. The power of multiple alignment hinges on the following insight: the probability that any two long data sequences are the same purely by chance is very low. Thus, if several long sequences can be aligned with respect to particular frequent items, we will have implicitly found sequential patterns that are statistically significant.

ApproxMAP has three steps. First, k nearest neighbor clustering is used to partition the database. Second, for each partition, the optimal multiple alignment is approximated by the following greedy approach: in each partition, two sequences are aligned first, and then a sequence is added incrementally to the current alignment until all sequences have been aligned. At each step, the goal is to find the best alignment of the added sequence, p, to the existing alignment of $p-1$ sequences. A novel structure, *weighted sequence*, is used to summarize the alignment information in each cluster. In short, a weighted sequence is a sequence of itemsets with a weight associated with each item. The item weight represents the strength of the item where *strength* is defined as the percentage of sequences in the alignment that have the item present in the aligned position. Third, a consensus pattern is generated for each partition.

Tables 2 to 4 is an example. Given Table 2, ApproxMAP (1) calculates the proximity matrix and partitions the data into two clusters ($k = 2$), (2) aligns the sequences in each cluster – the alignment compresses all the sequences into one weighted sequence per cluster, and (3) summarizes the weighted sequences into consensus patterns using the cutoff point *min_strength*. Note that the consensus patterns ⟨(A)(BC)(DE)⟩ and ⟨(IJ)(K)(LM)⟩ do not match any sequence exactly.

3 Optimizations to ApproxMAP

ApproxMAP has total time complexity of $O(N_{seq}^2 \cdot L_{seq}^2 \cdot I_{seq} + k \cdot N_{seq})$ where N_{seq} is the total number of sequences, L_{seq} is the length of the longest sequence, I_{seq} is the maximum number of items in an itemset, and k is the number of nearest neighbors considered. The time complexity is dominated by the clustering step which requires the computation of the proximity matrix ($O(N_{seq}^2 \cdot L_{seq}^2 \cdot I_{seq})$) and builds clusters ($O(k \cdot N_{seq})$). The quadratic run time with respect to the database size may limit its applications to very large databases. There are two components constituting the running time for calculating the proximity matrix: (1) the per cell calculation, $O(L_{seq}^2 \cdot I_{seq})$, and (2) the total of N_{seq}^2 cell calculations needed for the proximity matrix. We discuss how to optimize both in this section.

3.1 *k*-Nearest Neighbor (*k*-NN) Clustering

ApproxMAP uses uniform kernel density based k-nearest neighbor (k-NN) clustering. We have found that such a density based method worked well due to its ability to build clusters of arbitrary size and shape around similar sequences. In this agglomerative method, each point links to its closest neighbor, but (1) only with neighbors that have greater density, and (2) only up to k nearest neighbors. Thus, the algorithm essentially builds a forest of single linkage trees (each tree representing a natural cluster), with the proximity matrix defined as follows,

$$dist'(seq_i, seq_j) =$$

$$\begin{cases} dist(seq_i, seq_j) & \text{if } dist(seq_i, seq_j) \le dist_k(seq_i) \text{ and } Density(seq_j, k) < Density(seq_i, k) \\ MAX\_DIST & \text{if } dist(seq_i, seq_j) \le dist_k(seq_i) \text{ and } Density(seq_j, k) = Density(seq_i, k) \\ \infty & \text{otherwise} \end{cases} \quad (1)$$

where $dist(seq_i, seq_j) = \frac{D(seq_i, seq_j)}{\max\{\|seq_i\|, \|seq_j\|\}}$ and $MAX\_DIST = \max\{dist(seq_i, seq_j)\} + 1$. $D(seq_i, seq_j)$ is the commonly used hierarchical edit distance defined via a recurrence relation. $dist_k(seq_i)$ is the k-NN region defined as the maximum distance over all k-NN and $Density(seq_i, k) = \frac{n_k(seq_i)}{dist_k(seq_i)}$ where $n_k(seq_i)$ is the number of sequences in the k-NN region. An effective implementation has three steps : (1) build the proximity matrix, (2) build the k-NN list using the matrix, and (3) merge the k-NN sequences when applicable. The details are in [3].

3.2 Optimizing the Proximity Matrix Calculation

Each cell in the proximity matrix is calculated using Equation 1. Thus, the time complexity is $O(L_{seq}^2 \cdot I_{seq})$ for solving the recurrence relation for $D(seq_i, seq_j)$

Table 5. Recurrence relation table

	seq_7	(AJ)		(P)		(K)		(LM)	
seq_8	**0**	1		2		3		4	
(IJ)	1	$\frac{1}{2}$	2	2	3	3	4	4	5
		2	$\searrow$ $\mathbf{\frac{1}{2}}$	$1\frac{1}{2}$	$\rightarrow$ $\mathbf{1\frac{1}{2}}$	$2\frac{1}{2}$	$\rightarrow$ $2\frac{1}{2}$	$3\frac{1}{2}$	$\rightarrow$ $3\frac{1}{2}$
(KQ)	2	2	$1\frac{1}{2}$	$1\frac{1}{2}$	$2\frac{1}{2}$	$1\frac{1}{2}+\frac{1}{3}=1\frac{5}{6}$	$3\frac{1}{2}$	$3\frac{1}{2}$	$4\frac{1}{2}$
		3	$\downarrow$ $1\frac{1}{2}$	$2\frac{1}{2}$	$\searrow$ $1\frac{1}{2}$	$2\frac{1}{2}$	$\searrow$ $\mathbf{1\frac{5}{6}}$	$2\frac{5}{6}$	$\rightarrow$ $2\frac{5}{6}$
(M)	3	3	$2\frac{1}{2}$	$2\frac{1}{2}$	$2\frac{1}{2}$	$2\frac{1}{2}$	$2\frac{5}{6}$	$1\frac{5}{6}+\frac{1}{3}=2\frac{1}{6}$	$3\frac{5}{6}$
		4	$\downarrow$ $2\frac{1}{2}$	$3\frac{1}{2}$	$\searrow$ $2\frac{1}{2}$	$3\frac{1}{2}$	$\searrow$ $2\frac{1}{2}$	$3\frac{1}{2}$	$\searrow$ $\mathbf{2\frac{1}{6}}$

through dynamic programming such as shown in Table 5. Often a straight forward dynamic programming algorithm can be improved by only calculating up to the needed precision. Here we discuss how to reduce the per cell calculation time by stopping the calculation of such a table midway whenever possible.

In Table 5, we show the intermediate calculation along with the final cell value. Each cell $RR(p,q)$ has four values in a 2x2 matrix. Let us assume we are converting seq_7 to seq_8. Then, the top left value shows the result of moving diagonally by replacing itemset p with itemset q. The top right value is the result of moving down by inserting q. The bottom left cell is the result of moving right by deleting p. The final value in the cell, shown in the bottom right position, is the minimum of the three. The arrows indicate the direction. The minimum path to the final answer, $RR(\|seq_i\|, \|seq_j\|) = D(seq_i, seq_j)$, is shown in bold.

For example, when calculating the value for $RR(3,2) = 1\frac{5}{6}$, you can either replace (KQ) with (K) (upper left: $RR(2,1)+REPL((\mathsf{KQ}),(\mathsf{K})) = 1\frac{1}{2}+\frac{1}{3} = 1\frac{5}{6}$), insert (KQ) (upper right: $RR(3,1) + INDEL = 2\frac{1}{2} + 1 = 3\frac{1}{2}$), or delete (K) (lower left: $RR(2,2) + INDEL = 1\frac{1}{2} + 1 = 2\frac{1}{2}$). Since $1\frac{5}{6}$ is the minimum, the replace operation is chosen (diagonal). The final distance $2\frac{1}{6}$ can be found by following the minimum path: diagonal (REPLACE), right(DELETE), diagonal, and diagonal. This path gives the pairwise alignment shown in Table 4.

In ApproxMAP, we note that we do no need to know $dist(seq_i, seq_j)$ for all i, j to full precision. In fact, the modified proximity matrix based on $dist'(seq_i, seq_j)$ has mostly values of ∞ because $k \ll N$. Thus, if a cell is clearly ∞ at any point, we can stop the calculation and return ∞. This will reduce the per cell calculation time significantly. $dist'(seq_i, seq_j)$ is clearly ∞ if seq_i is not a k-nearest neighbor of seq_j, and seq_j is not a k-nearest neighbor of seq_i. Remember that the modified proximity matrix is not symmetric. The following theorems prove that seq_i and seq_j are not k-nearest neighbor of each other when $\frac{min\_row(p)}{max\{\|seq_i\|,\|seq_j\|\}} > max\{dist_k(seq_i), dist_k(seq_j)\}$ for any row p. Here $dist_k(seq_i)$ is the radius of the k-nearest neighbor region for sequence seq_i, and $min\_row(p)$ is the smallest value of row p in the recurrence table. In the following theorems, we denote a cell in the recurrence table as $RR(p,q)$ with the initial cell as $RR(0,0) = 0$ and the final cell as $RR(\|seq_i\|, \|seq_j\|) = D(\|seq_i\|, \|seq_j\|)$.

Theorem 1. *There is a connected path from $RR(0,0)$ to any cell $RR(p,q)$ such that (1) cells along the path are monotonically increasing, (2) the two indices*

never decrease (i.e. the path always moves downward or to the right), and (3) there must be at least one cell from each row 0 *to* $p-1$*, in the connected path.*

Proof. The theorem comes directly from the definitions. First, the value of any cell $RR(p,q)$ is constructed from one of the three neighboring cells (up, left, or upper left) plus a non-negative number. Consequently, the values have to be monotonically increasing. Second, every cell must be constructed from three neighboring cells - namely up, left, or upper left. Hence, the path must move downward or to the right. Finally, since there has to be a connect path from $RR(0,0)$ to $RR(p,q)$, there must be at least one cell from each row 0 to $p-1$.

Theorem 2. $RR(\|seq_i\|, \|seq_j\|)$ *is greater than or equal to the minimum row value in any row. (i.e.* $RR(\|seq_i\|, \|seq_j\|) \geq min\_row(p)$ *for all* $0 \leq p \leq \|seq_i\|$*)*

Proof. Let us assume that there is a row, p, such that $RR(\|seq_i\|, \|seq_j\|) < min\_row(p)$. Let $min\_row(p) = RR(p,q)$. There are two possible cases. First, $RR(p,q)$ is in the connected path from $RR(0,0)$ to $RR(\|seq_i\|, \|seq_j\|)$. Since the connected path is monotonically increasing by Theorem 1, $RR(\|seq_i\|, \|seq_j\|)$ must be greater then equal to $RR(p,q)$. Thus, $RR(\|seq_i\|, \|seq_j\|) \geq RR(p,q) = min\_row(p)$. This is a contradiction. Second, $RR(p,q)$ is not in the connected path from $RR(0,0)$ to $RR(\|seq_i\|, \|seq_j\|)$. Now, let $RR(p,a)$ be a cell in the connected path. Then, $min\_row(p) = RR(p,q)$ and $RR(p,a) \geq RR(p,q)$. Thus, $RR(\|seq_i\|, \|seq_j\|) \geq RR(p,a) \geq RR(p,q) = min\_row(p)$. This is also a contradiction. Thus, by contradiction $RR(\|seq_i\|, \|seq_j\|) < min\_row(p)$ does not hold for any rows p. In other words, $RR(\|seq_i\|, \|seq_j\|) \geq min\_row(p)$ for all rows p.

Theorem 3. *If* $\frac{min\_row(p)}{max\{\|seq_i\|, \|seq_j\|\}} > max\{dist_k(seq_i), dist_k(seq_j)\}$ *for any row* p*, then* seq_i *is not a* k*-NN of* seq_j*, and* seq_j *is not a* k*-NN of* seq_i*.*

Proof. By Theorem 2, $RR(\|seq_i\|, \|seq_j\|) = D(seq_i, seq_j) \geq min\_row(p)$ for any row p. Thus, $dist(seq_i, seq_j) = \frac{D(seq_i, seq_j)}{max\{\|seq_i\|, \|seq_j\|\}} \geq \frac{min\_row(p)}{max\{\|seq_i\|, \|seq_j\|\}}$ $> max\{dist_k(seq_i), dist_k(seq_j)\}$ for any row p. By definition, when $dist(seq_i, seq_j) > max\{dist_k(seq_i), dist_k(seq_j)\}$, seq_i and seq_j are not k-NN of each other.

In summary by Theorem 3, as soon as the algorithm detects a row p in the recurrence table such that $\frac{min\_row(p)}{max\{\|seq_i\|, \|seq_j\|\}} > max\{dist_k(seq_i), dist_k(seq_j)\}$, it is clear that $dist'(seq_i, seq_j) = dist'(seq_j, seq_i) = \infty$. At this point, the recurrence table calculation can stop and simply return ∞. Checking for the condition $\frac{min\_row(p)}{max\{\|seq_i\|, \|seq_j\|\}} > max\{dist_k(seq_i), dist_k(seq_j)\}$ at the end of each row takes negligible time and space when $k \ll N$ and $k \ll L$.

3.3 Optimizing the Clustering Method

Now, we investigate how to reduce the N_{seq}^2 cell calculations by using an iterative clustering method similar to the well known k-mediods clustering methods. k-mediods clustering is exactly the same as the more popular k-means algorithm, except it works with the representative points in clusters rather than the means

Algorithm 1 (Sample Based Iterative Clustering)
Input: *a set of sequences* $\mathcal{D} = \{seq_i\}$*, the sampling percentage* α*, and the number of neighbor sequences* k' *for the sampled database;*
Output: *a set of clusters* $\{C_j\}$*, where each cluster is a set of sequences;*
Method:
1. **Randomly sample the database** $\mathcal{D}$ **into** $\mathcal{D}'$ **using** α*. The size of* $\mathcal{D}'$ *will be a trade off between time and accuracy. The experiments indicate that at a minimum* $\|\mathcal{D}'\|$ *should be 4000 sequences for the default* $k' = 3$*. Furthermore, roughly* 10% *of the data will give comparable results when* $N_{seq} \geq 40,000$*.*
2. **Run uniform kernel density based** k**-NN clustering [3] on** $\mathcal{D}'$ **with parameter** k'**.** *The output is a set of clusters* $\{C'_s\}$
3. **Center: Find the representative sequence for each cluster** C'_s*. The representative sequence,* seq_{sr}*, for a cluster,* C'_s*, is chosen such that* $\Sigma_j dist(seq_{sr}, seq_{sj})$ *for all sequences,* seq_{sj}*, in cluster* C'_s *is minimized (minimum intra-cluster distance).*
4. **Initialization: Initialize each cluster,** C_s**, with the representative sequence,** seq_{sr}*, found in the previous step.*
5. **Cluster: Assign all other sequences in the full database,** $\mathcal{D}$**, to the closest cluster.** *That is assign sequence* seq_i *such that* $dist(seq_i, seq_{sr})$ *is minimum over all representative sequences,* seq_{sr}*.*
6. **Recenter: Find the representative sequence for each cluster** C_s*. Repeat the centering step in 3 for all clusters* C_s *formed over the full database.*
7. **Iteratively repeat Initialization, Cluster, and Recenter.** *Steps 5 through 7 are repeated until no representative point change for any cluster or a certain iteration threshold,* $MAX\_LOOP = 100$*, is met.*

of clusters. There are two major difficulties in using the k-mediods clustering methods directly in ApproxMAP. First, without proper initialization, it is impossible to find the proper clusters. Thus, finding a good starting condition is crucial for k-mediods methods to give good results in terms of accuracy and speed [2]. Second, the general k-mediods method requires that the user input the number of clusters. However, the proper number of partitions is unknown in advance.

To overcome these problems, we introduce a *sample based iterative clustering* method. It involves two main steps. The first step finds the clusters and its representative sequences based on a small random sample of the data, $\mathcal{D}'$, using the density based k-NN method. Then in the second step, the number of clusters and the representative sequences are used as the starting condition to iteratively cluster and recenter the full database until the algorithm converges. The full algorithm is given above. When $\|\mathcal{D}'\| \ll \|\mathcal{D}\|$, the time complexity for clustering is obviously $O(t \cdot N_{seq})$ where t is the number of iterations needed to converge. The experimental results show that the algorithm converges very quickly. Figure 1(a) shows that in most experiments it takes from 3 to 6 iterations.

When using a small sample of the data, k (for k-NN algorithm) has to be smaller than what is used on the full database to achieve the clustering at the same resolution because the k-NN in the sampled data is most likely $(k+\alpha)$-NN in the full database. In ApproxMAP, the default value for k is 5. Hence, the default for k' in the sample based iterative clustering method is 3.

4 Evaluation

In our previous work, we have developed a benchmark that can quantitatively assess how well different sequential pattern mining methods can find the embedded

patterns in the data [4]. In this section, we apply the benchmark to conduct an extensive performance study on the two optimizations.

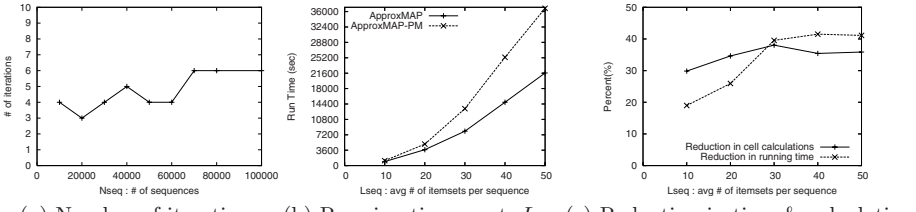
The benchmark uses the well known IBM data generator [1] which allows us to study the performance systematically. In addition, the IBM data generator embeds base patterns that represent the underlying trend in the data. By matching the results back to these embedded patterns, the benchmark can be used to measure the loss in accuracy due to the optimization. In particular, *recoverability* provides a good estimation of how well the items in the base patterns were detected. Recoverability is comparable to the commonly used recall except that it weights the results by the strength of the patterns in the database.

Most other criteria were not influenced by the optimizations. As expected in any alignment model, in all experiments there were no spurious patterns and negligible number of extraneous items resulting in excellent precision close to 100%. The amount of redundant patterns in the results remained similar to that of the basic ApproxMAP algorithm. The only criteria that was affected was the total number of patterns returned. Not surprisingly, recoverability is a good indicator for the number of total patterns returned increasing or decreasing accordingly. Thus, for simplicity we only report recoverability in our results.

4.1 Proximity Matrix Calculations

ApproxMAP can be optimized with respect to $O(L_{seq}^2)$ by calculating the proximity matrix used for clustering to only the needed precision. Here we study the speed up gained empirically. We only need to study the reduction in running time because this first optimization maintains the results of ApproxMAP. Figure 1(b) shows the speed up gained by the optimization with respect to L_{seq} in comparison to the basic algorithm. The figure indicates that such optimization can reduce the running time to almost linear with respect to L_{seq}.

To investigate the performance further, we examined the actual number of cell calculations reduced by the optimization. That is, with the optimization, the modified proximity matrix has mostly values of ∞ because $k \ll N$. For those $dist'(seq_i, seq_j) = \infty$, we investigated the dynamic programming calculation for $dist'(seq_i, seq_j)$ to see how many cells in the recurrence table were being skipped. To understand the savings in time, we report the following in Figure 1(c).



(a) Number of iterations (b) Running time w.r.t. L_{seq} (c) Reduction in time & calculation

Fig. 1. Results of optimizing the proximity matrix calculation (ApproxMAP-PM)

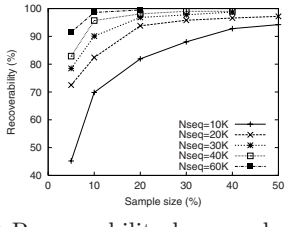
$$\frac{\sum \text{the number of cells in the recurrence table skipped}}{\sum \text{the total number of cells in the recurrence table}} \cdot 100\%$$

When $10 \leq L_{seq} \leq 30$, as L_{seq} increases more and more proportion of the recurrence table calculation can be skipped. Then at $L_{seq} = 30$, the proportion of savings levels off at around 35%-40%. This is directly reflected in the savings in running time in Figure 1(c). Figure 1(c) reports the reduction in calculations and running time due to the optimization as a proportion of the original algorithm. Clearly, the proportion of savings increase until $L_{seq} = 30$. At $L_{seq} = 30$ the running time levels off at around 40%. Thus, we expect that when $L_{seq} \geq 30$, the optimization will give a factor of 2.5 speed up in running time. This is a substantial improvement in speed without any loss in accuracy of the results.

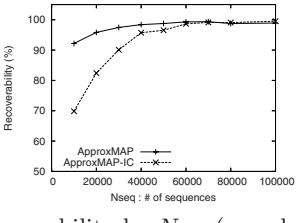
4.2 Sample Based Iterative Clustering

The sample based iterative clustering method can optimize the time complexity with respect to $O(N_{seq}^2)$ at the cost of some reduction in accuracy and larger memory requirement. The larger the sample size the better the accuracy with slightly longer running time. We investigate the tradeoff empirically.

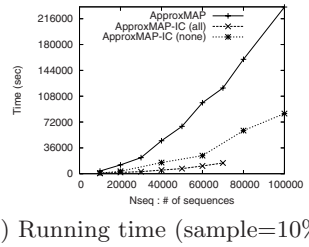
Figure 2(a) presents recoverability with respect to sample size ($k' = 3$). When $N_{seq} \geq 40,000$, recoverability levels off at 10% sample size with good recoverability at over 90%. When $N_{seq} < 40,000$, ApproxMAP requires a larger sample size of 20%-40%. In summary, the experiment suggests that the optimization should be used for databases when $N_{seq} \geq 40,000$ with sample size 10%. For databases with $N_{seq} < 40,000$ a larger sample size is required as 10% will result in significant loss in accuracy (Figure 2(b)). Essentially, the experiments indicate that the sample size be at least 4000 seqs to get comparable results when $k' = 3$.



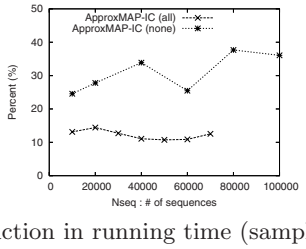
(a) Recoverability by sample size



(b) Recoverability by N_{seq} (sample=10%)



(c) Running time (sample=10%)



(d) Reduction in running time (sample=10%)

Fig. 2. Results for sample based iterative clustering (ApproxMAP-IC)

In the iterative clustering method more memory is required in order to fully realize the reduction in running time because the N_{seq}^2 proximity matrix needs to be stored in memory across iterations. In the basic method, although the full proximity matrix has to be calculated, the information can be processed one row at a time and there is no need to return to any values. That is, we only need to maintain the k-NN list without keeping the proximity matrix in memory. However in the iterative clustering method, it is faster to store the proximity matrix in memory over different iterations so as not to repeat the distance calculations. When N_{seq} is large, the proximity matrix is huge. Hence, there is a large memory requirement for the fastest optimized algorithm.

Nonetheless, the proximity matrix becomes very sparse when the number of clusters is much smaller than N_{seq}. Thus, much space can be saved by using a hash table instead of a matrix. Furthermore, a slightly more complicated scheme of storing only up to the possible number of values and recalculating the other distances when needed (much like a cache) will still reduce the running time compared to the basic method. Efficient hash tables are a research topic on its own and will be studied in the future. For now, the initial implementation of a simple hash table demonstrates the huge potential for reduction in time well. In order to fully understand the potential, we measured the running time assuming (1) memory was limitless and (2) no memory was available to store the proximity matrix. That is, distance values were never recalculated in the first experiment and always recalculated in the second experiment.

Figure 2(b) and (c) show the loss in recoverability and the gain in running time with respect to N_{seq} with the optimization (sample size=10%, $k' = 3$). Figure 2(d) depicts the relative running time with respect to N_{seq}. *optimized (all)* is a simple hash table implementation with all proximity values stored and *optimized (none)* is the implementation with none of the values stored. The implementation of a simple hash table was able to run up to $N_{seq} = 70,000$ with 2GB of memory (Figures 2(c) and 2(d) - *optimized (all)*). A more efficient hash table could easily improve the memory requirement.

A good implementation would give running times in between the *optimized (all)* and the *optimized (none)* line in Figure 2(c). The results clearly show that the optimization can speed up time significantly at the cost of negligible reduction in accuracy. Figure 2(d) show that the optimization can reduce running time to roughly 10%-40% depending on the size of available memory. Even in the worst case the running time is significantly faster by a factor of 2.5 to 4. In the best case, the running time is an order of magnitude faster.

5 Conclusions

Optimizing data mining methods is important in real applications which often have very large databases. In this paper, we proposed two optimization techniques for ApproxMAP that can reduce the running time significantly for consensus sequential pattern mining based on sequence alignment.

References

1. R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, pp. 3-14, 1995
2. A. Goswami, R. Jin, and G. Agrawal. Fast and Exact Out-of-Core K-Means Clustering. In *Proc. of the Int'l Conference on Data Mining (ICDM)*, pp. 83–90, 2004.
3. H.C. Kum, J.H. Chang, and W. Wang. Sequential pattern mining in multi-databases via multiple alignment. In *DMKD*, 12(2-3), pp. 151-180, 2006.
4. H.C. Kum, J.H. Chang, and W. Wang. Benchmarking the effectiveness of sequential pattern mining methods. In *Data and Knowledge Engineering*, 60, pp30-50, 2007.
5. H.C. Kum, J. Pei, W. Wang, and D. Duncan. ApproxMAP : Approximate mining of consensus sequential patterns. In *Proc. of SDM*, pp. 311-315, 2003.
6. A. Marascu and F. Masseglia. Mining data streams for frequent sequences extraction. In *Proc. of the IEEE Workshop on Mining Complex Data (MCD)*, 2005.
7. P. Tzvetkov, X. Yan, and J. Han. TSP: Mining top-k closed sequential patterns. In *Proc. of the Int'l Conference on Data Mining (ICDM)*, pp. 418–425, 2003.
8. X. Yan, J. Han, and R. Afshar. CloSpan: Mining closed sequential patterns in large datasets. In *Proc. of the SIAM Int'l Conf on Data Mining*, pp. 166-177, 2003.

A Hybrid Prediction Method Combining RBF Neural Network and FAR Model

Yongle Lü and Rongling Lang

School of Electronic and Information Engineering,
Beijing University of Aeronautics and Astronautics,
100083, Beijing, China
Lv_yongle@ee.buaa.edu.cn,
ronglinglang@163.com

Abstract. The classical autoregressive moving average model (ARMA) fails to satisfy the high request for precision in predicting nonlinear and nonstationary systems. Overcoming the difficulty, a hybrid prediction method is proposed in this paper, which organically couples the radial basis function prediction neural network (RBFPNN) and the functional-coefficient autoregressive prediction model (FARPM). An observation time series characterized by nonlinearity and nonstationarity can be technically decomposed with the wavelet analysis tool into two clusters of sequences, i.e. the smooth sequences and the stationary sequences, which can be effectively predicted with RBFPNN and FARPM respectively. Then, the integrated prediction is obtained by merging the results of RBFPNN and FARPM. It's indicated by the simulation that the prediction precision for one step, 4 steps and 12 steps can be improved at least by 41%, 60% and 60% respectively, compared to the prediction with ARMA, RBFPNN and FARPM separately.

Keywords: Nonlinear and nonstationary system; time series; prediction; radial basis function neural network; functional-coefficient autoregressive model.

1 Introduction

A time series is a sequence of observations taken sequentially in time [1]. In recent years, the prediction based on *time series analysis* has been popularly applied in the fields of biology, weather, economy, traffic, industry and other fields.

The actual observation time series from all kinds of applications such as stock price analysis, power load supervision and mechanical vibration monitoring, are usually characterized by nonlinearity and nonstationarity. The classical ARMA modeling and forecasting theory fails to deal with them with preferable performance [2].

Radial Basis Function Neural Network (RBFNN) is a local approximation neural network, which can model the inherent connections of training data and has favorable self-adaptive capability. Owning to the prominent advantages of simple structure, fast convergence and high prediction precision, RBFNN was efficiently applied to predict

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 598–605, 2007.

nonlinear systems as a parallel processing tool [3]. However, the weakness [4] in approximating non-smooth functions limits its popularization in dynamic systems.

Functio nal-coefficient Autoregressive model (FAR), which is a nonparametric model in statistics, has succeeded in modeling and analyzing the stationary nonlinear time series [5]. Compared to ARMA model, it's able to avoid effectively the deviation in modeling, and then reduce the prediction error caused by the model's unsuitability. However, it has limitation in dealing with a nonstationary time series.

The Hybrid Prediction combining RBF Neural Network and FAR model (HP-RBFNN&FAR), makes entire use of the prominence of RBFNN in predicting a smooth nonlinear time series and the superiority of FAR in predicting a stationary nonlinear time series, letting them work complementally and cooperatively. The basic idea is that the decomposition of the observation time series into several sequences, which can be separately predicted with their most suitable prediction methods, makes a good prediction. Here, the decomposition is based on the partition of spectrum.

2 RBF Prediction Neural Network (RBFPNN)

2.1 Structure of RBFPNN

A single variable time series $\{x_1, x_2, ..., x_n\}$ can be predicted with the neural network on the following hypothesis: the observation x_{t+k} at a future time $t+k$ is the function of the m available observations $\{x_t, x_{t-1}, ...x_{t-m+1}\}$ at the current time t, which can be expressed by the equation

$$x_{t+k} = F(x_t, x_{t-1}, ...x_{t-m+1}) \ . \tag{1}$$

Where, k ($k \geq 1$) is the prediction step ahead, F is a $\mathbf{R}^m \mapsto \mathbf{R}$ continuous function.

Letting $X_t = (x_t, x_{t-1}, ...x_{t-m+1})^{\mathrm{T}}$, $y_t(k) = x_{t+k}$, the equation (1) can also be written as

$$y_t(k) = F(X_t) \ . \tag{2}$$

RBFPNN is a 3 layer feed-forward neural network, as shown in Fig.1. The first layer is input layer, which has the m-dimension vector X_t ($t = m, m+1, ..., n$) as its input and connects with the hidden layer via unitary weights. The activation functions on the neurons of hidden layer are locally distributed nonlinear functions called *Radial Basis Functions (RBF)*, which symmetrically attenuate in the radial direction off centers. $\psi_j(X_t)$ is the RBF acting on the jth hidden neuron, which usually adopts Gauss function

$$\psi_j(X_t) = \exp(-\frac{1}{2\sigma_j^2}\left\|X_t - \mathbf{c}_j\right\|^2) \ . \tag{3}$$

Where, $\mathbf{c}_j$ which has the same dimension as X_t is the center of $\psi_j(\cdot)$; σ_j is the shape parameter of $\psi_j(\cdot)$, also called *RBF bandwidth*. M is the number of hidden neurons,

and ω_j is the connecting weight between the jth hidden neuron and the output neuron. The last layer is called *output layer*, whose output is given by

$$F(X_t) = \sum_{j=1}^{M} \omega_j \psi_j(X_t) \ . \tag{4}$$

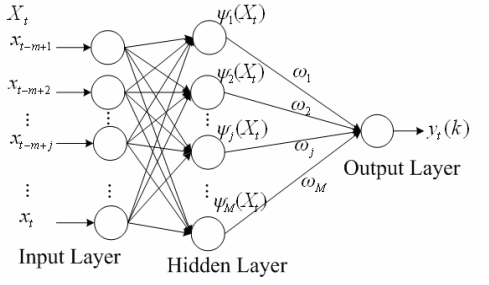


Fig. 1. Structure of RBFPNN

2.2 Training of RBFPNN

Training RBFPNN is the process to select the optimal network structure, determine the parameters $\mathbf{c}_j$, σ_j of $\psi_j(\cdot)$ and calculate the connecting weights ω_j ($j = 1,...M$). The improved RBFNN training algorithms [6] can also efficiently train RBFPNN.

How to determine the optimal dimension m of X_t is an interest of research. In the paper, for cases of short-term or medium-term prediction ($k \le 4$), an empirical method based on the Partial Autocorrelation Function (PACF) is put forward, i.e. $m=L_{max}+k$. Here, L_{max} is the largest of the lags, corresponding to which the PACF values of $\{x_1, x_2, ..., x_n\}$ are far more than a threshold [7] given by $\tau = 2/\sqrt{n}$.

3 FAR Prediction Modeling

3.1 FAR Model

Considering a single variable time series $\{x_1, x_2, ..., x_n\}$, the functional-coefficient autoregressive model FAR(p,d) (p, d are the model parameters) admits the form [5]

$$x_t = f_1(x_{t-d}) \cdot x_{t-1} + \cdots + f_p(x_{t-d}) \cdot x_{t-p} + \sigma(x_{t-d}) \cdot \varepsilon_t, (t = p+1,...,n; \quad d > 0) \ . \tag{5}$$

Where, $\{\varepsilon_t\}$ is an independent and identically distributed random variable sequence with zero mean and unity variance, and ε_t is independent of x_{t-j} ($j = 1,...p$). x_{t-d} is the model-dependent variable. The coefficient functions $f_j(\cdot)$ ($j = 1,...p$) and $\sigma(\cdot)$, which

are unknown continuous functions, can be estimated by using the local linear regression technique [8].

3.2 Factors Influencing on the Performance of FAR Prediction Model

Based on model (5), the one-step-ahead FAR Prediction Model (FARPM) is given by

$$\bar{x}_{t+1} = E_t(x_{t+1}) = \hat{f}_1(x_{t-d})x_t + ... + \hat{f}_p(x_{t-d})x_{t-p+1} \ . \tag{6}$$

Where, $\hat{f}_j(\cdot)$ is the estimator for $f_j(\cdot)$. $E_t(\cdot)$ is the *conditional expectation* at time t. As for the multiple-step-ahead predictor, the method of iteration is employed.

Since the coefficient functions' estimators $\hat{f}_j(\cdot)$ ($j = 1,...,p$) are influenced by the kernel function [9] $K(\cdot)$ and its bandwidth [10] b, $K(\cdot)$ and b, together with model parameters p, d are the influence factors on the performance of FARPM.

Overall Average Prediction Error (APE) introduced in the reference [11] is adopted to evaluate the performance of FARPM, which is the function of b, p and d, i.e. $APE(p,d,b)$. The optimal bandwidth b, order p and lag d can be obtained by minimizing $APE(p,d,b)$ simultaneously for b in a certain range, d over the set $\{1,2,\cdots,p\}$ and p over the set $\{1,2,\cdots,n\}$.

4 Hybrid Prediction Combining RBFNN and FAR

Usually, it is difficult to get a satisfactory result with RBFPNN or FARPM individually, when predicting an observation time series characterized simultaneously by nonlinearity (such as nonnormality, asymmetric cycles, bimodality, and others), nonstationarity and larger instability. However, HP-RBFNN&FAR can make a success with RBFPNN and FARPM complementally and cooperatively combined, overcoming their respective deficiencies in approximating non-smooth functions which describe dynamic systems and modeling a nonstationary time series.

As shown in Fig.2, the process of HP-RBFNN&FAR can be briefly summarized as follows: Firstly, spectrum of the nonstationary nonlinear observation time series is analyzed, and wavelet analysis tool [12] is applied to construct the optimal scale filters. Secondly, through scale filters, the observation time series is decomposed into $t+s$ sequences, classed into two clusters. One characterized by lower frequency includes t ($t \geq 1$) smooth trend sequences, while the other characterized by higher frequency includes s ($s \geq 1$) stationary sequences. Thirdly, the sequences characterized by smoothness and reflecting the trend can be precisely predicted with RBFPNN, while the stationary sequences fluctuating randomly at zero can be well modeled and predicted with FARPM. Lastly, the integrated prediction is obtained by merging the prediction results of RBFPNN and FARPM.

In the scheme of HP-RBFNN&FAR, the partition of spectrum directly affects the total prediction performance. So selection of wavelets, construction of scale filters and determination of t and s are all important things for HP-RBFNN&FAR.

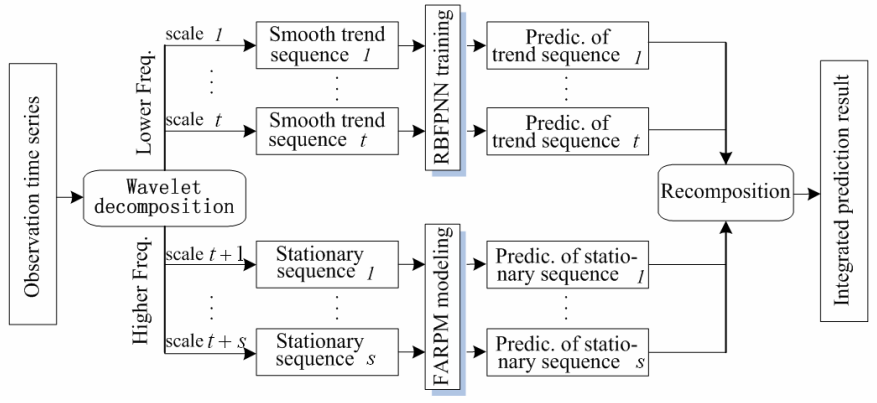


Fig. 2. Scheme of HP-RBFNN&FAR

5 Simulation

A technically constructed simulation model is employed to generate a nonstationary nonlinear observation time series with the length of 600 to analyze the capability of HP-RBFNN&FAR, which simultaneously contains trend elements, cycle elements and random elements just as most observation time series (e.g. the daily observations of water level, air temperature, insect population and stock turnover, hourly vehicle flowrate observations and minutely power load observations) do in actual applied areas such as hydrology, weather, biology, economics, traffic and industry.

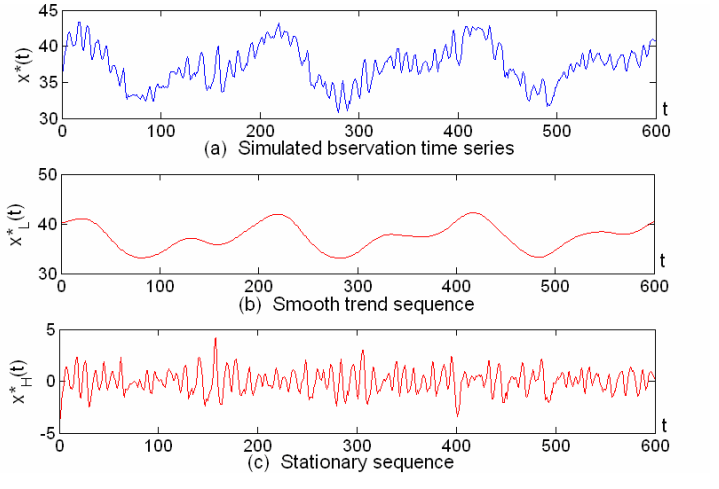


Fig. 3. Decomposition of the simulated observation time series

As can be seen in Fig.3, the simulated observation time series $\{x^*(t)\}$ can be decomposed into two parts of a smooth trend sequence $\{x_L^*(t)\}$ and a stationary sequence $\{x_H^*(t)\}$ by the well constructed filter with the Daubechies wavelet.

For each sequence, the first 500 data defined as training samples are used to train RBFPNN or determine the optimal parameters of FARPM, while the last 100 data defined as testing samples are used to evaluate the prediction performance. Here, *Mean Squared Error* (MSE) is adopted as a metric on prediction precision.

For $\{x_L^*(t)\}$, letting the prediction step $k=1$, the optimal dimension of input vector is obtained ($m=4$) with the proposed method in Section 2.2. The testing samples of $\{x_L^*(t)\}$ is predicted with the well trained RBFPNN, as illustrated in Fig. 4(a). The Mean Squared Error of One-Step-ahead Prediction (OSPMSE) is 9.1093×10^{-5}.

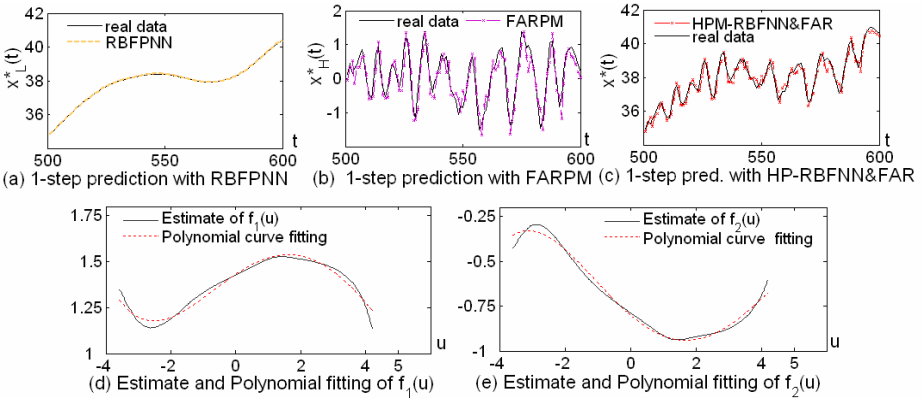


Fig. 4. Prediction with HP-RBFNN&FAR

For $\{x_H^*(t)\}$, using Epanechnikov kernel function: $K(u)=0.75(1-u^2)_+$. The optimal parameters can be obtained ($p=2$, $d=2$, $b=2.975$) by minimizing $APE(p,d,b)$. The estimated coefficient functions $\hat{f}_j(\cdot)$ ($j=1,2$) and their polynomial-fitting curves $\tilde{f}_j(\cdot)$ are shown in Fig. 4 (d) and (e). The 1-step FARPM for $\{x_H^*(t)\}$ is expressed by

$$\tilde{x}_H^*(t+1)=\tilde{f}_1(u)x_H^*(t)+\tilde{f}_2(u)x_H^*(t-1). \tag{7}$$

Where, t is the current time, $u=x_H^*(t-2)$ is the model-dependent variable, and the polynomial-fitting curves can be written as

$$\tilde{f}_1(u)=0.001u^4-0.0085u^3-0.0209u^2+0.1175u+1.4312, \tag{8}$$

$$\tilde{f}_2(u)=-0.0012u^4+0.0075u^3+0.0332u^2-0.1558u-0.7989. \tag{9}$$

With FARPM, the one-step-ahead predictor is got with OSPMSE 0.0959, as shown in Fig. 4(b). Finally the integrated prediction result is obtained with OSPMSE 0.0960.

Table 1. Comparison of the prediction performance of different methods. The data marked with stars in the brackets for the 4-step-ahead prediction and the 12-step-ahead prediction.

Prediction method	Optimal parameters	MSE		
		1-step	4-step	12-step
ARMA(p, q)	p=6,q=4	0.2734	1.7122	17.8989
RBFPNN	m=6(9*),M=475	0.7166	1.3214	2.3721
FARPM	p=2,d=2,b=1.025	0.1842	3.2561	15.4294
HP-RBFNN&FAR	m=4(7*),M=25(100*), p=2,d=2,b=2.975	0.0960	0.5297	0.9493

Simulation is also performed to compare the performance of different prediction methods. From Table 1, it is indicated that the prediction precision for one step, 4 steps and 12 steps (standing respectively for short term, medium term and long term) ahead with the proposed prediction method can be improved at least by 41%, 60% and 60% respectively, compared to the methods with ARMA, RBFPNN and FARPM.

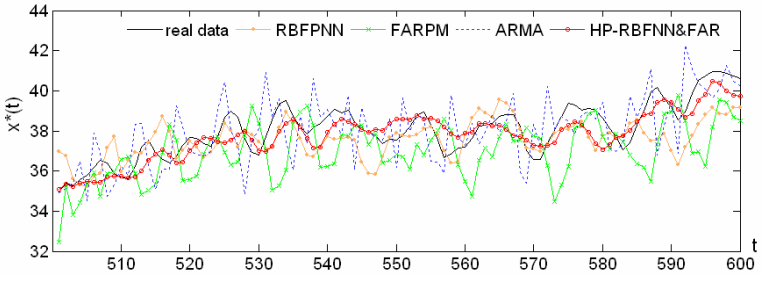


Fig. 5. Four-step-ahead prediction with the different prediction methods

6 Conclusion

In the paper, a hybrid prediction method combining organically RBFPNN and FARPM is put forward, with the algorithm of which is discussed. The proposed hybrid prediction method succeeds in predicting a nonlinear and nonstationary time series by combining the respective algorithms of RBFPNN and FARPM in a complemental and cooperative way. Compared to the prediction with ARMA, RBFPNN or FARPM, HP-RBFNN&FAR is deemed to result in a better accuracy in actual applied areas.

Another important finding of the research is that the total prediction performance is directly affected by the partition of spectrum. Thus how to select the suitable wavelet to construct optimal filters and make RBFPNN and FARPM more efficiently combined is the future work.

References

1. George E.P.Box, Gwilym M.Jenkins, Gregory C.Reinsel: Time Series Analysis: Forecasting and Control, 3rd Edition. Pearson Education Asia Ltd. (2005)
2. Fan, J., Yao, Q.: Nonlinear Time Series: Nonparametric and Parametric Methods. Springer Science+Business Media, Inc.(2005)
3. Wang L., Zheng Y., Pan S.: Intelligent Control Algorithm for VLSI Manufacturing Line Based on RBFNN Predictive Model. Control and Decision, Vol.21, No.3 (2006) 336-338,351
4. Robert J.,Schilling J. and Carroll J.: Approximation of nonlinear systems with radial basis function neural networks. IEEE Trans. On Neural Networks, Vol.12, No.1 (2001) 21-28
5. Chen, R., Tsay, R.S: Functional-coefficient autoregressive models. Journal of American Statistical Association, Vol.88 (1993) 298-308
6. Mehdi F., Mehdi R., Faridoon S.: New Training Methods for RBF Neural Networks, IEEE (2005) 1322-1327
7. Jenkins, G.M.: Tests of hypotheses in the linear autoregressive model, II. Biometrika,Vol.43 (1956) 186-199
8. Fan, J., Gijbels, I.: Local Polynomial Modeling and Its Applications.Chapman and Hall, London (1996)
9. Fan, J., Gasser, T., Gijbels, I., and etal: Local polynomial fitting: optimal kernel and asymptotic minimax efficiency. Annals of the Institute of Statistical Mathematics, Vol.49 (1996) 79-99
10. Fan,J., Gijbels, I.: Data-driven bandwidth selection in local polynomial fitting: variable bandwidth and spatial adaptation. Journal of the Royal Statistical Society, Series B, Vol.57 (1995) 371-394
11. Cai,Z., Fan,J., Yao,Q: Functional-coefficient regression models for nonlinear time series. Journal of American Statistical Association, Vol.95 (2000) 941-956
12. Donald B. Percival, Andrew T. Walden: Wavelet Methods for Time Series Analysis. Cambridge University Press (2000)

An Advanced Fuzzy C-Mean Algorithm for Regional Clustering of Interconnected Systems

Sang-Hyuk Lee, Jin-Ho Kim*, Se-Hwan Jang, Jong-Bae Park, Young-Hwan Jeon, and Sung-Yong Sohn

School of Mechatronics Engineering, Changwon National University, Changwon 641-773, Korea
School of Electrical Engineering, Kyungwon University, Seongnam 461-701, Korea
School of Electrical Engineering, Pusan National University, Pusan 609-735, Korea
School of Electrical Engineering, Konkuk University, Seoul 143-701, Korea
School of Electrical Engineering, Hongik University, Seoul 121-791, Korea

Abstract. An advanced fuzzy C-mean(FCM) algorithm for the efficient regional clustering of multi-nodes interconnected systems is presented in this paper. Owing to physical characteristics of the interconnected systems, nodes or points in the interconnected systems have their own information indicating the network-related characteristics of the system. However, classification for the whole system into distinct several subsystems based on a similarity measure is typically needed for the efficient operation of the whole system. In this paper, therefore, a new regional clustering algorithm for interconnected systems based on the modified FCM is proposed. Moreover, the regional information on the system are taken into account in order to properly address the geometric mis-clustering problem such as grouping geometrically distant nodes with similar measures into a common cluster. We have presented that the proposed algorithm has produced proper classification for the interconnected system and the results are demonstrated in the example of IEEE 39-bus interconnected electricity system.

Keywords: Fuzzy C-mean, similarity measure, interconnected power systems.

1 Introduction

In the regional management of interconnected network systems, the efficient and economical operation of the networked systems in terms of system coherency is essential. Hence the research of system coherency has been made by numerous researchers [1-4]. However, most of the studies are focused on the dynamic grouping. At this point, we need a novel approach to partition the total system into several regions considering locational information, such as locational cost, loss, regional distances, and so on. In this paper, grouping the locations in a networked system with similar locational prices has been proposed considering the regional coherency.

* Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 606–615, 2007.

Locational prices in a networked system implies the price at which the good is consumed at each location. Due to the physical characteristics of the transmission network of the systems, the good is lost as it is transmitted from supplying locations to consuming locations, and an additional supply must be provided to compensate the loss. Also, the transmission network of the systems has a capacity limitation preventing full uses of cheap production. Therefore, location prices at each point or node, is differently decided depending the network topology and supply/demand configuration. Similarity measure has been known as the complementary meaning of the distance measure [5-9]. Hence, we consider the partitioning measure not only similarity measure but also regional information, that is, distance measure. In the previous literatures, we had constructed similarity measure through distance measure or fuzzy entropy function [10]. Well known-Hamming distance was used to construct similarity measure. With only similarity measure, we can obtain unpractical results, which partition physically distant locations into the same group. Hence we add the regional information to complete modified similarity measure. In the next section, FCM and similarity measure are introduced and proved. In Section 3, similarity measures with distance measure is introduced and modified with additional regional information. In Section 4, illustrative examples are shown. In the example, we obtain a proper partitioning result, which consider both similarity and regional information. Conclusions are followed in Section 5. Notations of Liu's are used in this paper [5].

2 Fuzzy C-Means Clustering and Similarity Measure

Fuzzy C-means clustering was proposed by Bezdek in 1973 as an improvement over HCM(Hard C-means)[10]. FCM play a roll of partitioning arbitrary n vectors into c fuzzy groups, also it finds a cluster center for each group such that a cost function of similarity measure is maximized, or dissimilarity measure is minimized. Well known fact about FCM and HCM indicates that FCM employs fuzzy partitioning such that a data point can belong to several groups with the degree of membership grades between 0 and 1.

2.1 Preliminaries

We will illustrate the FCM result briefly [11]. Membership matrix U is satisfied as follows

$$\sum_{i=1}^{c} u_{ij} = 1, \ \forall j = 1,....,n \tag{1}$$

The cost function for FCM is constructed by

$$J(U, c_1,...,c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 \tag{2}$$

Where u_{ij} is between 0 and 1, c_i is the center of fuzzy group i, $d_{ij} = |c_i - x_j|$ is the Euclidean distance between i-th cluster center and the j-th data point x_j, and m is the weighting value. With Lagrange multiplier, the necessary conditions for (2) to reach a minimum are [11]

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \text{ and } u_{ij} = \frac{1}{\sum_{k=1}^{c} (\frac{d_{ij}}{d_{kj}})^{2/(m-1)}}$$

With these results, well known FCM algorithms are listed below:

Step 1: Initialize membership matrix U
Step 2: Calculate $c_i, i = 1, \cdots, c$
Step 3: Compute (2). Stop if either it is below a certain tolerance
Step 4: Compute a new U.

Now for minimizing of (2), the less distance is $d_{ij} = |c_i - x_j|$, the smaller cost function become. Hence distance means the similarity between two data points. Finding similarity is determined from the types of data, time series signal, image, sound, *etc*.. Now we need proper similarity measure. In this subsection, we introduce a similarity measure for the fuzzy sets. And the proposed similarity measure can be applied to our problem.

2.2 Similarity Measure with Distance Function

We define modified similarity measure, which is different from that of Liu's.

Definition 2.1: A real function $s: P^2 \to R^+$ or $F^2 \to R^+$ is a modified similarity measure for regional point, if s has the following properties :

(S1) $s(A,B) = s(B,A), \forall A, B \in P(X)$ or $F(X)$

(S2) $s(A,A^c)$ satisfies minimum value, $\forall A \in P(X)$ or $F(X)$, where A^c is the farthest point from A

(S3) $s(D,D) = \max_{A,B \in P} s(A,B), \forall A, B \in P(X)$ or $F(X)$

(S4) $\forall A, B, C \in P(X)$ or $F(X)$,if $A \subset B \subset C$, then $s(A,B) \geq s(A,C)$ and $s(B,C) \geq s(A,C)$.

With Definition 2.1, we propose the following theorem as the modified similarity measure.

Theorem 2.1. For any set $A, B \in F(X)$ or $P(X)$ if d satisfies Hamming distance measure, then

$$s(A,B) = 4 - 2d((A \cap B),[1]) - 2d((A \cup B),[0]) \tag{3}$$

is the similarity measure between set A and B.

Proof. We prove that the eq. (3) satisfies the similarity definition. (S1) means the commutativity of set A and B, hence it is clear from (3) itself. From (S2),

$$s(A, A^c) = 4 - 2d((A \cap A^c),[1]) - 2d((A \cup A^c),[0])$$

then $2d((A \cap A^c),[1])$ and $2d((A \cup A^c),[0])$ are the maximum values of between A and arbitrary set. For arbitrary sets A, B, inequality of (S3) is proved by

$$\begin{aligned} s(A,B) &= 4 - 2d((A \cap B),[1]) - 2d((A \cup B),[0]) \\ &\le 4 - 2d((D \cap D),[1]) - 2d((D \cup D),[0]) \\ &= s(D,D). \end{aligned}$$

Inequality is satisfied from $d((A \cap B),[1]) \ge d((D \cap D),[1])$ and $d((A \cup B),[0]) \ge d((D \cup D),[0])$.

Finally, (S4) is $\forall A, B, C \in F(X), A \subset B \subset C$,

$$\begin{aligned} s(A.B) &= 4 - 2d((A \cap B),[1]) - 2d((A \cup B),[0]) \\ &= 4 - 2d(A,[1]) - 2d(B,[0]) \\ &\ge 4 - 2d(A,[1]) - 2d(C,[0]) \\ &= s(A,C), \end{aligned}$$

similarly $s(B,C) \ge s(A,C)$ is obtained through $d(B,[0]) \le d(C,[0])$ and $d(B,[1]) \le d(A,[1])$.

Therefore proposed similarity measure (3) satisfy modified similarity measure. Similarly, we propose another similarity measure in the following theorem.

Theorem 2.2. For any set $A, B \in F(X)$ or $F(X)$, if d satisfies Hamming distance measure, then

$$s(A,B) = 2 - 2d((A \cap B^c),[0]) - 2d((A \cup B^C),[1]) \tag{4}$$

is the similarity measure between set A and set B.

Proof. Proofs are shown similarly as Theorem 2.1.

3 New Similarity with Regional Information

In the previous section we have derived the modified similarity measures which satisfying the definition of similarity. To apply FCM with $d_{ij} = |c_i - x_j|$, it is required that d_{ij} has to satisfy similarity property. Hence we can consider d_{ij} as the proposed modified similarity measure in (2). However proposed similarity measure can group for the point that having similar characteristic values. For the large scale system whose similar measure values are close, however they are located far away. Then it is not realistic to gather even though they have similar valued measure. So we need another characteristic values considering regional information. With (3) and (4), we consider

$$s_2(A,B) = 2/(1+\text{distance}) \tag{5}$$

where distance is the geometrical distance value.
We consider the combined similarity measure as

$$s(A,B) = \omega_1 s_1(A,B) + \omega_2 s_2(A,B) \tag{6}$$

where, $s_1(A,B) = 2 - 2d((A \cap B),[1]) - 2d((A \cup B),[0])$, and ω_1, ω_2 are the weighting values.

We can verify the usefulness of (6) as follows, properties of $s_1(A,B)$ are proved in Theorem 2.1 and 2.2. Usefulness for the similarity of $s_2(A,B)$ can be verified as follows:

Commutative values of distance are the same, hence (S1) is easily shown. From (S2), distance of A and A^c is the longest, hence $s_2(A,A^c)$ is the minimum value. For all $A,B \in P(X)$, inequality of (S3) is proved by

$$s_2(A,B) = 2/(1+\text{distance}(A,B)) \le 2/(1+\text{distance}(D,D)) = s_2(D,D).$$

In the above distance(D,D) is the smallest value, *i.e.*, zero. So (S3) can be verified.

Finally, (S4) is $\forall A,B,C \in P(X)$, and A,B,C satisfy triangular points, then

$$s_2(A,B) = 2/(1+\text{distance}(A,B)) \ge 2/(1+\text{distance}(A,C)) = s_2(A,C),$$

where distance(A,C) is longer than distance(A,B).

Similarly,

$$s_2(B,C) = 2/(1+\text{distance}(B,C)) \ge 2/(1+\text{distance}(A,C)) = s_2(A,C)$$

is satisfied. Hence we can verify that $s_2(A,B)$ satisfies the Definition 2.1.

Then, we use (6) as the modified similarity measure for the measuring of particular points which have characteristic values and regional information at the same time. Modified similarity measures are used in the following example.

4 Illustrative Example

With FCM, we replace $s(A,B)$ in (6) into $d_{ij} = |c_i - x_j|$ in (2), and illustrate the system which has characteristic values and regional information at the same time. As an illustrative application, we consider the interconnected electricity system. The IEEE reliability test system which is prepared by the reliability test system task force of the application of probability methods subcommittee on 1996 [12] is considered as a test system. In the test system, 39 nodes (buses) and 10 generators are contained and each bus has its own locational price and information.

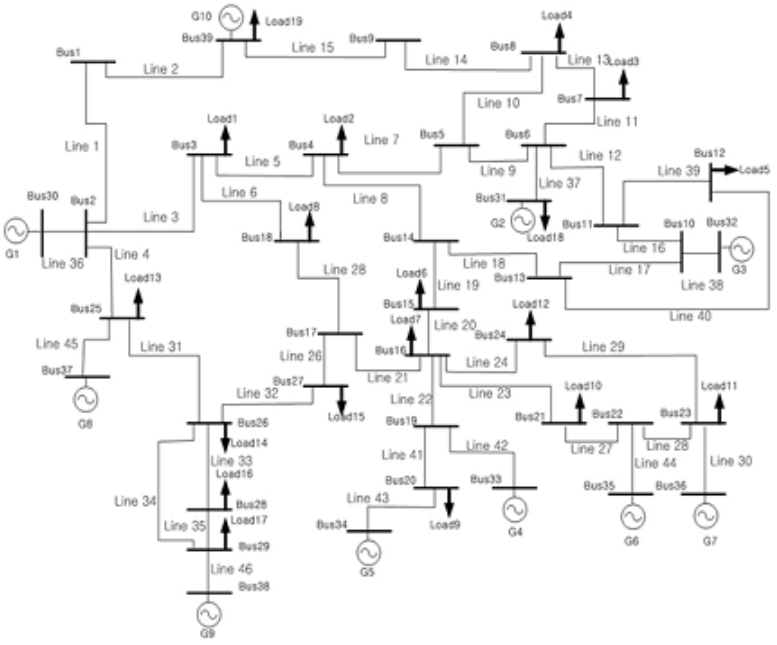


Fig. 1. A networked electricity system

In networked electricity systems, due to the physical characteristics of the electricity transmission network, electricity is lost when it is transmitted from supplying nodes (*i.e.*, supplying buses) to consuming nodes (consuming buses), and additional generation must be supplied to provide energy in excess of that consumed by customers. Moreover, the capacity limitation of the transmission network of electricity systems prevents full uses of system wide cheap electricity. Therefore, electricity price at each node, *i.e.*, the price at which the electricity is consumed at each node is differently decided depending on the network topology and energy configuration.

The electricity prices at each node is defined as locational prices at each node and the locational prices represent the locational value of energy, which includes the cost of electricity and the cost of delivering it, *i.e.*, the delivery losses and network congestion. In Table 1 each locational price per kWh are illustrated for the 39 buses, and per unit geometrical information for each nodes are also shown.

Locational prices of each nodes are from 28.53 to 55.00, and 39 locational information are represented through 2-dimensional plane at which plane is assumed to be flat. Considered combined similarity measures constitute as follows with the proposed measure:

$$s(A,B) = \omega_1 s_1(A,B) + \omega_2 s_2(A,B)$$

At first, we partitioned the 39 buses to the 3 groups, and the result is illustrated in Fig. 2. 39 buses are shown in 3 dimensional space, x-y plane is represented as the

locational information and height means the locational price. 3 dimensional 39 vectors are projected to the x-y plane, in Fig. 2 we obtain the result of with only locational information. Result shows that there are no changes with only locational consideration. This strict condition does not satisfies the user's request. Hence we will consider the locational price and locational information simultaneously.

Table 1. Locational prices and per unit locations at each node

Bus	Locational price ($/kWh)	Location (per unit)	Bus	locational price ($/kWh)	Location (per unit)	Bus	locational price ($/kWh)	Location (per unit)
BUS1	29.21	(0.9, 9)	BUS14	41.74	(6.6, 6)	BUS27	51.45	(4.6, 3.5)
BUS2	28.53	(0.6, 6.2)	BUS15	43.79	(6.6, 4.9)	BUS28	55.00	(2.7, 1.5)
BUS3	31.40	(3, 7.5)	BUS16	45.84	(6.5, 4)	BUS29	55.00	(2.7, 0.8)
BUS4	32.78	(4.7, 7.5)	BUS17	47.90	(5, 4.5)	BUS30	28.53	(0, 6.2)
BUS5	37.57	(7, 7.6)	BUS18	46.40	(4.2, 6)	BUS31	38.26	(8.3, 6.6)
BUS6	38.26	(8.5, 7.6)	BUS19	45.84	(6.9, 2.8)	BUS32	40.00	(11.3, 5.8)
BUS7	37.81	(9.6, 8.4)	BUS20	45.84	(6.9, 1.7)	BUS33	45.84	(8, 1.7)
BUS8	37.35	(8.5, 9.1)	BUS21	45.84	(8.7, 2.8)	BUS34	45.84	(5.5, 1)
BUS9	30.56	(6.1, 9.5)	BUS22	45.84	(10, 2.8)	BUS35	45.84	(10, 1.6)
BUS10	40.00	(10.8, 5.8)	BUS23	45.84	(11.1, 2.8)	BUS36	45.84	(11.1, 1.6)
BUS11	39.42	(9.7, 6.3)	BUS24	45.84	(8.2, 4.3)	BUS37	24.98	(0.7, 3.7)
BUS12	40.00	(11.1. 7.1)	BUS25	24.98	(1.4, 4.7)	BUS38	55.00	(2.7, 0)
BUS13	40.58	(8.5, 5.5)	BUS26	55.00	(2.7, 3)	BUS39	29.88	(3.4, 9.5)

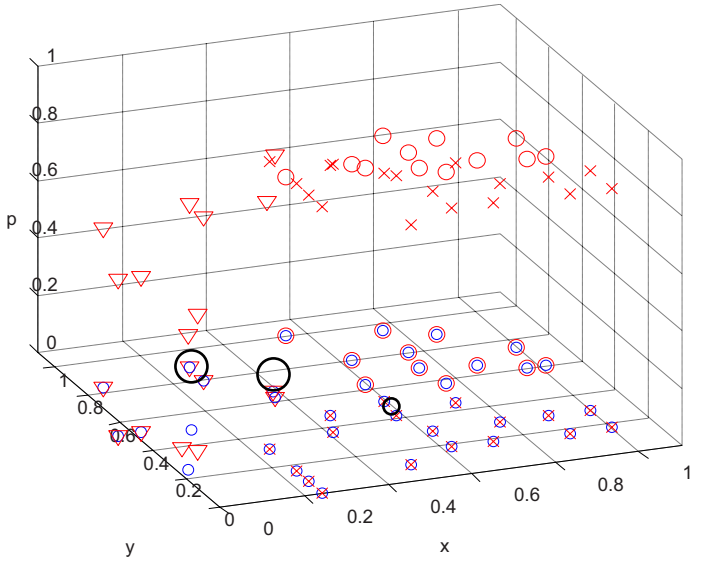


Fig. 2. Clustering by FCM (ω_1 =0, ω_2 =1)

Next, we consider weighting values ω_1 and ω_2 as 0.2 and 0.8 respectively. Results are illustrated in Fig. 3. We can notice that one × and two ▽ are all changed as ○, and 6 × elements are also changed as ▽.

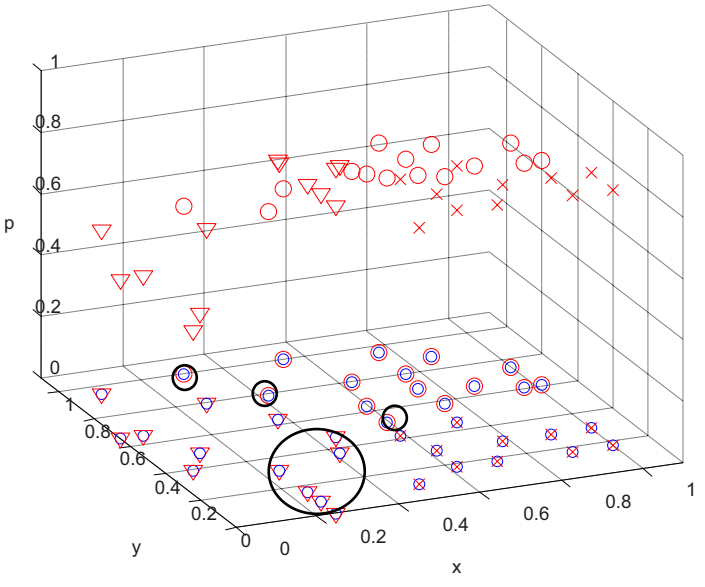


Fig. 3. Clustering by FCM (ω_1 =0.2, ω_2 =0.8)

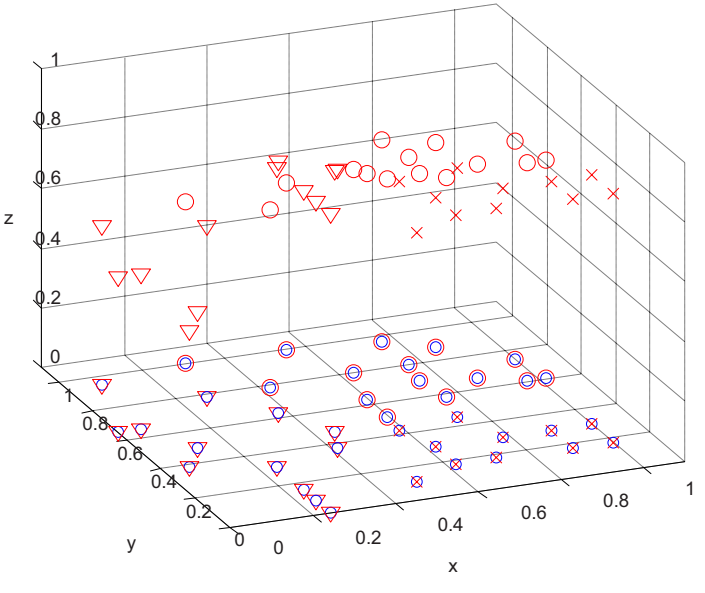


Fig. 4. Clustering by FCM (ω_1 =0.27, ω_2 =0.73)

With these results, we change the weighting values more. ω_1 and ω_2 are 0.2 and 0.8 respectively. In Fig. 4, there is just one circle compared with Fig. 3. However Fig. 4 has more changes than Fig. 3 if we consider initial condition of Fig. 2.

Finally, we consider ω_1 and ω_2 as 0.27 and 0.73 respectively. The result is shown in Fig. 4. In final result we cannot notice any special changing, however there are so many changes near the cluster boundaries at each iterations. As a result, we have to determine weighting values properly for the useful applications.

5 Conclusion

In this paper, we have introduced FCM and similarity. We have also constructed the similarity measure using the distance measure. For grouping of the interconnected networked system in terms of an appropriate similarity measure, regional information should be properly considered in the formulation of the similarity measure. In this paper, therefore, we have proposed a modified similarity measure accompanied with regional information, followed by example on the IEEE reliability test system to verify the usefulness of the proposed idea of the modified similarity measure. From the results, we can check the coherency between the degree of similarity level and the number of clusters.

Acknowledgement

This work has been supported by KESRI(R-2005-B-112), which is funded by MOCIE (Ministry of commerce, industry and energy).

References

1. W. Li and A. Bose, "A coherency based rescheduling method for dynamic security", IEEE Transactions on Power Systems, Vol. 13, No. 3, 810-815, 1998.
2. S.K. Joo, C.C. Liu, L.E. Jones, and J.W. choe, "Coherency and aggregation techniques incorporating rotor and voltage dynamics", IEEE Transactions on Power Systems, Vol. 19, No. 2, 1068-1075, 2004.
3. A.M. Gallai and R.J. Thomas, "Coherency Identification for large electric power systems", IEEE Transactions on Circuits and Systems, Vol. CAS-29, No. 11, 777-782, 1982.
4. F.F. Wu. N. Narasmithamurthi, "Coherency Identification for power system dynamic equivalents", IEEE Transactions on Circuits and Systems, Vol. CAS-30, No. 3, 140-147, 1983.
5. Liu Xuecheng, "Entropy, distance measure and similarity measure of fuzzy sets and their relations," Fuzzy Sets and Systems, 52, 305-318, 1992.
6. J. L. Fan, W. X. Xie, "Distance measure and induced fuzzy entropy," Fuzzy Set and Systems, 104, 305-314, 1999.
7. J. L. Fan, Y. L. Ma, and W. X. Xie, "On some properties of distance measures," Fuzzy Set and Systems, 117, 355-361, 2001.

8. S.H. Lee, S.P. Cheon, and Jinho Kim, "Measure of certainty with fuzzy entropy function", LNAI, Vol. 4114, 134-139, 2006.
9. S.H. Lee, J.M. Kim, and Y.K. Choi, "Similarity measure construction using fuzzy entropy and distance measure", LNAI Vol.4114, 952-958, 2006.
10. J.C. Bezdek, *Fuzzy Mathematics in Pattern Classification*, Ph.D Thesis, Applied Math. Center, Cornell University, Ithaca, 1973.
11. J.S.R. Jang, C.T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, 1997.
12. The IEEE Reliability Test System-1996, A report prepared by the Reliability Test System Task Force of the Application of Probability Methods Subcommittee, IEEE Transactions on Power Systems, Vol. 14, Issue 3, 1999

Centroid Neural Network with Bhattacharyya Kernel for GPDF Data Clustering

Song-Jae Lee and Dong-Chul Park

Dept. of Information Eng.and Myongji IT Eng. Research Inst.
Myong Ji University, Korea
{songjae, parkd}@mju.ac.kr

Abstract. A clustering algorithm for GPDF data called Centroid Neural Network with Bhattacharyya Kernel (BK-CNN) is proposed in this paper. The proposed BK-CNN is based on the unsupervised competitive centroid neural network (CNN) and employs a kernel method for data projection. In order to cluster the GPDF data, the Bhattacharyya kernel is used to measure the distance between two probability distributions for data projection. When applied to GPDF data in an image classification model, the experiment results show that the proposed BK-CNN algorithm is more efficient than other conventional algorithms such as k-means algorithm, SOM and CNN with Bhattacharyya distance.

Keywords: kernel, clustering.

1 Introduction

Conventional studies on data analysis, image classification, pattern recognition and speech recognition have used competitive algorithms based on k-means algorithm[1] and Self-Organizing Map (SOM)[2]. Gaussian Mixture Models (GMMs) and maximum or minimum-likelihood classifiers are used for modeling and classification, respectively. To cluster the GPDF data in GMMs, the conventional k-means algorithm and SOM clustering algorithms are widely used[1]. However, because of the selection of parameters such as learning rates and total number of iterations and the initialized conditions, the k-means algorithm and SOM algorithms often give unstable results. Park proposed a competitive clustering algorithm called the Centroid Neural Network (CNN)[3]. Compared with the conventional k-means algorithm and SOM, the CNN converges stably to suboptimal solutions[3].

In order to improve the recognition accuracy in GPDF data clustering problems, the maximum likelihood (ML) estimation is one of the empirical approaches. To utilize the full information contained in data as specified by the probability density function, an alternative method is the cross-entropy[4]. The Kullback-Leibler and the Bhattacharyya distance measures are the representative examples of the cross-entropy. The Centroid Neural Network with the Bhattacharyya distance (B-CNN) was first proposed by Park and Kwon[5].

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 616–622, 2007.

In this paper, we propose a new algorithm for clustering of GPDF data called Centroid Neural Network with Bhattacharyya Kernel (BK-CNN). The proposed BK-CNN is based on the CNN algorithm and employs a kernel method for data projection. Though the kernel method has been successfully applied in various fields such as Support Vector Machine[6] and Fuzzy Clustering [7], it was designed for clustering of deterministic data because of its Euclidean distance. In this paper, the Bhattacharyya kernel is used to measure the distance between two probability distributions for clustering probability data[8].

The remaining of this paper is organized as follows: In Section 2, we briefly review the conventional Centroid Neural Network (CNN) and the Bhattacharyya distance as a distance measure between two GPDFs distributions. Section 3 introduces the proposed BK-CNN algorithm. Section 4 shows experiments and results including performance comparison of the BK-CNN with some conventional algorithms. Finally, conclusions are provided in Section 5.

2 Centroid Neural Network and a Divergence Measure

2.1 Centroid Neural Network

The CNN algorithm has been shown excellent results as an unsupervised competitive algorithm based conventional k-means algorithm[3,5]. It finds the centroids of clusters at each presentation of data vector. Unlike conventional unsupervised algorithms such as k-means algorithm and Self-Organizing Map, the CNN algorithm updates its weights only when the status of the output neuron for the current data has changed.

The following equations show the weight update equations for winner neuron j and loser neuron i when an input vector $\mathbf{x}$ is presented to the network at epoch n.

$$\begin{aligned} \mathbf{w}_j(n+1) &= \mathbf{w}_j(n) + \frac{1}{N_j+1}[\mathbf{x}(n) - \mathbf{w}_j(n)] \\ \mathbf{w}_i(n+1) &= \mathbf{w}_i(n) - \frac{1}{N_i-1}[\mathbf{x}(n) - \mathbf{w}_i(n)] \end{aligned} \tag{1}$$

where $\mathbf{w}_j(n)$ and $\mathbf{w}_i(n)$ represent the weight of the winner neuron and the loser neuron, respectively while N_i and N_j describe the number of data vectors in cluster i and j, respectively.

More detailed description on CNN can be found in [3,5].

2.2 Clustering in GPDF Data with Divergence Measure

The conventional k-means algorithm and its variants have been most widely used in practice for clustering GPDF data. In order to exploit entire information including the mean and covariance information in the GPDF data for clustering, the divergence measure is employed as similarity distance between two probability distributions. The popular Bhattacharyya measure distance is adopted as

divergence measure in this paper. The Bhattacharyya distance is a separability measure between 2 Gaussian distributions and is defined as follows:

$$D(G_i, G_j) = \frac{1}{8}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \left[\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}\right]^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2}\ln \frac{\left|\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}\right|}{\sqrt{|\boldsymbol{\Sigma}_i|\,|\boldsymbol{\Sigma}_j|}} \quad (2)$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ denote the mean vector and covariance matrix of a Gaussian distribution G_i, respectively. T denotes the transpose matrix.

3 Kernel Method and CNN with the Bhattacharyya Kernel(BK-CNN)

3.1 Updating Cluster Prototypes

The energy function with a kernel can be written in feature space with the mapping function Φ:

$$E_i^{\Phi} = \sum_{j=1}^{N_i} \|\Phi(\mathbf{x}_i(j)) - \Phi(\mathbf{w}_i)\|^2 \quad (3)$$

$\mathbf{x}_i(j)$ denotes the data j in the cluster i.

Through the kernel substitution in Eq.(4), we obtain

$$\begin{aligned}\|\Phi(\mathbf{x}_i(j)) - \Phi(\mathbf{w}_i)\|^2 &= (\Phi(\mathbf{x}_i(j)) - \Phi(\mathbf{w}_i))(\Phi(\mathbf{x}_i(j)) - \Phi(\mathbf{w}_i))^T \\ &= K(\mathbf{x}_i(j), \mathbf{x}_i(j)) + K(\mathbf{w}_i, \mathbf{w}_i) - 2K(\mathbf{x}_i(j), \mathbf{w}_i)\end{aligned}$$

In the case of the Gaussian kernel function, we have $K(\mathbf{x}_i(j), \mathbf{x}_i(j)) = 1$ and $K(\mathbf{w}_i, \mathbf{w}_i) = 1$, and the objective function becomes:

$$E_i^{\Phi} = 2\sum_{j=1}^{N_i}(1 - K(\mathbf{x}_i(j), \mathbf{w}_i)) \quad (4)$$

In order to minimize the objective function with a kernel, we use the steepest gradient descent algorithm. The learning rule can be summarized as follows:

$$\Delta\mathbf{w}_i = \eta(\mathbf{x}_i(j) - \mathbf{w}_i) = \eta\frac{\partial E_i^{\Phi}}{\partial \mathbf{w}_i} \quad (5)$$

In the case of the Gaussian kernel function, the objective function in Eq.(5) can be rewritten as:

$$E_i^{\Phi} = 2\sum_{k=1}^{c}(1 - D(\mathbf{x}_i(j), \mathbf{w}_i)) \quad (6)$$

From Eq.(7), we obtain:

$$\begin{aligned}\Delta\mathbf{w}_i &= 2\eta(D(\mathbf{x}_i(j), \mathbf{w}_i))'(\mathbf{x}_i(j) - \mathbf{w}_i) \\ &= 4\eta(\boldsymbol{\Sigma}_{x_i(j)} + \boldsymbol{\Sigma}_{w_i})^{-1} D(\mathbf{x}_i(j), \mathbf{w}_i)(\mathbf{x}_i(j) - \mathbf{w}_i)\end{aligned} \quad (7)$$

In the BK-CNN, $(N_i + 1)^{-1}$ is used instead of 4η like the CNN,

$$\Delta \mathbf{w}_i = (N_i + 1)^{-1}(\boldsymbol{\Sigma}_{x_i(j)} + \boldsymbol{\Sigma}_{w_i})^{-1} D(\mathbf{x}_i(j), \mathbf{w}_i)(\mathbf{x}_i(j) - \mathbf{w}_i) \tag{8}$$

3.2 CNN with the Bhattacharyya Kernel(BK-CNN)

Recently, the kernel method has been used in various clustering algorithms [9,10,11]. The kernel method is based on mapping data from the input space to a feature space of a higher dimensionality, and then solving a linear problem in that feature space. It has been successfully employed in many traditional clustering algorithms such as Support Vector Machine[6], Fuzzy Kernel Perceptron [12]. In order to calculate the kernel between two GPDF data,the Bhattacharyya kernel is employed. The Bhattacharyya kernel is an extension of the standard Gaussian kernel. The Bhattacharyya kernel function between two GPDF data is defined as follows:

$$BK(\mathbf{x}(n), \mathbf{w}_j(n)) = \exp\left(-\alpha D(\mathbf{x}(n), \mathbf{w}_j(n)) + b\right) \tag{9}$$

where $BK(\mathbf{x}(n), \mathbf{w}_j(n))$ is the Bhattacharyya with a kernel distance between two Gaussian distributions $\mathbf{x}(n)$ and $\mathbf{w}_j(n)$. In this paper, the Bhattacharyya distance with a kernel as shown in Eq. (10) is employed for GPDF data.

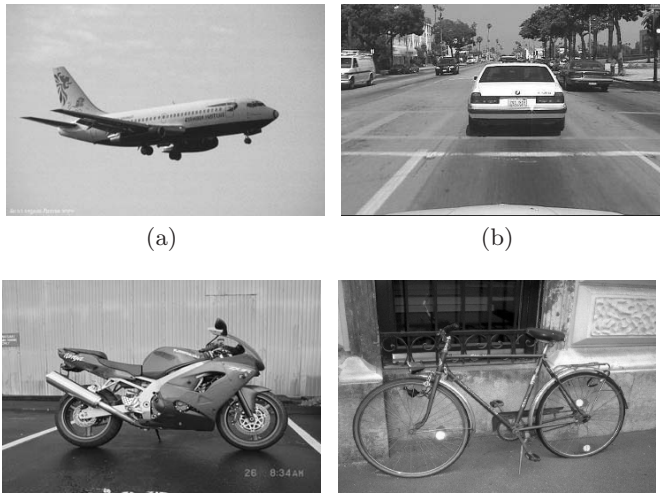
Unlike the CNN, we should consider the variance, Σ as well as the mean, μ. The rule of the weight update for the mean is similar to the CNN: that is, the μ of the winner weight go close to the input data vectors while the one of the looser weight goes away from the input vectors at every iteration. Therefore, the update rule is defined as follows:

$$\begin{aligned}
\mathbf{w}_j(n+1) &= \mathbf{w}_j(n) + \frac{\alpha BK(\mathbf{x}(n), \mathbf{w}_j(n))}{(N_j + 1) * (\boldsymbol{\Sigma}_{x(n)} + \boldsymbol{\Sigma}_{w_j(n)})}[\mathbf{x}(n) - \mathbf{w}_j(n)] \\
\mathbf{w}_i(n+1) &= \mathbf{w}_i(n) - \frac{\alpha BK(\mathbf{x}(n), \mathbf{w}_i(n))}{(N_i - 1) * (\boldsymbol{\Sigma}_{x(n)} + \boldsymbol{\Sigma}_{w_i(n)})}[\mathbf{x}(n) - \mathbf{w}_i(n)]
\end{aligned} \tag{10}$$

4 Experiments and Results

The performance of the proposed BK-CNN is evaluated and compared with other conventional clustering algorithms by applying to the Caltech image data set. The Caltech image data set consists of different image classes (categories) in which each class contains different views of an object. The Caltech image data were collected by the Computational Vision Group and can be downloaded at http://www.vision.caltech.edu/html-files/archive.html

From these classes, we selected the 4 most easily confused classes: airplane, car, bike, and motorbike for experiments. Each class consists of 200 images with different views resulting in a total of 800 images in the data set. From this data set, 100 images were randomly chosen for training while the remaining images were used for testing. The entire images are converted to grey scale and the same resolution. Fig. 1 shows an example of 4 image categories used in the experiments.



(a) (b)

(c) (d)

Fig. 1. (a) Airplane (b) Car (c) Motorbike (d) Bike

Figs. 1(a), 1(b), 1(c), and 1(d) are examples of car, airplane, motorbike, and bike, respectively. For the localized representation, the images are transformed into a collection of 8×8 blocks. The block is then shifted by an increment of 2 pixels horizontally and vertically. The DCT coefficients of each block are then computed and return in 64 dimensional coefficients. Only the 32 lowest frequency DCT coefficients that are visible to the human eye were kept. Therefore, the feature vectors that are obtained from each block have 32 dimensions. In order to calculate the GPDF for the image, the mean vector and the covariance matrix are estimated from all blocks obtained from the image. Finally, a GPDF with 32-dimensional mean vectors and 32×32 covariance matrixes is used to represent the content of images.

After mixtures are built, the minimum-likelihood classifier is adopted for choosing the class that the tested image belongs to:

$$Class(x) = \arg\min_{i} D(\boldsymbol{x}, C_i) \tag{11}$$

$$D(G(\boldsymbol{x}; \mu, \Sigma), C_i) = \sum_{k=1}^{N_i} w_{ik} D(G(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), G(\boldsymbol{x}; \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})) \tag{12}$$

where $\boldsymbol{x}$ is the tested image represented by a Gaussian distribution feature vector with mean vector, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$. $\boldsymbol{\mu}_{ik}$ and $\boldsymbol{\Sigma}_{ik}$ represent for the mean vector and covariance matrix of cluster k in class C_i, respectively. w_{ik} is the weight component of cluster k in class C_i. N_i is the number of clusters in the class C_i.

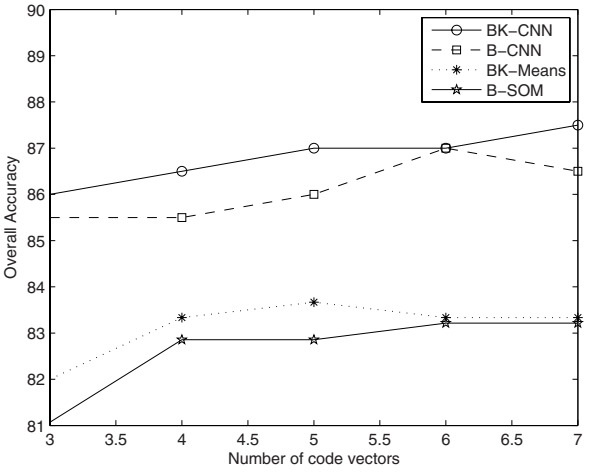


Fig. 2. Overall classification accuracies using different algorithms

Fig. 2 shows the classification accuracy of the classification model using SOM with a Bhattacharyya measure (B-SOM), the k-means algorithm with a Bhattacharyya measure(Bk-means), the CNN with a Bhattacharyya measure (B-CNN) and the proposed BK-CNN. In this figure, the number of code vector is varied from 3 code vectors to 7 code vectors in order to determine a sufficient number of code vectors to represent for the number of mixtures in GMMs. As can be seen from Fig. 2, the most algorithms tend to saturate at the point of 5 or 6 code vectors, while the proposed algorithm doesn't. The BK-CNN shows the better results than the other algorithms.

Table 1 shows the confusion matrix that describes the classification results of the proposed classification model in detail. As can be inferred from Table 1, cars can be well discriminated from the others while bikes and motorbikes are easily confused. These are logical results because motorbikes and bikes are quite similar even to the human eye while the cars are significantly different.

Table 1. Confusion matrix of image categories, using 5 code vectors

	Airplane	Car	Bike	Motorbike	Accuracy
Airplane	**84**	4.0	10.0	2.0	**84.0%**
Car	0.0	**100**	0.0	0.0	**100%**
Bike	12.0	0.0	**70.0**	18.0	**70.0%**
Motorbike	0.0	0.0	4.0	**96.0**	**96.0%**

5 Conclusions

A new clustering algorithm for clustering of GPDF data called Centroid Neural Network with a Bhattacharyya Kernel (BK-CNN) is proposed in this paper.

The proposed BK-CNN is formulated by a incorporation of the Kernel method, the Bhattacharyya distance and the competitive learning algorithm. The kernel method adopted in the proposed BK-CNN is used for transformation of data from input space into feature space of higher dimensionality to obtain nonlinear solutions. By using the Bhattacharyya divergence distance, the BK-CNN can be used for clustering the GPDF data to utilize entire the mean values and covariance information of the GPDF data. The proposed BK-CNN is applied to cluster the GPDF data in the images. These encroaching results imply that the proposed BK-CNN can be used as an efficient clustering tool for GPDF data in other practical applications.

References

1. Hartigan J.:Clustering Algorithms. New York, Wiley, (1975)
2. Kohonen T.:The Self-Organizing Map Processing of the IEEE, vol. 78 (1990) 1464-1480.
3. Park D.C.:Centroid Neural Network for Unsupervised Competitive Learning. IEEE Transaction on Neural Networks. vol. 11 (2000) 520-528.
4. Gokcay E., Principe, J.C.:Information theoretic clustering. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 24 (2002) 158-171.
5. Park D.C., Oh H. K., Min S.S.:Clustering of Gaussian Probability Density Functions Using Centroid Neural Networks. IEE Electronic Letters, vol. 49 (2003) 381-382.
6. Cristianini N., Shawe-Taylor J.:An Introduction to Support Vector Machine. Cambridge, Cambridge Univ. Press (2000)
7. Chen S., Zhang D.:Robust Image Segmentation using FCM with Spatial Constraints Based on New Kernel-Induced Distance Measure. IEEE Trans. on Systems Man and Cybernetics, vol. 43 (2004) 1907-1916.
8. Jebra T., Kondor.:Bhattacharyya and Expected Likelihood Kernels. Proc. COLT (2003)
9. Muller K.R., Mika S., Ratsch G., Tsuda K., Scholkopf B.:An Introduction to Kernel-Based Learning Algorithms. IEEE Transactions on Neural Networks, vol. 12 (2001) 181-201.
10. Cover T.M.:Geomeasureal and Statistical Properties of Systems of Linear Inequalities in Pattern Recognition. Electron, Computing, vol. EC-14 (1965) 326-334.
11. Girolami M.:Mercer Kernel-Based Clustering in Feature Space. IEEE Trans. on Neual Networks, vol. 13 (2002) 780-784.
12. Chen J.H., Chen C.S.:Fuzzy Kernel Perceptron. IEEE Trans. on Neural Networks, vol. 13 (2002) 1364-1373.

Concept Interconnection Based on Many-Valued Context Analysis

Yuxia Lei[1,2,3], Yan Wang[1], Baoxiang Cao[1], and Jiguo Yu[1]

[1] College of Computer Science and Technology, Qufu Normal University, Rizhao, Shandong, 276826, P.R. China
yx_lei@126.com
[2] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
[3] Graduate School of the Chinese Academy of Sciences, Beijing 100039, China

Abstract. This paper proposes an interconnection approach, which is based on *extended many-valued context* and *extended formal descriptions*. An *extended many-valued context* $\Pi=(G,M,Q,W,I)$ consists of sets G,M,Q and W and a quaternary-relation $I \subseteq G\times M\times Q\times W$. An *extended formal description* is regarded as a mapping from the set of attributes to the power set of the values, assigning to each attribute the set of allowed values under some conditions. The extended formal descriptions are naturally ordered by preciseness, and then a concept lattice is obtained according to the theory of FCA. This concept lattice is the well structure interconnection among concepts. The paper also proposed some important propositions, which are used to decide whether two concepts have semantic interconnections. In the end, the paper describes an interconnection algorithm with the time complexity $O(n^2)$.

Keywords: Formal Concept Analysis; Extended Many-Valued Context; Semantic Interconnection; Structure Interconnection; Formal Description.

1 Introduction

As an effective tool for data analysis and knowledge processing, formal concept analysis has been applied to various fields. Most of the researches on formal concept analysis focus on the following topics: construction of concept lattice[1],[2], revision of concept lattice[3], acquisition of rules[1],[2],[4], relationship with rough set [5], and concept interconnection[6],[7].

Knowledge processing in formal concept analysis usually starts with object-attribute-value relationships, which is a frequently used method to code real-world problems[8]. They can be represented in many-valued contexts as a quadruples (G, M, W, I). Unfortunately, there usually exist the following challenges in knowledge texts: 1) Some data most often have missing values, and 2) Some attributes may have different values under different conditions.

In order to discuss the concept interconnection with the problems, the paper proposes an approach based on the *extended many-valued context* and *extended formal descriptions*. A *many-valued context* forms an *extended many-valued context* by added

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 623–630, 2007.

a set of conditions. An *extended formal description* is regarded as a mapping from the set of attributes to the power set of the attribute-values under some conditions, assigning to each attribute the set of allowed values under some conditions.

The paper is organized as follows. Section 2 mainly discusses the structure interconnection based on the extended many-valued context and extended descriptions. Section 3 introduces the semantic interconnection based on description, and describes an algorithm for interconnecting two objects. Finally, section 4 concludes the paper and provides some interesting problems.

2 Structure Interconnection Based on Many-Valued Context

In the section, we firstly introduced the many-valued context based on descriptions, and then analyzed some important properties of the *extended many-valued context* and *extended formal descriptions*. Some important original definitions such as formal concept, many-value context and concept lattice are described in detail in article [9].

In order to handle empty cell in the theory of formal concept analysis mathematically, some researchers extended the power set of values, $P(W)$, by adding a special maximal element ∞, and extend the order on $P(W)$ given by inclusion to $P^*(W)$ by defining[8]:

$$P^*(W) := P(W) \cup \{\infty\}. \tag{1}$$

$$\forall A,B \in P(W)(A \subseteq B \rightarrow A \leq B);\ \forall A \in P(W)(A \leq \infty \wedge A \neq \infty). \tag{2}$$

Furthermore, each attribute $m \in M$ is regarded as a mapping from the $P(G)$ to $P^*(W)$:

$$m: P(G) \rightarrow P^*(W), m(A) := \{m(g) | g \in A \wedge (g,m) \in I\} \text{ or } \{\infty | (g,m) \notin I\}. \tag{3}$$

Definition 1[8]: Let K=(G, M, W, I) be a *many-valued context*. A *formal description* d in K is a *mapping* $d: M \rightarrow P^*(W)$ from the set M of attributes to the extended power set $P^*(W)$ of values. D denotes the set of all *descriptions* in K: $D := \{d: M \rightarrow P^*(W)\}$. This set is ordered by preciseness: $d_1 \leq d_2 :\Leftrightarrow d_1(m) \geq d_2(m), \forall m \in M$, An object $g \in G$ fulfills a description d, iff $m(g) \leq d(m)$, $\forall m \in M$.

Theorem 1[8]: The set D together with the order of preciseness forms a complete lattice. The supremum $\vee d_j$ of a family of descriptions dj; $j \in J$ is given by the conjunction of the descriptions, and the infimum $\wedge d_j$ is given by their disjunction:

$$\vee d_j := m | \rightarrow \cap d_j(m);\ \wedge d_j := m | \rightarrow \cup d_j(m) \tag{4}$$

Definition 2[8]: Let K=(G,M,W,I) be a *many-valued context*, and let D be the set of descriptions in K. A *formal concept* of the context K is a pair (B, d),with $B \subseteq G$, $d \in D$ and $B^\circ = d$ and $d^\circ = B$. B is called the *extent* and d the *intend* of the concept (B, d). The set of all concepts of the context K is denoted $\mathfrak{B}(K)$. Where $B^\circ := M \rightarrow P^*(W), m \rightarrow m(B)$, $d^\circ := \{g \in G | m(g) \leq d(m), \forall m \in M\}$.

Proposition 1: Let (A_1,d_1) and (A_2,d_2) be formal concepts in K, then $A_1 \subseteq A_2 \Rightarrow d_2 \leq d_1$.

The set of concepts is ordered by extent. In the case of $(A_1,d_1) \leq (A_2,d_2)$, we call (A_1,d_1) a *sub-concept* of (A_2,d_2), and (A_2,d_2) a *super-concept* of (A_1,d_1). (g, g°) is called object concept, obviously $g^\circ = d = \{m \to m(g) | m \in M \wedge m(g) \neq \infty\}$,in the case d can be reformulate $\{(m, m(g)) | m \in M \wedge m(g) \neq \infty\}$.Furthermore, $\mathfrak{B}(K)$ together with this order forms a complete lattice:

Theorem 2[8]: $\mathfrak{B}(K)$ forms a complete lattice under the following Infimum and supremums defined by (Where $\{(A_j,d_j) | j \in J\} \subseteq \mathfrak{B}(K)$.):

$$\wedge(A_j,d_j)=(\cap A_j,(\cap A_j)^\circ)=(\cap A_j,(\vee d_j)^{\circ\circ}); \vee(A_j,d_j)=((\wedge d_j)^\circ,\wedge d_j)=((\cup A_j)^{\circ\circ},\wedge d_j)). \quad (5)$$

The definition of descriptions so far would suffice, if each object would have in every attribute some values or missing values under different conditions. From the mathematical point of view, we assigned to each attribute the set of values under some conditions, which are allowed for the objects we want to describe. Therefore, we treat time and quantity as atomic knowledge building blocks. We have defined several time constructs for uniquely formalizing a time. For example, to formalize a concrete day, we use the construct <MM>.<DD>.<YYYY>. A quantity is usually characterized by a number and a unit of measurement. For example, "100km/h" and "1.8m" are quantities, where "km/h" and "m" are the units.

Definition 3: An *extended many-valued context* $\Pi=(G, M, Q, W, I)$ consists of sets G, M, Q and W and quaternary-relation I among G, M,Q and W (i.e. $I \subseteq G \times M \times Q \times W$) for which it holds that $(g, m, q, w) \in I$ and $(g, m, q, v) \in I$ always imply w=v.

The elements of G are called objects, those of M (many-valued) attributes, those of Q limited conditions and those of V attribute values. $(g, m, q, w) \in I$ means that "under the condition q, the attribute m has the value w for the object g." Obviously, under different conditions, the attribute m may have the different values the object g.

For example, the extended formal descriptions about Abraham Lincoln and John F. Kennedy as following: *Abraham Lincoln*{(Date of Birth:<02>.<12>.<1809>), (Height.Being an adult: About 1.95m), (Status. From 1847 to1859: Lawyer), (Status. from 1861 to 1865: President), (Date of Death:<04>.<14>.<1865>),. . .}, and *John F. Kennedy*{(Date of Birth: <05>.<29>.<1917>), (Height.Being an adult: About 1.67m), (Status. From 1947 to 1953: Representative), (Status. from 1961 to 1963: President), (Date of Death: <11>.<22>.<1963>),...}. In this case, we use g(m) instead of m(g) to distinct the descriptions of object g in K and ones in Π.

We regard each attribute $m \in M$ as a mapping from the P(G) to $P^*(Q \times W) := P^*(Q \times W) \cup \{(\infty,\infty)\}$:

$$\begin{gathered} m: P(G) \to P^*(Q \times W), \\ m(A) := \{(q,g(m)) | g \in A \wedge \exists q \in Q(g,m,q,g(m)) \in I\} \text{or } \{(\infty,\infty) | (g,m,q,g(m)) \notin I\} \end{gathered} \quad (6)$$

We extend the order on $P(Q \times W)$ given by inclusion to $P^*(Q \times W)$ by defining

$$\begin{gathered} \forall A,B \in P(Q \times W)((\alpha(A) \subseteq \alpha(B) \to A \leq B); \forall A \in P(Q \times W)(A \leq (\infty,\infty) \wedge A \neq (\infty,\infty)), \\ \text{Where } \alpha: Q \times W \to W,(q,v) | \to v. \end{gathered} \quad (7)$$

Definition 4: Let $\Pi=(G,M,Q,W,I)$ be a *extended many-valued context*. A *formal description* d in K is a *mapping* $d: M\rightarrow P^*(Q\times W)$ from the set M of attributes to the extended power set $P^*(Q\times W)$ of values under some conditions. The set of all *descriptions* in Π is denoted by $\Theta:=\{d:M\rightarrow P^*(Q\times W)\}$. This set is ordered by preciseness: $d_1 \le d_2 :\Leftrightarrow d_1(m) \ge d_2(m), \forall m\in M$, An object $g \in G$ fulfills a description d, iff $(q, m(g)) \le d(m)$, $\forall m\in M$.

Every finite many-valued context can be brought into an extended one without changing the structure of the concept lattice.

Proposition 2: $(\Theta,\le)$ is a complete lattice. The supremum $\vee dj$ of a family of descriptions dj; $j\in J$ is given by the conjunction of the descriptions, and the infimum $\wedge dj$ is given by their disjunction:

$$\vee d_j := m \mid \rightarrow \cap d_j(m);\ \wedge d_j := m \mid \rightarrow \cup d_j(m) \tag{8}$$

Definition 5: Let $\Pi=(G, M, Q, W, I)$ be a *extended many-valued context*, and let Θ be the set of descriptions in Π.A *formal concept* of the context Π is a pair (A, d),with $A\subseteq G$, $d\in\Theta$ and $A'=d$ and $d'= A$. A is called the *extent* and d the *intend* of the concept (A,d). $\mathfrak{B}(\Pi)$ denotes the set of all concepts of the context K. Where $A':=M\rightarrow P^*(Q\times W)$, $m\mid\rightarrow m(A)$, $d':=\{g\in G \mid (q, g(m))\le d(m), \forall m\in M\}$.

Proposition 3: Let (A_1,d_1) and (A_2,d_2) be formal concepts in Π, then $A_1\subseteq A_2 \Rightarrow d_2\le d_1$.

The set of concepts is ordered by extent (which is dual to the order given by intent). In the case of $(A_1,d_1)\le(A_2,d_2)$, we call (A_1,d_1) a *sub-concept* of (A_2,d_2), and (A_2,d_2) a *super-concept* of (A_1,d_1). Furthermore, $\mathfrak{B}(\Pi)$ together with this order forms a complete lattice:

Proposition 4: $\mathfrak{B}(\Pi)$ is a complete lattice under the following Infimum and supremum. Given a subset of concepts $\{(A_j, d_j)\mid j\in J\}\subseteq \mathfrak{B}(\Pi)$, we can define Infimum and supremum by:

$$\wedge(A_j,d_j)=(\cap A_j,(\cap A_j)')=(\cap A_j,(\vee d_j)'');\ \vee(A_j,d_j)=((\wedge d_j)',\wedge d_j)=((\cup A_j)'',\wedge d_j)). \tag{9}$$

3 Semantic Interconnection Based on Description

Given an extended many-valued context $\Pi=(G,M,Q,W,I)$, in order to discuss semantic interconnection from the formal description point of view, we define two two-argument functions as follows:

$$\psi: G\times M\rightarrow P(Q\times W),(g,m)\mid\rightarrow\{(q,v)\mid\forall v\in W\rightarrow\exists q\in Q(g(m)=v\wedge(g,m,q,v)\in I\} \tag{10}$$

$$\Gamma: G\times M\rightarrow P(W),(g,m)\mid\rightarrow\{v\in W\mid\forall v\in W\rightarrow\exists q\in Q(g(m)=v\wedge(g,m,q,v)\in I\} \tag{11}$$

The function ψ assigns to each object $g\in G$ and attribute $m\in M$ a subset $\psi(g, m)\subseteq Q\times W$, and Γ assigns to each object $g\in G$ and attribute $m\in M$ a subset $\Gamma(g, m)\subseteq W$.

Proposition 5: For each description d and each attribute m, we can get the following results:1)$\psi(g,m)=d(m)|_g=m(g)\backslash\{(\infty,\infty)\}$; 2) $\Gamma(g,m)=\alpha(\psi(g,m))=\alpha\ (m(g)\backslash\{((\infty,\infty))\})$; 3) For each concept (A, d), $d(m)|_A=m(A)=\cup_{g\in A}\{\psi(g,m)\cup\{(\infty,\infty)\}$.

Definition 6: Two object concepts $C_1=(g_1,d_1)$ and $C_2=(g_2,d_2)$, we say C_1 and C_2 are object semantic interconnect if satisfying the following conditions:

$$\cup_{m\in M}(\alpha(d_1(m)|g_1)\cap\alpha(d_2(m)|g_2))\backslash\{\infty\}\neq\Phi),\text{or} \tag{12}$$

$$\exists m_1\in M\exists m_2\in M\ (\alpha(d_1(m_1)|g_1)\cap\alpha(d_2(m_2)|g_2)\backslash\{\infty\}\neq\Phi),\text{or} \tag{13}$$

$$\exists m_1\in M\exists m_2\in M\ (\alpha(d_1(m_2)|g_1)\cap\alpha(d_2(m_1)|g_2)\backslash\{\infty\}\neq\Phi) \tag{14}$$

Proposition 6: The object semantic interconnection has the following basic forms:

$$g_1\approx^{++}g_2 \text{ iff } (\forall m\in M\ (\Gamma(g_1,m)\cap\Gamma(g_2,m))\neq\varnothing) \tag{15}$$

$$g_1\approx^{+}g_2 \text{ iff } (\exists m_1\in M\exists m_2\in M(\Gamma(g_1,m_1)\cap\Gamma(g_2,m_2))\neq\varnothing\vee(\Gamma(g_1,m_2)\cap\Gamma(g_2,m_1))\neq\varnothing) \tag{16}$$

$$g_1\approx^{-+}g_2 \text{ iff } (\exists m\in M\ (\Gamma(g_1,m)\cap\Gamma(g_2,m))\neq\varnothing) \tag{17}$$

Proposition 7: The following first-order conditions are true about semantic-interconnection among object concepts g_1, g_2 and g_3 of an extended many-value context: 1) $g_1\approx^{?}g_{1,}\ ?\in\{++,-+,+\}$;2) $g_1\approx^{++}g_2\leftrightarrow g_2\approx^{++}g_1$;3) $g_1\approx^{+-}g_2\leftrightarrow g_2\approx^{-+}g_1$;4) $g_1\approx^{?}g_2\rightarrow g_1\approx^{?}g_1,?\in\{++,-+,+\}$;5) $g_1\approx^{?}g_2\rightarrow g_2\approx^{?}g_2,?\in\{++,-+,+\}$

Definition 7: Given two concepts $C_1=(A_1,d_1)$ and $C_2=(A_2,d_2)$, we say C_1 and C_2 are semantic interconnect if satisfying the following conditions:

$$\cup_{m\in M}(\alpha(m(A_1))\cap\alpha(m(A_2)))\backslash\{\infty\}\neq\Phi),\text{or} \tag{18}$$

$$\exists m_1\in M\exists m_2\in M\ (\alpha(m_1(A_1))\cap\alpha(m_2(A_2)))\backslash\{\infty\}\neq\Phi),\text{or} \tag{19}$$

$$\exists m_1\in M\exists m_2\in M\ (\alpha(m_2(A_1))\cap\alpha(m_1(A_2)))\backslash\{\infty\}\neq\Phi) \tag{20}$$

Proposition 8: Given two concepts $C_1=(A_1,d_1)$ and $C_2=(A_2,d_2)$, the semantic interconnection has the following basic forms: 1) $C_1\approx^{++}C_2$ iff $\forall x\in A_1\forall y\in A_2\ (x\approx^{++}y)$;2) $C_1\approx^{+-}C_2$ iff $\exists x\in A_1\exists y\in A_2\ (x\approx^{++}y)$;3)$C_1\approx^{+}C_2$ iff $\forall x\in A_1\forall y\in A_2\ (x\approx^{+}y)$;4) $C_1\approx^{-}C_2$ iff $\exists x\in A_1\exists y\in A_2\ (x\approx^{+}y)$;5) $C_1\approx^{-+}C_2$ iff $\forall x\in A_1\forall y\in A_2\ (x\approx^{-+}y)$;6) $C_1\approx^{--}C_2$ iff $\exists x\in A_1\exists y\in A_2\ (x\approx^{-+}y)$

Proposition 9: The following first-order conditions are true about semantic-interconnection among concepts C_1, C_2 and C_3 in an extended many-value context: 1) $C_1\approx^{?}C_1,^{?}\in\{++,+-,+,-,\ -+,--\}$;2) $C_1\approx^{?}C_2\rightarrow C_2\approx^{?}C_1,^{?}\in\{++,+-,+,-,\ -+,--\}$;3) $C_1\approx^{?}C_2\wedge C_2\approx^{?}C_3\rightarrow C_1\approx^{?}C_3,^{?}\in\{++,+,\ -+\}$; 4) $C_1\approx^{?}C_2\wedge C_2\leq C_3\rightarrow C_1\approx^{?}C_3,^{?}\in\{++,+-,+,-,\ -+,--\}$;5)$C_1\leq C_3\wedge C_1\approx^{?}C_2\rightarrow C_1\approx^{?}C_3,^{?}\in\{++,+-,+,-,\ -+,--\}$.

Proposition 10: Given two concepts $C_1=(A_1,d_1)$ and $C_2=(A_2,d_2)$, then C_1 and C_2 are structure interconnect in the concept lattice if and only if they are semantic interconnect.

Definition 8: Let $\leq_m$ be an order relation on the set of values of m for every attribute m $\in$ M and let $C_1=(A_1,d_1)$ and $C_2=(A_2,d_2)$ be concepts. C_1 and C_2 are order interconnect if satisfying one of the following conditions:

$$\forall g_1\in A_1\forall g_2\in A_2\exists m\in M\exists g_1(m)\in \alpha(m(g_1))\exists g_2(m)\in \alpha(m(g_2))(g_1(m)\leq_m g_2(m)) \quad (21)$$

$$\forall g_1\in A_1\forall g_2\in A_2\exists m\in M\exists g_1(m)\in \alpha(m(g_1))\exists g_2(m)\in \alpha(m(g_2))(g_2(m)\leq_m g_1(m)) \quad (22)$$

$$\forall g_1\in A_1\forall g_2\in A_2\exists m_1\in M\exists m_2\in M\exists g_1(m_1)\in \alpha(m_1(g_1))\exists g_2(m_2)\in \alpha(m_2(g_2))\{(g_1(m_1)\leq_{m1\vee m2} g_2(m_2))\} \quad (23)$$

$$\forall g_1\in A_1\forall g_2\in A_2\exists m_1\in M\exists m_2\in M\exists g_1(m_1)\in \alpha(m_1(g_1))\exists g_2(m_2)\in \alpha(m_2(g_2))\{(g_1(m_1)\leq_{m1\vee m2} g_2(m_2))\} \quad (24)$$

Proposition 11: Order interconnection is reflexive and anti-symmetric.

Proposition 12: If $C_1\approx^{++} C_2$, or $C_1\approx^{+} C_2$, or $C_1\approx^{-+} C_2$, then C_1 and C_2 are order interconnect.

Definition 9: Two object g_1 and g_2 are $\cong$- interconnect if satisfying the following conditions:

$$\exists\, m_1\in M\exists\, m_2\in M\exists\, g_1(m_1)\in \alpha(m_1(g_1))\exists\, g_1(m_2)\in \alpha(m_2(g_1))\exists\, g_2(m_1)\in \alpha(m_1(g_2))\ \exists\, g_2(m_2)\in\ \alpha(m_2(g_2))\ \{|\, g_1(m_1)\text{-}g_2(m_1)|=|\, g_1(m_2)\text{-}g_2(m_2)|\} \quad (25)$$

Proposition 13: $\cong$- interconnection is reflexive, symmetric and anti-symmetric.

In order to analyze relationships between attributes and concept interconnection based on attributes, we introduce implications between descriptions, which based on the some definitions described in [8],[9]. Given an extended many-valued context Π=(G, M, Q, W, I), an implication $d_1\rightarrow d_2$ between two descriptions $d_1,d_2 \in D$ *holds* in Π, iff each object $g\in G$, which fulfills description d_1, also fulfills description d_2**.**More formally, we define:

Definition 10: *Let* Π=(G, M, Q, W, I) *be an extended many-valued context, and let D be the set of all descriptions within* Π. *Furthermore, let* $d_1,d_2 \in D$ *be descriptions. A extended description* $d\in D$ respects *the implication* $d_1\rightarrow d_2$, *if* $\neg(d_1\leq d)$ *or* $(d_2\leq d)$. *d respects a set* $\wp$ *of implications if d respects every single implication in* $\wp$. $d_1\rightarrow d_2$ holds *in a set* $\{d_3, d_4,\ldots\ldots, d_n\}$ *of descriptions if each of the descriptions* d_i $(3\leq i\leq n)$ *respects the implication* $d_1\rightarrow d_2$. $d_1\rightarrow d_2$ holds in the context Π *if it holds in the system of object intents. In this case, we also say, that* $d_1\rightarrow d_2$ *is* an implication of the context Π *or, equivalently, that within the context* Π, d_1 *is a* premise of d_2.

Proposition 14: Let $d_1 \rightarrow d_2$ be an implication of the context Π. If C_1 and C_2 are semantic interconnect (or order interconnection) under the description d_2, then they are too semantic interconnect (or order interconnection) under the description d_1.If g_1 and g_2 are $\cong$-interconnect under the description d_2, then they are too $\cong$-interconnect under the description d_1 .

These propositions are used to decide whether two concepts have interconnect tions. Now, we describe an object interconnection algorithm and analyze its time complexity.

An Object Interconnection Algorithm

Input: *Concept Lattice $\aleph(\Pi)$ and two object concepts (g_1,d_1) and (g_2,d_2)*

Output: *A set of interconnection $\Im$ between (g_1,d_1) and (g_2,d_2)*

Process:

1: $\Im \leftarrow \{\}$

2: $\Im'' \leftarrow (g_1,d_1) \wedge (g_2,d_2),\ (g_1,d_1) \vee (g_2,d_2)$

3: ***For*** *each attribute* $m \in M$ ***Do***

Case 1*:If* $\alpha(d_1(\mathrm{m})) \cap \alpha(d_2(\mathrm{m})) \setminus \{\infty\} \neq \Phi$ *, Then* $\Im \leftarrow = (g_1(m),\ g_2(m))$.

Case 2*:If* $\exists m_1 \in M \exists m_2 \in M\ ((\alpha(d_1(\mathrm{m}_1)) \cap \alpha(d_2(\mathrm{m}_2)))) \setminus \{\infty\} \neq \Phi$,

Then $\Im \leftarrow = (g_1(m_1),\ g_2(m_2))$.

Case 3*:If* g_1 *and* g_2 *are satisfying (21) or (22),*

Then $\Im \leftarrow \leq_m (g_1(m),\ g_2(m))$ *or* $\leq_m (g_2(m),\ g_1(m))$.

Case 4*: If* g_1 *and* g_2 *are satisfying (23) or (24),*

Then $\Im \leftarrow \leq_{(m1 \vee m2)} (g_1(m_1),\ g_2(m_2))$ *or* $\leq_{(m1 \vee m2)} (g_1(m_2),\ g_1(m_1))$.

Case 5*:If* g_1 *and* g_2 *are satisfying (25), Then* $\Im \leftarrow \cong (g_1,\ g_2)$.

Case 6*:If* g_1 *and* g_2 *have implication interconnection* $\copyright$, *Then* $\Im \leftarrow \copyright$

Endfor

4: $\Im \leftarrow \Im \cup \Im''$

Proposition15: Time complexity of the interconnection algorithm is $O(n^2)$.Where $|M|=n$.

Proof. We can only take into account the compare number from case 1 to case 5. In the best condition, the compare number from case 1 to case 5 is n, n^2, $2n$, $2n^2$ and C^2_n, respectively. Therefore, the total compare number is $f(n)= C^2_n+3n^2+3n \leq 7n^2 (n_0 \leq n)$.In the worst condition, the compare number from case 1 to case 5 is $|Q|^2 n$, $|Q|^2 n^2$,$2|Q|^2 n$,$2|Q|^2 n^2$ and $|Q|^2 C^2_n$, respectively. Therefore, the total compare number is $f(n)= |Q|^2 C^2_n+3|Q|^2 n^2+|Q|^2 3n \leq 7|Q|^2 n^2$. □

Now, we provide an example to explain these interconnection definitions. From the extended formal descriptions about Lincoln and Kennedy, we can easily obtain the following interesting knowledge: 1) Lincoln *was higher than* Kennedy; 2) Lincoln certainly *did not know when* Kennedy died and *what* the reason was; 3) *One hundred years* after Lincoln becoming a president, John F. Kennedy won president position.

4 Conclusion

In section 2, we mainly introduced the structure interconnection based on the extended many-valued context and extended description. From the concept interconnection point of view, concept Lattice is a well structure interconnection. Section 3 mainly dealt with semantic interconnection based on the extended description and implication between descriptions, and described an algorithm for interconnecting two objects. The paper proposed some important propositions, which are used to decide whether two concepts have interconnections.

Several problems remain to be investigated. One of the interesting questions is how to interconnect fuzzy concepts and distinguishable objects in the real world. In the future, we will focus on these questions and formalize the concept interconnection more deeply.

Acknowledgments. This work is supported by a grant from Foundation of Natural Science(No.Y2003G05), Promotional Foundation for Excellent Middle-aged or Young Scientists(No.2005BS01016), and Foundation of Teaching Reformation of Higher Education (No.B05042) of Shandong Province.

References

1. Carpineto, C., Romano, G. GALOIS.: An order-theoretic approach to conceptual clustering. In: Utgoff, P. (ed.), Proceedings of ICML 293: 33-40, (1993).
2. Godin, R.: Incremental concept formation algorithm based on Galois (concept) lattices. Computational Intelligence, 11(2):246-267, (1995).
3. Oosthuizen, G.D.: The Application of Concept Lattice to Machine Learning. Technical Report, University of Pretoria, South Africa, (1996).
4. Wen-Xiu Zhang, LingWei, and Jian-Jun Qi.: Attribute Reduction in Concept Lattice Based on Discernibility Matrix. RSFDGrC 2005, LNAI 3642:157–165, (2005).
5. Yiyu Yao.: A Comparative Study of Formal Concept Analysis and Rough Set Theory in Data Analysis. RSCTC 2004, LNAI 3066:59–68, (2004).
6. Yuxia LEI, Baoxiang Cao, Yan Wang, and Jiguo Yu.: Formal Analysis of Concept-Interconnection. Journal of Computational Information Systems. Vol 1(4):863-869, (2005).
7. Yuxia LEL, Shuxi Wang, and Baoxiang Cao.: Analysis and Design of Concept Interconnection Based on Concept Lattice.Computer Engineering and Application,Vol41 (33):42-44, (2005).
8. Ralf Gugisch.: Many-Valued Context Analysis Using Descriptions. H.Delugach and G.Stumme(Eds.):ICCS 2001, LNAI 2120:157-168, (2001).
9. Bernhard Ganter · Rudolf Wille.: Formal Concept Analysis: Mathematical Foundations. Springer,(1999).

Text Classification for Thai Medicinal Web Pages

Verayuth Lertnattee[1] and Thanaruk Theeramunkong[2]

[1] Faculty of Pharmacy, Silpakorn University
Nakorn Pathom, 73000, Thailand
verayuths@hotmail.com,verayuth@email.pharm.su.ac.th
[2] Sirindhorn International Institute of Technology
Thammasat University, Pathum Thani, 12000
thanaruk@siit.tu.ac.th

Abstract. Automatic text classification of web texts in Asian languages is a challenging task. For text classification of Thai web pages, it is necessary to cope with a problem called word segmentation since the language has no explicit word boundary delimiter. While a set of terms for any texts can be constructed with a suitable word segmentation algorithm, Thai medicinal texts usually has some special properties, such as plentiful of unique English terms, transliterates, compound terms and typo errors, due to their technical aspect. This paper presents an evaluation of classifying Thai medicinal web documents under three factors; classification algorithm, word segmentation algorithm and term modeling. The experimental results are analyzed and compared by means of standard statistical methods. As a conclusion, all factors significantly affect classification performance especially classification algorithm. The TFIDF with term distributions, as well as SVM, achieves high performance on non-segmented and segmented Thai medicinal web collection as they efficiently utilize technical terms.

1 Introduction

The fast growth and dynamic change of online information, have provided us a very large amount of information. Text Categorization (TC) is an important tool for organizing documents into classes by applying statistical methods or artificial intelligence (AI) techniques. With this organization, the utilization of the documents is expected to be more effective. A variety of learning techniques for TC, have been developed, such as Bayesian approaches [1], linear classifiers [2] and support vector machines [3]. Most research works have used standard data sets which is published in English, a language with word boundary delimiter such as Reuters, OHSUMED, 20-Newsgroups for evaluating the experiment. Recently, some research works [4,5] have applied TC techniques to medical documents (e.g., OHSUMED). However, there is no systematic research work on text categorization in other language, especially for Thai medicinal collection. Like other Asian languages such as Chinese, Korean and Japanese, Thai language also has no explicit word boundary delimiter. For languages without word

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 631–638, 2007.

boundary, word segmentation plays an important role to construct a set of terms for classification process. One more feature in most Thai medicinal web pages is that they usually include a plentiful of technical terms in both Thai and English. This property implies a large number of unique terms in the web collection. Most of technical terms in Thai medicinal web pages are compound terms which can be captured by the concept of n-gram (i.e., bigram or trigram). Moreover, such technical web pages also contain transliterates, and typo errors in both Thai and English, due to their technical aspect.

This paper presents an evaluation of several techniques in handling these problems. Taken into account are three different factors; classification algorithm, word segmentation algorithm and term modeling. In the rest of this paper, Section 2 presents a number of well-known classification algorithms. Some characteristics of Thai medicinal web pages and the ways to construct representative models for Thai medicinal web pages are given in Section 3. The data sets and experimental settings are described in Section 4. In Section 5, a number of experimental results are given. A conclusion is made in Section 6.

2 Classification Algorithms

This section gives a brief introduction to three well-known algorithms that were widely used for text classification i.e. naïve Bayes, centroid-based algorithms and the support vector machine (SVM).

2.1 Naïve Bayes Algorithm

As a statistical-based algorithm, the naïve Bayes classifier (NB) first calculates the posterior probability $P(c_k|d_j)$ of class c_k that the document belongs to different classes, and assigns it to the class with the highest posterior probability. Basically, a document d_j can be represented by a bag of words $\{w_{1j}, w_{2j}, \ldots, w_{nj}\}$ in that document (i.e., a vector of occurrence frequencies of words in the document). NB assumes that the effect of a word's occurrence on a given class is independent of other words' occurrence. With this assumption, a NB classifier finds the most probable class $c_k \in \mathcal{C}$, called a *maximum a posteriori* (MAP) c_{MAP} for the document which is determined by $\arg\max_{c_k} \prod_{i=1}^{n} P(w_{ij}|c_k)P(c_k)$. As a preliminary experiment, occurrence frequency for calculating the posterior probability $P(w_{ij}|c_k)$ outperforms the binary frequency. Therefore, this method will be used in this work.

2.2 Centroid-Based Algorithm

The centroid-based algorithm is a linear classification algorithm. Only positive documents are taken into account for constructing a centroid vector of a class. The vector is normalized with the document length to a unit-length vector (or prototype vector). In the classification stage, a test document is compared with these prototype vectors by dot product (cosine measure) in order to find the

nearest class. Normally, a centroid-based classifier (CB) obtained high classification accuracy with small time complexity. The techniques to improve the CB by introducing term-distribution factors to term weighting, in additional to the standard $tf \times idf$ was proposed [6]. In this work, we also use term distributions (TD) with CB. The term weighting formula is $\frac{\overline{tf}_{ik} \times idf_i}{\sqrt{csd_{ik}} \times sd_i}$. From this formulas, $\overline{tf}_{ik}$ and csd_{ik} are average class term frequency and class standard deviation of a term t_i in class c_k, respectively. The idf_i and sd_i are inverse document frequency and standard deviation of term t_i, respectively. Normally, combination of TD and TFIDF in an appropriate way, outperforms TFIDF.

2.3 Support Vector Machines

Support vector machines (SVMs) are based on the structure risk minimization principle [7]. It has been shown in previous works [3] to be effective for text categorization. SVM divides the term space into hyperplanes or surface separating the positive and negative training samples. An advantage of SVM is that it can work well on very large feature spaces, both in terms of the correctness of the categorization results and the efficiency of training and categorization algorithm. However, a disadvantage of SVM training algorithm is that it is a time consuming process, especially training with a large corpus.

3 Thai Medicinal Web Collection

The WWW permitted the general public to navigate easily across the global Internet and view a variety of information. In Thailand, most of web documents are conducted in Thai. However, in practical situation, English language is frequently contributed to Thai web pages, especially educational web pages. There are a large number of medical information related to disease, drug, herbal medicine and so on. Most of them are written in Thai with English terms attached. So far, several research works have developed methods for classifying medical document written in English. Up to present, there is no systematic work for classifying Thai medicinal web collection. From our preliminary study, we found that this type of documents has some special properties. These documents usually contain a lot of technical terms in both Thai and English. Most technical terms are compound noun and they are easily mistyped. From these properties, the system for classifying the collection are investigated in a systematic way. In order to construct a set of terms (features) in the step of learning and classifying phases, two factors are considered. Firstly, word segmentation is an essential component in creating features (words) from a sentence. Lastly, higher-grams seem to be a good representative for constructing features for medicinal documents. The detail of these factors are described.

3.1 Thai Word Segmentation Algorithms

In Thai language, a word segmentation plays as an important role to construct a set of terms for classification process. Most researchers had implemented their

Thai word segmentation system based on using a dictionary. Currently, three algorithms: *longest matching, maximal matching* and n-gram are well-known and widely used. Most of early works in Thai word segmentation are based on a longest matching algorithm [8]. The algorithm scans an input sentence from left to right, and selects the longest match with a dictionary entry at each point. In case that the selected match cannot lead the algorithm to find the rest of the wards in the sentence, the algorithm will backtrack to find the next longest one and continue finding the rest and so on. It is obvious that this algorithm will fail to find the correct the segmentation in many cases because of its greedy characteristics. The maximal matching algorithm was proposed to solve the problem of the longest matching algorithm [9]. This algorithm first generates all possible segmentations for a sentence and then select the one that contain the fewest words, which can be done efficiently by using dynamic programming technique. Because the algorithm actually finds real maximal matching instead of using local greedy heuristics to guess, it always outperforms the longest matching method. Besides the two algorithms, another popular statistical model is n-gram. The n-gram assumes that the probability of the event depends on n previous events. For a word segmentation, n-gram is applied as a measure of whether a word boundary is likely to locate between a current character and its preceding characters or not. Probability of this likelihood is represented by $p(c_{i+1}|c_1, c_2, \ldots, c_{i-1}, c_i)$ where c is a character in a word. It is well known that to obtain a good estimation for this statistic, a large corpus is required. While up to three contiguous characters (3-gram) is taken into account, this work apply bigram (n=2) due to computation issue.

3.2 Representation Basis

The frequently used document representation in IR and TC is the so-called bag of words (BOW) [10] where words in a document (or a class) are used as basics for representing that document. In this representation, each element (or feature) in the vector is equivalent to a unique word with a weight. In a more general framework, the concept of n-gram can be applied. Instead of a single isolated word, a sequence of n words will be used as representation basis. In several applications, not specific for classification, the most popular n-grams are 1-gram (unigram), 2-gram (bigram) and 3-gram (trigram). Alternatively, the combination of different n-grams, for instance the combination of unigram and bigram, can also be applied. Although a higher-gram provides more information and this may effect in improving classification accuracy, more training data and computational power are required. Therefore, unigram and bigram are considered in this work.

4 Data Sets and Preprocessing

The dataset used for evaluation is a set of web documents which are collected from several Thai medicinal websites. Composed of eight categories (i.e. Education, Disease, Drug, Food, Herbal, Toxic, Organization and Dental), the numbers

of documents for these classes are 1298, 1615, 716, 426, 761, 323, 1492 and 258, respectively. This collection was constructed under the research in a project named "Research and Development of Resources for Processing Very Large-Scaled Information on the Internet –Information Retrieval and Data Mining–". The total number of pages is 6889. In this collection, several web pages contain English technical terms. Without any term selection, the number of terms in the unigram model after passing any Thai word segmentation algorithm is approximately seventy thousand. In details, there are a lot of errors triggered by mis-typing, mis-segmenting and so on. To partially solve this problem, we select a set of terms which appear in a collection at least three times. Focusing on unique words, the number of English terms is nearly the same as Thai terms. The total numbers of terms using longest matching, maximal matching and bigram algorithms in a unigram model are 27945, 28120 and 28587. They are 221376, 220960 and 224643 in bigram models, respectively.

Three models of features are investigated. The features of the first model (non-segmented model) are constructed from English unigram terms, Thai unigram terms and Thai phrases. Due to the fact that Thai word segmentation is not applied, the number of features in Thai for non-segmented model is almost as twice as that of unigram model. In the second and third models, three Thai word segmentation algorithms are used for the preprocessing step before starting the training process. The second model, all features are unigram term in both Thai and English. For the third model, we add the bigram terms into the second model and it is so-called bigram model. As a preprocessing, some stopwords (e.g., a, an, the) and all tags (e.g., <B>, </HTML>) were omitted from documents to eliminate the affect of these common words and typographic words. This may be helpful to make classification processes not depend on any specific format.

Three experiments are performed. In the first experiment, four types of classifiers are evaluated i.e. naïve Bayes classifier (NB), centroid-based classifier with $tf \times idf$ term weighting (TFIDF), centroid-based classifier with $tf \times idf$ and term distributions (TFIDF*TD, see detail in 2.2) and support vector machine (SVM). The query weighting is $tf \times idf$ for centroid-based classifiers. For SVM, the linear function is applied and the term weighting is $tf \times idf$. Three Thai word segmentations are considered i.e. longest matching, maximal matching and bigram. The second and last experiments investigate the effect of our focused four classifiers and three Thai word segmentation algorithms in unigram and bigram models, respectively . All experiments were performed using 90% for the training set and 10% randomly for the test set. We performed 10 trials for each experiment. The performance was measured by classification accuracy defined as the ratio between the number of documents assigned with correct classes and the total number of test documents. Due to the fact that one factor and two factors are involved for each experiment, one-way and two-way analysis of variance (ANOVA) are used as a statistical method for evaluating the difference of mean with the significant level of 0.05. The difference of average classification accuracy between methods for each factor is compared by Scheffé's test, a method which is suitable for both multiple comparison and range test.

5 Experimental Results

5.1 Effect of Classifiers

In the first experiment, performance of four classifiers, i.e. NB, TFIDF, TFIDF*TD and SVM is explored. Table 1 showed the result in forms of the average classification accuracy±standard error of the mean (SEM).

Table 1. Effect of classifiers in non-segmented Thai medicinal web collection

Classifier	Accuracy
NB	77.24±0.39
TFIDF	63.23±0.49
TFIDF*TD	84.78±0.42
SVM	82.43±0.48

According to 1-way ANOVA between classifiers, the average accuracies of the four classifiers are significantly different ($p < 0.05$). NB is an intermediated performance classifier. Standard TFIDF classifier achieves the lowest performance among the four classifiers. However, term distributions improve its performance and outperforms SVM classifier. As describe in Section 4, the number of unique terms in non-segmented collection is greater than segmented unigram model collection. Term distributions efficiently utilizes these terms (or phrases).

5.2 Effect of Thai Word Segmentation Algorithms

In the second experiment, the effects of three Thai word segmentation algorithms i.e. longest matching (Longest), maximal matching (Maximal) and bigram (Bigram), and four classifiers are explored. The result is shown in Table 2.

Table 2. Effect of word segmentation algorithms and classifiers (unigram models)

Classifier	Thai Word Segmentation Algorithm		
	Longest	Maximal	Bigram
NB	80.68±0.59	80.64±0.41	80.26±0.39
TFIDF	76.74±0.43	76.60±0.35	76.66±0.61
TFIDF*TD	82.06±0.30	82.79±0.44	83.66±0.62
SVM	86.35±0.35	86.79±0.39	86.58±0.35

According to 2-way ANOVA between classifiers and Thai word segmentation algorithms, there is no significantly difference between the average accuracies of the three Thai word segment algorithms. This indicates that the types of word segmentation has no effect on performance and then we can apply any of them for classifying Thai medicinal Web pages. On the contrary, the average

accuracies of the four classifiers are significantly different ($p < 0.05$). Thai word segmentations segment Thai sentences into separate words. In this case, all features are represented with unigram model. The number of unique features in unigram model is less than non-segmented collection. However, the number of features in a document vector increases (longer document vector). The result is that all classifiers except TFIDF*TD, achieves higher performance.

5.3 Effect of Bigram Models

In the last experiment, bigram models of features are used instead of unigram models in previous experiment. The result is shown in Table 3.

Table 3. Effect of word segmentation algorithms and classifiers (bigram models)

Classifier	Thai Word Segmentation Algorithm		
	Longest	Maximal	Bigram
NB	82.73±0.49	83.11±0.35	83.28±0.39
TFIDF	74.10±0.31	75.65±0.48	74.04±0.40
TFIDF*TD	85.39±0.40	85.90±0.34	85.78±0.27
SVM	87.47±0.40	87.31±0.46	87.31±0.31

According to 2-way ANOVA between classifiers and models, the average accuracies of the four classifiers are significantly different ($p < 0.05$). There is also significantly difference ($p < 0.05$) between the average accuracies on unigram and bigram models. The bigram models usually achieve higher performance than unigram models for all classifiers except only TFIDF. The SVM on bigram model is still the best classifier in term of accuracy. However, time for learning phase increases a lot while a little bit improvement over unigram model. The TFIDF*TD is the second classifier in terms of performance. However, its performance in bigram models is comparable to SVM.

6 Conclusion

This paper investigated a set of methods to classify Thai medicinal Web documents in a systematic way and analyzed the results by statistical methods. Three factors are taken into account i.e classification algorithm, word segmentation algorithm and term modeling. Normally, This collection has some special properties i.e. a large number of terms in both Thai and English, several terms are represented in higher-gram and a lot of typing errors. From the experimental results, classification performance depends on the three factors especially classification algorithm. In a model without Thai word segmentation, the number of unique features is greater than those of a unigram model with Thai word segmentation. For this case, TFIDF with term distributions efficiently utilize these unique features. It outperformed other classifiers including SVM. When

Thai word segmentations were applied in both unigram and bigram models, SVM was superior to the others. However, the performance of TFIDF with term distributions in the bigram model was higher than the unigram and only less than the gap of 2% from the state-of-art algorithm, SVM. The results suggested that the bigram model of TFIDF with term distributions was a good model but we need to accept trade-off between time spent in both training phase and the accuracy for classifying Thai medicinal Web collection. It was the classifier of choice when Thai word segmentation was not applied or in the adaptive learning environment. The SVM is the classifier of choice when accuracy is the most important and any of Thai word segmentation algorithms can be applied.

In the future, we will consider the Thai medicinal Web documents with less English terminology and the other types of Thai Web documents. We will also consider other feature selection techniques.

Acknowledgment

This work was funded by the National Electronics and Computer Technology Center (NECTEC) via research grant NT-B-22-I4-38-49-05.

References

1. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: Proceedings of AAAI-98, Workshop on Learning for Text Categorization. (1998)
2. Han, E.H., Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: Principles of Data Mining and Knowledge Discovery. (2000) 424–431
3. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In Nédellec, C., Rouveirol, C., eds.: Proceedings of ECML-98, 10th European Conference on Machine Learning. Number 1398, Chemnitz, DE, Springer Verlag, Heidelberg, DE (1998) 137–142 Published in the "Lecture Notes in Computer Science" series, number 1398.
4. Chen, H., Ho, T.K.: Evaluation of decision forests on text categorization. In Lopresti, D.P., Zhou, J., eds.: Proceedings of the 7th SPIE Conference on Document Recognition and Retrieval, San Jose, US, SPIE - The International Society for Optical Engineering (2000) 191–199
5. Ruiz, M., Srinivasan, P.: Hierarchical text classification using neural networks. Information Retrieval **5** (2002) 87–118
6. Lertnattee, V., Theeramunkong, T.: Effect of term distributions on centroid-based text categorization. Information Sciences **158** (2004) 89–115
7. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag, New York (1995)
8. Poowarawan, Y.: Dictionary-based thai syllable separation. In: Proc. 9th Electronics Engineering Conference, Bangkok, TH (1986) 409–418
9. Sornlertlamvanich, V.: Word segmentation for thai in machine translation system. (1993) 50–56
10. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management **24** (1988) 513–523

A Fast Algorithm for Finding Correlation Clusters in Noise Data*

Jiuyong Li[1], Xiaodi Huang[2], Clinton Selke[3], and Jianming Yong[4]

[1] School of Computer and Information Science, University of South Australia, Mawson Lakes Adelaide, Australia, 5095
jiuyong.li@unisa.edu.au
[2] Department of Mathematics, Statistics and Computer Science, The University of New England, Armidale, Australia, 2350
xhuang@turing.une.edu.au
[3] Department of Mathematics and Computing, The University of Southern Queensland, Australia
clinux_rulz@hotmail.com
[4] Department of Information Systems, University of Southern Queensland, Australia
Jianming.Yong@usq.edu.au

Abstract. Noise significantly affects cluster quality. Conventional clustering methods hardly detect clusters in a data set containing a large amount of noise. Projected clustering sheds light on identifying correlation clusters in such a data set. In order to exclude noise points which are usually scattered in a subspace, data points are projected to form dense areas in the subspace that are regarded as correlation clusters. However, we found that the existing methods for the projected clustering did not work very well with noise data, since they employ randomly generated seeds (micro clusters) to trade-off the clustering quality. In this paper, we propose a divisive method for the projected clustering that does not rely on random seeds. The proposed algorithm is capable of producing higher quality correlation clusters from noise data in a more efficient way than an agglomeration projected algorithm. We experimentally show that our algorithm captures correlation clusters in noise data better than a well-known projected clustering method.

Keywords: generalised projected clustering, SVD decomposition.

1 Introduction

Clustering is a classical technique in computing and statistics. Noise deteriorates cluster quality significantly and prevents finding meaningful clusters when the amount of noise is big. It is difficult to distinguish noise data objects from normal ones when we do not have prior knowledge about the data. However, clustering can serve as the first step to explore such a data set with noise, particularly when the prior knowledge about the data is unavailable.

* This work was done when J Li was with Department of Mathematics and Computing of University of Southern Queensland. This work was partially supported by Australian Research Council Discovery Grant DP0559090.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 639–647, 2007.

Generalised projected clustering sheds light on solving this problem by finding correlated clusters. When data objects are projected to a data subspace using Singular Value Decomposition (SVD) or PCA, the correlation clusters are condensed to a small area whereas noise data objects scatter across the projected space. Therefore, it is possible to separate correlated data objects from noise ones.

Most existing projected clustering methods use agglomeration methods to find correlation clusters. A big data set is randomly partitioned into a large number of micro clusters, and then an agglomeration approach is used to group correlation clusters. When correlated data are split into a number of micro clusters, they themselves become noise too. This process of randomly generated seeds affects the quality of found clusters. When a process for noise elimination is employed, many data objects in the correlation clusters are removed before they are grouped into clusters. Some well known examples of generalized projected clustering are PROCLUS [2], ORCLUS [1], $4C$ [4], CURLER algorithm [5] and HARP [7]. We do not consider axis-parallel projection methods, also called subspace clustering, such as CLIQUE [3], and EPCH [6].

Instead of agglomerating randomly generated micro clusters into final clusters, we partition a data set into clusters in a top-down manner. The key idea for such a divisive method is to find a suitable criterion for data partition. We capture the direction of the largest variance of data using the corresponding principle vector, thereby taking small risk of partition correlation clusters into separate clusters. We employ grouping technique used in agglomeration methods to group correlation clusters after partitions. The proposed divisive projected clustering method preserves the essence of projected clustering, overcoming the drawbacks of existing projected clustering methods. In addition, the proposed algorithm is significantly more efficient than most agglomeration algorithms.

2 Problem Definitions

Projected clustering searches for hidden subspaces together with a set of data objects such that data objects are closed with each other in the lower dimensional subspaces. The hidden data spaces are found by using SVD decomposition. Eigenvectors corresponding to eigenvalues with low spreads forms a subspace. The intuitive explanation for this is as follows (see more justifications in Section 4). When the covariance matrix of a set of correlated points is decomposed by SVD, some eigenvalues should be zero or close to zero. All the points are projected along a line in the subspace spanned by eigenvectors corresponding to these zero eigenvalues. In other words, the tightness of objects in the subspace defined by eigenvectors associated with the lowest eigenvalues is an alternative to measuring the correlation level of data objects.

Formally, let D be a dataset of m data objects (row vectors) being treated as d-dimensional feature (column) vectors. $\mathbf{o}_i \in D$ stands for the i-th object in D where $\mathbf{o}_i = (o_{i1}, o_{i2}, \ldots, o_{id})$. Simply, we have $D = [o_{ij}]$, $1 \leq i \leq m$, and $1 \leq j \leq d$.

Definition 1. *Generalised projected clustering*
Given the user-specified l and k, a data set D is partitioned into k disjoint subsets D_1, D_2, ..., D_k horizontally, such that, for all $1 \leq p \leq k$, S_p contains l close to zero eigenvalues, where $U_p S_p V_p^T = \mathrm{cov}(D_p)$ (cov is the covariance matrix of D_p, the SVD decomposition of which results in U_p, S_p, and V_p^T).

Data points are clustered based on their closeness in some projected subspaces instead of the original space. This clustering captures correlations among data points.

To measure the closeness of data points in a subspace, the projected distance is defined as following.

Definition 2. *Projected distance*
Let D_p be a subset of a data D. $U_p S_p V_p^T = \mathrm{cov}(D_p)$ ($\mathrm{cov}(D_p)$ is the $d \times d$ covariance matrix of D_p, and U_p, S_p, and V_p^T are results of SVD decomposition). Let E be the set of eigenvectors corresponding to l smallest eigenvalues. A data object $\mathbf{p} \in D$ is projected to $E = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_l)$ space as $(\mathbf{p} \cdot \mathbf{e}_1, \mathbf{p} \cdot \mathbf{e}_2, \ldots, \mathbf{p} \cdot \mathbf{e}_l)$. The projected distance of objects $\mathbf{p}$ and $\mathbf{q}$, denoted by $\mathrm{Pdist}(\mathbf{p}, \mathbf{q}, E)$,is their Euclidean distance in projected space E.

The projected distance between two data points is the Euclidean distance between their projected images in a subspace. This distance varies in different subspaces.

To measure the projected distance variation over a group of data points in a subspace, the projected energy is defined as the following.

Definition 3. *Projected energy*
The projected energy of data set D_p is defined as $\mathrm{Energy}(D_p, E) = \sum_{i=1}^{i=N} \mathrm{Pdist}(\mathbf{o_i}, \mathbf{c}, E)/N$, where N is the number of objects in D_p, $\mathbf{c}$ the centroid of all objects in D_p, and E an associated subspace.

The smaller the projected energy, the denser the data point in the subspace. In clustering, low projected energy is preferred.

In projected clustering, the traditional distances between data objects are replaced by the projected distances in subspaces. However, there are no uniform and invariant distances in projected clustering since each tentative cluster has its own subspace. In other words, the distance between two objects varies in different subspaces.

Projected distances have been studied in statistics. The Mahalanobis distance [8] measures distance between two objects by using a set of reference data. But the Mahalanobis distance is the projected distance in the entire space. The generalised projected distance is defined in a subspace, and is a generalised Mahalanobis distance.

As mentioned before, the ORCLUS algorithm [1] presents a variant agglomerative method to find k projected clusters. First, D are randomly partitioned into k_0 initial data subsets D_1, D_2, ..., D_{k_0}, where $k_0 >> k$. If each data object is considered as an initial micro cluster, then the computational cost will be too expensive. The smaller k_0, the faster the ORCLUS. However, the high quality of clusters is sacrificed if k_0 is small.

Second, ORCLUS performs the following two iterations:

1. Merge pairs of clusters with the smallest, combined project energy until the number of clusters is down to k_p (determined by a parameter for the step size α).
2. Redistribute all data objects to the k_p clusters according to their respective, projected distance to each cluster center. An object is assigned to the cluster with the smallest, projected distance.

The above procedure terminates until $k_p = k$ with a parameter α to control the step size. If the step parameter is big, then a lot of merge occur in one iteration and the quality of final clusters is not guaranteed. If the step parameter is small, the execution time is increased.

A significant computational cost of the algorithm is from the decomposition of a data subset D_i. The complexity of such a decomposition is determined by the the number of dimension (attributes). Specifically, it costs $O(d^3)$. Moreover, the computation has to be done in each merger of two data subsets. The complexity of the ORCLUS algorithm is therefore as high as $k_0^3 + k_0 N d + k_0^2 d_0^2$.

A heuristic way of speeding up the ORCLUS algorithm is to make k_0 small and to conduct more merges in each step. However, the quality of clustering has been traded off. This problem is caused by the fact that ORCLUS is an agglomerative algorithm and too many merges are required to form a small number of clusters. In contrast, the divisive method needs much less steps to form clusters.

3 Divisive Projected Clustering (DPCLUS)

Large computational costs of projected clustering lie in computing covariance matrixes and SVD (or PCA) decomposition. The computational costs of covariance matrixes and SVD (or PCA) decomposition is largely determined by the number of attributes, d, rather than the number of objects in a data set N_i.

In most applications, we have $k << m$ where m is the number of objects in the data set, and k is the number of clusters. Therefore, a top-down method (divisive method) needs significantly less number of computations of covariance matrixes and SVD (or PCA) decomposition than a bottom-up one (agglomerative method).

A key question is how to partition the data. Given a dataset, the projected cluster problem can be regarded as the one of partition of the dataset into k clusters such that the sum of the projected distance of each data object to its cluster centroid is minimized. Compared to the clustering in full-dimensional space, the projected clustering makes use of the projected distance instead of the full-dimensional distance. Recall that we need to find a best subspace and their associated subsets of data objects such that the sum of the projected distances of data objects to their centroids is minimized. The number of the projected clusters in our algorithm is given. So it is to determine only the directions of spanning vectors. Within one cluster the optimal direction of the vector to which its associated data objects are projected should reflect the minimal variance of these data. An eigenvalue is numerically related to the variance it captures. The higher the value, the more variance it has captured. The principal vector defines a projection that encapsulates the maximum amount of variation in a dataset. This principal vector is in fact the eigenvector with the highest corresponding eigenvalue.

We make use of the principle vector of eigenvectors. All data objects D are projected to the principle vector as discussed in the previous section, and the centroid separates data into two groups: D_1 and D_2. D_1 contains data objects whose projected values are greater than or equal to the means, and D_2 contains the rest.

The pseudo code of the algorithm is listed below.

DPCLUS algorithm (Divisive Projected clustering)
Input: data set D, cluster number k, subspace dimension l, the minimum object number $\min_N$, and the minimum distance for excluding outliers δ
Output: $\geq k$ projected clusters

initialise an empty tree T;
let the root of T store D;
where (the number of leaves of $T < k$)
  foreach D_i stored in a leaf of the newest layer
    Partition (D_i);
  Redistribute (all data sets stored in the leaves of the newest layer);
output data sets stored in all leaves of T;

Function Partition(D_i)
if D_i satisfies Definition 1 or $|D_i| < \min_N$
  then terminate the leaf storing D_i and *return*;
split D_i into D_{i1} and D_{i2} by the centroid in the principal vector;
insert two son leaves of the node storing D_i to store D_{i1} and D_{i2};

Function Redistribution(all data sets stored in the newest layer)
foreach data object **p** in all data sets stored in the newest layer
  foreach data set D_j stored in the newest layer
    compute the projected distance between **p** and the center of $\mathbf{D}_j$;
    if projected distances of **p** to all data sets $> \delta$
      then exclude **p** from future clustering;
      else assign **p** to the data set with the smallest projected distance;

The DPCLUS algorithm partitions a data set into clusters in a top- down manner. The splitting point is the centroid of data objects projected to the principal vector. This saves a lot of computation for covariance matrixes and SVD decompositions as done in the ORCLUS algorithm. The sole dependence on the principal vector to separate data is rough and does not produce quality clusters. We design the $Redistribution$ function to minimise the projected energy of each clusters after clusters are formed by partitions.

The number of final clusters can be greater than k because the number of leaves is not tested until all data sets stored in the newest tree layer are split and redistributed.

Outliers affect the quality of final clusters very much since they change the orientations of data objects greatly. Some data objects may not belong to any cluster and are considered outliers. To deal with this problem, we set an outlier threshold in Redistribution step, say δ. When the projected distance of a data object to any cluster is greater than δ, then the data object is considered as an outlier and is excluded from the subsequent clustering.

We discuss the complexity of the algorithm in the following.

It is assumed that k denotes the number of final classes, N the number of data objects, d the dimension of the data set, and l the dimension of subspace.

The cost for partition is $kd^3/2$. d^3 comes from computing covariance matrices and SVD decomposition for a cluster. Since the partition is conducted in a binary way, the number of total partitions is k. Each partition requires a SVD decomposition to determine the subspace.

After the partition, each cluster has to be decomposed again to determine whether or not it satisfies the projected clustering requirement. If no, it will participate in redistribution. The number of such decomposition is $2k$, and hence the costs for the decompositions is $2kd^3$. All clusters are projected to l dimensional subspaces, and each data object has to be checked against each cluster. Therefore, the total costs for distribution is kNl.

In sum, the computational complexity for the DPCLUS algorithm is $O(3kd^3+kNd)$. Note that we could not do much for term d^3 since it is for computing a covariance matrix and a SVD decomposition. However, the proposed algorithm has reduced the number of such computations significantly.

Our DPCLUS algorithm is faster than most exiting generalised projected clustering algorithms. We compare the time complexities in the table below.

Algorithms	Time complexities
ORCLUS [1]	$O(k_0^3 + k_0Nd + k_0^2d_0^2)$
$4C$ [4]	$O(d^2NlogN + d^3N)$(with data index)
	$O(d^2N^2 + d^3)$(without data index)
CURLER [5]	$O(k_0Nd^2 + k_0d^3) + O(Nl^2 + k_0^2), where k_0 > k$
HARP [7]	$O(d^2N^2 + N^2log^2N)$
DPCLUS	$O(3kd^3 + kNd)$

It should be noted that in order to speed up, some algorithms make use of techniques such as heuristics, small number of micro-clusters, and random samples. However, these techniques come with a price; that is, some of the clustering quality must be sacrificed.

4 Experimental Evaluation

4.1 Efficiency Comparison to ORCLUS

We use synthetic data sets for this experiment. More details about how these data sets were generated will be given in the following subsection.

For the test of scalability with the size of data sets, the data sets each contain 10 attributes and up to 300,000 objects. For the test of scalability with the number of attributes, the data sets each contain 100,000 objects and up to 50 attributes. The number of embedded clusters is fixed to 20 for the above two tests.

l is set to 10 for both methods. k_0 is set to $15 \times k$ to make ORCLUS efficient. δ for both methods is set as 0.01. $\min_N$ varies for different data sets, but is set as the same for both methods. A value less than 0.0001 is consider as 0 in the experiments to test the satisfaction of Definition 1.

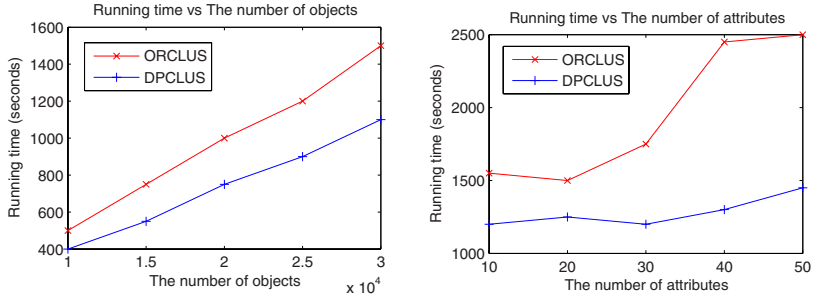


Fig. 1. The scalability of DPCLUS in comparison to that of ORCLUS

Figure 1 shows that DPCLUS is more efficient than ORCLUS in large data sets as well as in high dimensional data sets. Consider that most computational time for projected clustering is spent on data decomposition, whose time complexity is cubic to the dimension and independent of the data set size. DPCLUST outperforms ORCLUS significantly in high dimensional data sets since it reduces the number of data decompositions significantly.

4.2 Clustering Quality

To demonstrate the clustering quality of DPCLUS, we compare it to three clustering methods on a synthetic data set. We embedded 20 clusters that are correlated in some subspaces over a set of random data objects. The data set contains 10,000 data objects, with each object having 20 attributes. Each embedded cluster contains 250 data objects, which have 10% variations from the original pattern. Other 5,000 data objects are random data objects generated by the uniform distribution.

We set the parameters of DPCLUS as $l = 10$, $k = 20$, $\min_N = 50$, and $\delta = 0.01$. The results from DPCLUS are shown in Figure 2. DPCLUS is able to find all embedded clusters correctly. Although k is set as 20 in the experiment, the number of final clusters can be any integer number between 20 to 32, because the number of clusters is not tested until all data sets stored in the newest tree layer are split and redistributed. The number of the found clusters are greater than 20, since some clusters are split into two. For example, clusters at row 2: 1 and 2 are from the same cluster. DPCLUS has successfully identified cluster patterns from random data.

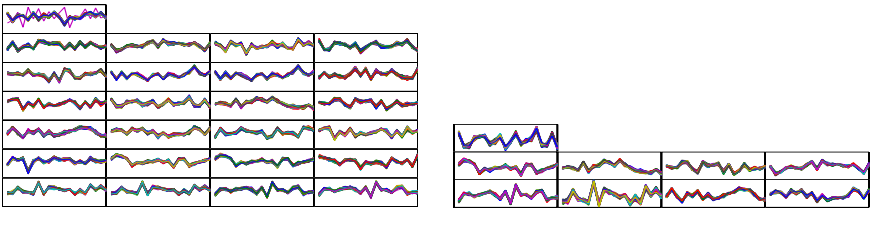


Fig. 2. Left: clusters found by DPCLUS, Right: clusters found by ORCLUS

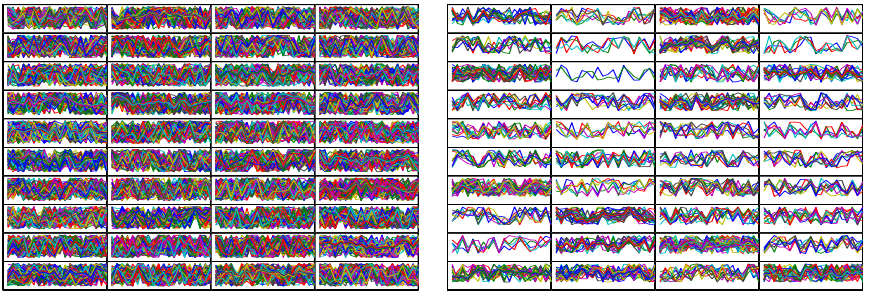


Fig. 3. Left: clusters found by kmeans. Right: Clusters found by hierarchical clustering.

We set the parameters of ORCLUS as $k = 20$, $l = 10$, $k_0 = 350$, and $\delta = 0.01$. Figure 2 shows a good result. ORCLUS identified fewer than a half of embedded clusters with high quality. Since initial micro-clusters in ORCLUS is randomly chosen, the final clusters vary in different executions.

We further show that both *k*-means and hierarchical clustering methods failed to find quality clusters in such noise data in Figure 3. Data is sampled for hierarchical clustering method because of efficiency constraint.

5 Conclusions

We have presented a divisive, projected clustering algorithm for detecting correlation clusters in highly noised data. The distinction of noise points from correlated data points in a projected space offers benefits for projected clustering algorithms to discover clusters in noise data. The proposed algorithm mainly explores this potential. Further, the proposed algorithm is faster than most existing general projected clustering algorithms, which are agglomerative clustering ones. Unlike those agglomerative algorithms, the produced clusters by the proposed algorithm do not rely on the choice of randomly generated initial seeds, and are completely determined by the data distribution. We experimentally show that the proposed algorithm is faster and more scalable than than ORCLUS, a well-known agglomerative projected clustering, and that the proposed algorithm detects correlation clusters in noise data better than ORCLUS.

References

1. C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data (SIGMOD)*, pages 70–81, 2000.
2. C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast algorithms for projected clustering. In *Proceedings of the ACM SIGMOD Conference*, pages 61-72, 1999.
3. R. J. Aggrawal, J. Gehrke, D.Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD Conference*, pages 94-105, 1998, Seattle, WA.

4. C. Bhm, K. Kailing, P. Krger, and A. Zimek. Computing Clusters of Correlation Connected objects. In *Proceedings of the ACM SIGMOD international conference on Management of data*, June 13-18,2004, Paris, France.
5. A. K. H. Tung, X. Xu, and B. C. Ooi. CURLER: finding and visualizing nonlinear correlation clusters. In *Proceedings of the ACM SIGMOD international conference on Management of data*,pages 467-478, 2005.
6. E. Ng, A. Fu, and R. Wong. Projective Clustering by Histograms. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 3, pages 369-383, March 2005.
7. K. Y. Yip, D. W. Cheung, and M. K. Ng. HARP: A Practical Projected Clustering Algorithm. *IEEE Transactions on Knowledge and Data Engineering* , Vol. 16, No. 11, pp. 1387-1397, Nov. 2004.
8. G. Taguchi, R. J. Taguchi, and R. Jugulum. *The Mahalanobis-Taguchi Strategy: A Pattern Technology System*. John Wiley & Sons, 2002.

Application of Discrimination Degree for Attributes Reduction in Concept Lattice

Ming Li and De-San Yang

School of Computer and Communication, Lanzhou University of Technology, Lanzhou, Gansu, 730050, P. R. China
Lim3076@163.com, ydsan@163.com

Abstract. This paper presents a new concept, discrimination degree theory, which is complementary of inclusion degree. Then the theoretical and practical significance of the discrimination degree is discussed, and the concept formation theorem of discrimination degree is given. The relationship between the discrimination degree and discernibility matrix is explained in attributes reduction of formal context. Finally, under a biology formal context, concept lattice is built after attributes reduced. By comparison with the lattice which didn't reduce attributes, it shows that reduction make the complexity of building lattice distinctly simplified while the key information is still retained.

Keywords: Concept lattice; Inclusion degree; Discrimination degree; Attributes reduction.

1 Introduction

Formal concept analysis is a mathematical framework developed by Rudolf Wille and his colleagues at Darmstadt/Germany that is useful for representation and analysis of data [1]. It is virtually a concept clustering process to build the concept from the formal context, so it is also a kind of classification [2, 3, 4, 5, 6]. In database, redundant information is useless for extracting rules, but can increase the complexity of lattice building. So attributes reduction is necessary. This paper presents a new method for the attributes reduction, which based on discrimination degree.

Inclusion degree theory has been proposed by Professor Wen-Xiu Zhang ten years ago, which is an effective measurement of uncertain relations [7]. It has generalized many methods of inference uncertain issue, such as probability inference method, evidence inference method, fuzzy inference method as well as information inference method and so on. Thus it provides a general principle for the uncertain inference. In addition, inclusion degree also offers a quantitative analysis method for the ordered mathematics theory. In some domains, such as artificial intelligence, expert system and rough set, the theory has been applied [8, 9]. Moreover, formal concept analysis is also an ordered mathematics theory. Inclusion degree has been brought into formal concept analysis in literature [8, 10].

After researching inclusion degree from the literature [7], we present discrimination degree theory which is complementary to inclusion degree, and give the attributes reduction algorithm that based on the new theory. The paper is organized as follows: In section 2, concept lattice and inclusion degree are introduced briefly. In section 3,

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 648–655, 2007.

discrimination degree is presented, the complementary relation with inclusion degree is exemplified, the significance is discussed and concept formation theorem based on discrimination degree is given, finally the relationship of the new degree with discernibility matrix is also shown. In section 4, the attributes reduction algorithm based on the new degree is indicated, and we get a conclusion that reduction can make the complexity of building lattice distinctly simplified after comparing with the lattice which didn't reduce attributes. Section 5 is the conclusion and future work.

2 Correlative Definitions

2.1 Formal Concept Analysis (FCA) Bases [1]

Definition 1. A formal context $K := (G, M, I)$ consists of two sets G and M , and a relation I between G and M . The elements of G are called the objects and the elements of M are called the attributes of the context. In order to express that an object g is in a relation I with an attribute m , we write gIm or $(g, m) \in I$ and read it as "the object g has the attribute m ".

Definition 2. For a set $A \subseteq G$ of objects we define $A' := \{ m \in M \mid gIm$ for all $g \in A \}$ (the set of attributes common to the objects in A). Correspondingly, for a set B of attributes we define $B' := \{ g \in G \mid gIm$ for all $m \in B \}$ (the set of objects which have all attributes in B).

Definition 3. A formal concept of the context $K := (G, M, I)$ is a pair (A, B) with $A \subseteq G, B \subseteq M$, $A' = B$ and $B' = A$. We call A the extent and B the intent of the concept (A, B) .

Definition 4. A formal context $K := (G, M, I)$, for $g \in G$, we define the corresponding object concept as: $\tilde{\gamma} g := (\{g\}'', \{g\}')$. In the same way, for $m \in M$, we define the corresponding attribute concept as: $\tilde{\mu} m := (\{m\}', \{m\}'')$. For convenient, we replace $\{g\}'$ by g' and $\{m\}'$ by m' . All object concepts compose the set of $\gamma(G)$, and all attribute concepts compose $\mu(M)$.

2.2 Inclusion Degree

Definition 5. The inclusion degree D of two sets is defined as the degree of one set contained in another set [7].

X is universe, A and B are two subsets of X, the degree that set A is contained by set B is called an inclusion degree, denoted as $D(B / A) = \dfrac{|A \bigcap B|}{|A|}$, where $|\cdot|$ represents the number of elements of the set. In general, it should satisfy the following conditions:

(1) $0 \le D(B/A) \le 1$;

(2) $D(B/A) = 1$ if and only if $A \subseteq B$;

(3) $D(A/C) \le D(A/B)$ if and only if $A \subseteq B \subseteq C$;

(4) For $\forall\ C$, $D(A/C) \le D(B/C)$ if and only if $A \subseteq B$.

Condition (1) is the standardization of inclusion degree in [0, 1]; Condition (2) indicates the coordination between inclusion degree and the classical inclusion relation, classical inclusion is 1 or 0 which is the special case of inclusion degree; Condition (3) and (4) show the monotone, it is to say that a smaller set is more easily contained by other sets than a larger set.

3 Discrimination Degree Theory

3.1 Definition of Discrimination Degree

Definition 6. The discrimination degree $\bar{D}$ between two sets is defined as the degree of one set differentiates from the other set.

X is universe, A and B are two subsets of X, the degree that set A differentiates to set B is called a discrimination degree, denoted as $\bar{D}(B/A) = \frac{|A \cap \bar{B}|}{|A|}$, where $|\cdot|$ also represents the number of elements of the set. In general, it should satisfy the following conditions:

(1) $0 \le \bar{D}(B/A) \le 1$;

(2) $\bar{D}(B/A) = 0$ if and only if $A \subseteq B$;

(3) $\bar{D}(A/B) \le \bar{D}(A/C)$ if and only if $A \subseteq B \subseteq C$;

(4) For $\forall C$, $\bar{D}(B/C) \le \bar{D}(A/C)$ if and only if $A \subseteq B$.

Like inclusion degree, so we have that condition (1) is the standardization of discrimination degree value in [0, 1]; Condition (3) and (4) show the monotone, it means that a smaller set is more easily differentiated from other sets than a larger one, which accords with people habit.

Example. X is a finite set, subsets $A, B, C \subset X$, Records X ={x1,x2,x3,x4, x5,x6,x7}, A ={x1,x2,x3,x4}, B ={x3,x4,x5,x6,x7}, C ={x1,x3,x4,x5,x6,x7}.

From the above definition, $D(C/A) = \frac{|A \cap C|}{|A|}$, $D(B/A) = \frac{|A \cap B|}{|A|}$ are inclusion degrees; And $\bar{D}(C/A) = \frac{|A \cap \bar{C}|}{|A|}$, $\bar{D}(B/A) = \frac{|A \cap \bar{B}|}{|A|}$ are discrimination degrees; $\bar{C}$ is the complement set of C.

After calculating, we have $D(C/A)$ =3/4, $D(B/A)$ =2/4; $B \subset C$, $D(C/B)$ =1; $D(B/A) < D(C/A)$. $\overline{D}(C/A)$ =1/4, $\overline{D}(B/A)$ =2/4, $\overline{D}(C/B)$ =0, $\overline{D}(A/B)$ =3/5. So $\overline{D}(C/A) + D(C/A) = \overline{D}(B/A) + D(B/A)$ =1, which exemplifies the complementarity of two degrees; for $B \subset C$, $\overline{D}(C/B)$ =0 satisfies Condition (2), and $\overline{D}(C/A) < \overline{D}(B/A)$ satisfies Condition (4).

3.2 The Significance of Discrimination Degree

From the above we can see, discrimination degree is complementary to inclusion degree and a quantitative measurement for uncertain relations. In fact, it is a traditional cognitive process. For example, children know that apples and oranges are juicy, sweet and have delicious flavor. They can distinguish potato out from fruit. But it can't be accomplish only by the common characters such as juicy, sweet and delicious flavor when people divide apples and oranges from a heap mixture of apples and oranges. So the separation must rely on the differences in shape, color, size and rough degree. It will enhance classification accuracy if discrimination degree and inclusion degree are combined. Another example is to identify twin brothers, we can easily identify that they come from the same family (by inclusion degree), then distinguish their differences (by discrimination degree), compare them by the memory features (in the formal context), we will be able to tell who is the elder and who is the younger (classification).

The two examples above show the importance of differences among the concepts, which is our original intention to present the discrimination degree. We hope to improve classification accuracy and supplement to inclusion degree theory by using the new degree.

3.3 Concept Formation Theorem Based on Discrimination Degree

Theorem. Given a formal context $K := (G, M, I)$, for $\forall g \in G$ and $\forall m \in M$, then
(1) $g'' = \{\tilde{g} \in G \mid \tilde{g} \le g\} = \{\tilde{g} \in G \mid \overline{D}(g/\tilde{g}) = 0\}$,
(2) $m'' = \{\tilde{m} \in M \mid m \le \tilde{m}\} = \{\tilde{m} \in M \mid \overline{D}(\tilde{m}/m) = 0\}$.

Proof. For $\overline{D}(g/\tilde{g}) = 0 \Leftrightarrow |\overline{g} \cap \tilde{g}| = 0 \Leftrightarrow \tilde{g} \subseteq g \Leftrightarrow \tilde{g} \le g$, so we need only prove $g'' = \{\tilde{g} \in G \mid \tilde{g} \le g\}$. $\tilde{g} \le g \Leftrightarrow \tilde{g}' \supseteq g \Leftrightarrow \tilde{g}'' \subseteq g'' \overset{\tilde{g} \in \tilde{g}''}{\Longleftrightarrow} \tilde{g} \in g'' \Rightarrow \{\tilde{g} \in G \mid \tilde{g} \le g\} \subseteq g''$.

When $\tilde{g} \in g''$, for $\forall g_1 \in g''$, $(g'')' \in g_1'$, we have $g_1 \le g$ from definition 2, so $g_1 \in \{\tilde{g} \in G \mid \tilde{g} \le g\}$ and $\{\tilde{g} \in G \mid \tilde{g} \le g\} \subseteq g''$. In sum, $g'' = \{\tilde{g} \in G \mid \tilde{g} \le g\}$. And the second section can be proved in the same way.

From the theorem, the set composed by all objects which are less than or equal to a certain object is the extent of the corresponding object in a concept, and the set composed by all attributes which are more than or equal to a certain attribute is the

intent of the corresponding attribute in a concept [10]. Thus, the extent and intent constitute a new concept.

3.4 The Relationship Between Discrimination Degree and Discernibility Matrix

In Rough set theory, discernibility matrix is used to reduce the redundant data [11]. And some reduction is implemented by the matrix in FCA. Based on the differences among concepts we present discrimination degree. So we research on the two theories to find their relations.

The discernibility matrix $C_{D_{M\cdot N}} = (C_{ij})$, where C_{ij} are elements of matrix, composed by the discriminative attributes between objects g_i and g_j, and $1 \le i, j \le N$, N is the number of objects in the formal context.

And discrimination degree here is: $\overline{D}(B/A) = \dfrac{A \cap \overline{B}}{A}$, where A, B represent the elements sets, and $\overline{B}$ represents the complement set of B.

In a formal context $\mathrm{K} := (G, M, I)$, $g_i \in G$, $g_i^{'} \subseteq M$, g_i' is the attribute set of object g_i. Using discrimination degree, we get C_{ij} by:

$$C_{ij} = g_j' \cdot \overline{D}(g_i'/g_j') + g_i' \cdot \overline{D}(g_j'/g_i') = \overline{g}_i' \bigcap g_j' + \overline{g}_j' \bigcap g_i'$$

4 Building Concept Lattice

For a large number of redundant information, it is very complicated to build lattice directly without reduction. So it is necessary to remove redundant attributes and extract core attributes [4, 12]. The algorithm uses discrimination degree for attribute reduction, thus the formation of concept nodes and lattice construction can be simplified.

Following is the algorithm:

Step1: Each element of discernibility matrix is obtained by:

$$C_{ij} = g_j' \cdot \overline{D}(g_i'/g_j') + g_i' \cdot \overline{D}(g_j'/g_i')$$

Step2: Disjunctive logic expressions are composed by elements which are nonempty sets ($C_{ij} \neq 0, C_{ij} \neq \Phi$, a_i isn't the intercommunity attribute between object g_i and g_j), denoted as $L_{ij} = \bigvee\limits_{a_i \in C_{ij}} a_i$;

Step3: All the disjunctive logic expressions L_{ij} intersect to get $L = \bigwedge\limits_{C_{ij} \neq 0, C_{ij} \neq \Phi} L_{ij}$, then core attributes and redundant attributes can be obtained;

Step4: Then L is transformed to the disjunctive normalize formulation $L' = \bigvee\limits_i L_i$;

Step5: According to the result of reduction, concepts are formed based on the concept formation theorem which depicted in section 3.3: for $\forall i \in N$

1° Formation of concept extent: if $\overline{D}(g_i / g_j) = 0$ then $g_i'' = \{ g_j \}, \forall j \in N$,

2° Formation of concept intent: if $\overline{D}(m_j / m_i) = 0$ then $m_i'' = \{ m_j \}, \forall j \in N$:

Step6: Concept lattice is built according to the algorithm given in literature [13].

Table 1. A biology field formal context

	nw	lw	ll	nc	2lg	1lg	mo	lb	sk
Leech	1	1	0	0	0	0	1	0	0
Bream	1	1	0	0	0	0	1	1	0
Frog	1	1	1	0	0	0	1	1	0
Dog	1	0	1	0	0	0	1	1	1
Spike-weed	1	1	0	1	0	1	0	0	0
Reed	1	1	1	1	0	1	0	0	0
Bean	1	0	1	1	1	0	0	0	0
Maize	1	0	1	1	0	1	0	0	0

Table 1 is a formal context about biology field, for convenient, objects are recorded as {1,2,3,4,5,6,7,8}, and attributes(nw—need water; lw—lives in water; ll—lives on land; nc—needs chlorophyll; 2lg—2 leaf germination; 1lg—1 leaf germination; mo—is motile; lb—has limbs; sk—suckles young) are recorded as {a,b,c,d,e,f,g,h,i}.

Through the beginning three steps, we have the following:

L =h ∧ (c ∨ h) ∧ (b ∨ c ∨ h ∨ i) ∧ (d ∨ f ∨ g) ∧ (c ∨ d ∨ f ∨ g) ∧ (b ∨ c ∨ d ∨ e ∨ g) ∧ (b ∨ c ∨ d ∨ f ∨ g) ∧ c ∧ (b ∨ c ∨ i) ∧ (d ∨ f ∨ g ∨ h) ∧ (c ∨ d ∨ f ∨ g ∨ h) ∧ (b ∨ c ∨ d ∨ e ∨ g ∨ h) ∧ (b ∨ c ∨ d ∨ f ∨ g ∨ h) ∧ (b ∨ i) ∧ (c ∨ d ∨ f ∨ g ∨ h) ∧ (d ∨ f ∨ g ∨ h) ∧ (b ∨ d ∨ e ∨ g ∨ h) ∧ (b ∨ d ∨ f ∨ g ∨ h) ∧ (b ∨ c ∨ d ∨ f ∨ g ∨ h ∨ i) ∧ (b ∨ d ∨ f ∨ g ∨ h ∨ i) ∧ (d ∨ e ∨ g ∨ h ∨ i) ∧ (d ∨ f ∨ g ∨ h ∨ i) ∧ c ∧ (b ∨ c ∨ e ∨ f) ∧ (b ∨ c) ∧ (b ∨ e ∨ f) ∧ b ∧ (e ∨ f)=b ∧ c ∧ h ∧ (e ∨ f) ∧ (d ∨ f ∨ g),

Where (b,c,h) are core attributes, (a,i) are redundant attributes and (d,e,f,g) are the attributes that can be omitted;

Step4: The output is L' =bchef ∨ bched ∨ bcheg ∨ bchf ∨ bchfd ∨ bchfg=bchf ∨ bched ∨ bcheg. So we have three choices;

Step5: We choose (b,c,e,h,g) to form concept nodes according to the concept formation theorem;

Step6: Building concept lattice, as Fig.2.

Without attributes reduction, there are 19 concept nodes and 30 lines in Fig.1. But in Fig.2, we reduce the formal context firstly, then 14 concept nodes are formed according to the concept formation theorem, and there are only 22 lines in the whole lattice. Moreover the concept nodes (6, bc) and (4, cgh) can be removed, while 4 correlative lines also can be subtracted from the lattice. According to the formal context in Table 1, the concept node (7, ce) still can be identified Bean exclusively and node (3, bcgh) can

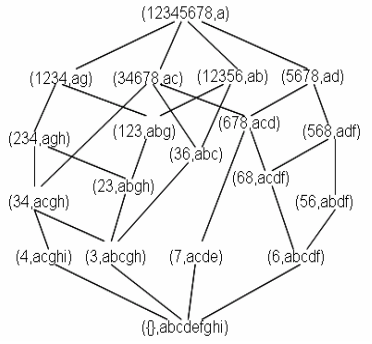


Fig. 1. Concept lattice without reduction

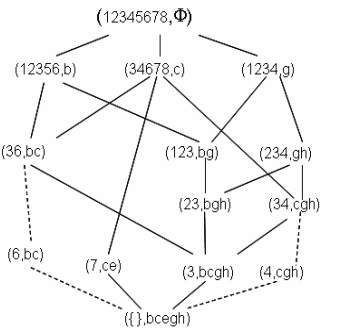


Fig. 2. Concept lattice based on reduction

be identified only Frog in Fig.2. By comparison, the complication of lattice building is greatly simplified after reduction but the key information is retained.

5 Conclusion

This paper presented discrimination degree theory based on inclusion degree theory, and proved the complementary relationship between the two degrees, then discussed the theoretical and practical significance of the new theory. For attributes reduction, the relationship between the discrimination degree and discernibility matrix was explained. By comparing, the new method had greatly simplified the complexity for building lattices.

At information age, intelligence and high efficiency of information processing have been put on the agenda. Attributes reduction of high-dimensional information is an effective way to improve the efficiency of information processing. It has been still a hot topic in the study [9, 14]. Moreover, the discrimination degree theory need to further perfect, applying the new degree into engineering and processing some indefinite information are future work.

Acknowledgement. This paper is supported by Natural Science Foundation of Gansu, China.

References

1. Ganter B., Wille R.: Formal Concept Analysis: Mathematical Foundations. Berlin: Springer- Verlag (1999)
2. Huai-Guo Fu, Engelbert Mephu Nguifo.: A Parallel Algorithm to Generate Formal Concepts for Large Data. ICFCA, LNAI 2961, P. Eklund (2004) 394–401
3. Y.Y. Yao.: Neighborhood systems and approximate retrieval, Information Sciences, 176 (2006) 3431–3452
4. Hong-Ru Li, Wen-Xiu Zhang, Hong Wang.: Classification and reduction of attributes in concept lattices. 2006 IEEE International Conference on Granular Computing, Atlanta: IEEE (2006)142–147

5. A. Jaoua, S. Elloumi.: Galois connection, formal concepts and Galois lattice in real relations: application in a real classifier, The Journal of Systems and Software, 60 (2002) 149–163
6. Besson J., Robardet C., Boulicaut J.F.: Mining formal concepts with a bounded number of exceptions from transactional data. In: Proceedings KDID' 04. Volume 3377 of LNCS, Springer-Verlag (2004) 33–45
7. Wen-Xiu Zhang, Zong-Ben Xu, Yi Liang etal.: Inclusion degree theory. Fuzzy Systems and Mathematics, 10 (1996) 1–9 (in Chinese)
8. Ju-Sheng Mi, Wen-Xiu Zhang, Wei-Zhi Wu.: Optimal Decision Rules Based on Inclusion Degree Theory. Proceedings of the First International Conference on Machine Learning and Cybernctics, Beijing: IEEE (2002)1223–1226
9. Peter Burmeister, Richard Holzer.: Treating Incomplete Knowledge in Formal Concept Analysis. B. Ganter et al. (Eds.): Formal Concept Analysis, LNAI 3626 (2005) 114–126
10. Kai-She Qu, Yan-Hui Zhai.: Posets, Inclusion Degree Theory and FCA. Chinese journal of computers, 29 (2006) 219–226 (in Chinese)
11. Skowron A, Rauszer C.: The discernibility matrices and functions in information system. Kluwer Academy Publishers (1992)
12. Petko Valtchev, Rokia Missaoui, Robert Godin.: Formal Concept Analysis for Knowledge Discovery and Data Mining: The New Challenges. ICFCA, LNAI, 2961, P.Eklund (2004) 352–371
13. Godin R, M issaoui R, A laoui H.: Incremental concept formation algorithms based on Galois (concept) lattices. Computational Intelligence, 11 (1995) 246–267
14. Y.Y. Yao.: Concept lattices in rough set theory. In: Proceedings of 2004 Annual Meeting of North American Fuzzy Information Processing Society, Canada (2004) 796–801

A Language and a Visual Interface to Specify Complex Spatial Patterns

Xiaohui Li and Yan Huang

Computer Science and Engineering
University of North Texas
{xl0023, huangyan}@unt.edu

Abstract. The emerging interests in spatial pattern mining lead to the demand for a flexible spatial pattern mining language, on which easy to use and understand visual pattern language could be built. This motivates us to define a pattern mining language called CSPML to allow users to specify complex spatial patterns they are interested in mining from spatial datasets. We describe our proposed pattern mining language in this paper. Unlike general pattern languages proposed in literature, our language is specifically designed for specifying spatial patterns. An interface which allows users to specify the patterns visually is designed. The visual language is based on and goes beyond the visual language proposed in literature in the sense that users use CSPML to retrieve patterns instead of the results of a simple spatial query.

1 Introduction

Mining implicit, potentially useful, and previously unknown patterns from spatial data is becoming increasingly important due to the availability of large spatial database collected by inexpressive location enabled devices such as GPS and RFID. The applications of spatial data mining range from animal movement tracking, environmental monitoring, transportation, to national security [6,14].

Mining *spatial co-location patterns* [9,15,7,17] is an important spatial data mining task with broad applications. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ be a set of spatial features. A spatial feature categorizes or groups spatial objects that have the same characteristics together. Example spatial features include car accident, traffic jam, Chromium 6 polluted water source, West Nile disease, and deforestation. Consider a number of l spatial datasets $\{SD_1, SD_2, \ldots, SD_l\}$, such that $SD_i, i \in [1, l]$ contains all and only the objects that have the spatial feature f_i, e.g. the spatial objects of West Nile disease. Let $\mathcal{R}$ be a given spatial neighbor relation (e.g. distance less than 1.5 miles). A set of spatial features $X \subseteq \mathcal{F}$ is a co-location if its value $im(X)$ of an interesting measure, which is specified differently by variants of the mining problem, is above a threshold min_im. The problem of finding the complete set of co-location patterns is called the co-location mining problem.

Recently, co-location pattern has been extended to include positive relationships, self-co-location/ self-exclusion relationships, one-to-many relationships,

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 656–663, 2007.

and multi-feature exclusive relationships [2]. Mining complex spatial patterns from large spatial datasets is an emerging area in data mining [12]. Many spatial features interact with each other through complex spatial relationships, e.g. topological and directional [16], other than metric ones. A complex spatial pattern refers to a subset of spatial features whose spatial objects tend to appear in some spatial configuration specified by some spatial predicate. An example complex spatial pattern may be (car accident, yield sign, service road) with the spatial predicate (yield sign *on* service road AND car accident *close* yield sign). These complex spatial patterns may be summarized by the interactions of their spatial objects. When happenings of this configuration are observed and the pattern is significant enough according to some *significant measure*, we report such pattern.

Most current spatial data mining algorithms follow an almost exhaustive exploration of the problem space and recommends a large set of non-trivial, potentially useful, and previously unknown patterns. This mining process entails costly computational cost. However, most of the time domain experts posses some basic knowledge about the data and this knowledge may be used to guide the mining process. So a spatial data mining language that is both expressive and easy to use will allow domain experts to incorporate their interests and knowledge to specify complex spatial patterns. In this patter, we describe such a language and a visual interface to accept visual query which will be translated into the proposed language.

We make the following two contributions in this paper. First, we provide a spatial pattern language called CSPML to specify complex spatial pattern. The language allows users to specify a large set of complex patterns that they are interested in. Second, we design a visual interface to relieve users from writing CSPML queries.

2 Related Work

Existent data mining query languages are not sufficient for spatial data mining. DBQL (Data Mining Query Language) [8] is a general purpose data mining language and does not allow expression of spatial patterns. SDMOQL (Spatial Data Mining Object Query Language) [11] is a spatial mining language designed for INGENS(Inductive Geographic Information System). It trains the inductive geographic information system by inductive queries first and creates a special user view. SDMOQL focuses more on querying spatial database than retrieving spatial patterns. ATLaS (Aggregate and Table Language and System) [10] is an native extension of SQL language which adds to SQL the ability of defining new User Defined Aggregates and Table Functions. It extends DBMS to support efficiently database-centric data mining, stream computation, and decision support applications. Snoop [3] is a model independent event specification language. It distinguishes events from conditions of event and allows the construction of complex events needed for a large class of applications.

A user friendly visual interface to alleviate users from the burden of writing sophisticated query language, is important for both spatial query language and

spatial pattern mining. Topological relationships among arbitrary spatial objects using 9-intersection model is discussed in [5]. Regions, lines and points and all the possible relations between them were defined [5] using point topology. Spatial-Query-by-Sketch tries to offer an efficient visual interface for retrieving spatial objects satisfying a spatial query in geographic information systems [4]. Twelve positional operators and a set of their specifications in object-oriented geographic database are considered in PQL(Pictorial Query Language) in [13]. The system developed in this context aims at providing a visual interface with the expressive power of a database query language such as SQL. It concerns less about spatial pattern mining where users are more interested in patterns than spatial objects.

3 Complex Spatial Pattern Query Language

In this section, we introduce the definition of the proposed pattern mining language for complex spatial patterns (CSPML). The formal definition of CSPML is of the form:

FIND t
WHERE $P(t)$
WITH *interestingness* θ c

where t is variable to present an ordered (alphabetically by spatial feature name) subset of all the spatial features in $\mathcal{F}$, P is a *formula*, *interestingness* specifies the interestingness measures used for the complex spatial pattern, θ is a comparison operator that includes $<, \leq, >$, and $\geq$, and c is a constant to specify the interestingness measure threshold. The formula $P(t)$ is built up using conjunctive forms, which may be one of the following:

1. $t.size\ \theta\ c'$ where $t.size$ is the cardinality of the spatial feature set t and c' is a constant integer;
2. $Q(t)$ where $Q(t)$ is a constraint a conjunctive form with components from the following format:
   (a) $t[i] \odot t[j], \forall i, j \in integer$
   (b) $t[i].type \in \{geo\_point, geo\_line, geo\_area, geo\_donotcare\}, \forall i \in integer$
   where $\odot$ is a spatial predicate, e.g. *overlap*, and $t[k]$ specifies the kth spatial feature in an ordered set t. A metric spatial predicate $\odot$ has an attribute attached which denotes distance information; There are three types of spatial features in a geography database, namely geo_point, geo_line and geo_area. Every spatial feature in a pattern has one attribute called *type*. If the *type* of a feature is known, user can provide the type. For unknown types, we introduce a new type called *geo_do_not_care*, which means any one of the three types.
3. $s \subseteq t\{\bigwedge Q'(s,t)]\}$, where s is any ordered (again alphabetically) subset of the spatial feature set $\mathcal{F}$ specified by a user; $[Q'(s,t)]$ is an optional constraint consisting one of the following form or their conjunctive combinations:
   (a) $(t-s)[i] \odot (t-s)[j], \forall, i, j \in integer$;
   (b) $(t-s)[i] \odot s[j], \forall, i, j \in integer$;

(c) $s[i] \odot s[j] \forall, i, j \in integer$;
where $\odot$ is a spatial predicate, e.g. overlap, and $t[k]$ specifies the kth spatial feature in an ordered set t

4. $t.pairwise(\odot)$, meaning pairwise spatial features in t satisfying spatial predicate $\odot$

Spatial predicate $\odot$ in the formula $P(t)$ may be metric, topological, and directional [16] relationships. The *interestingness* measure can be anything that is well defined for a complex spatial pattern, such as participation index [9], join cardinality, join selectivity and support [17]. We now describe a few examples using the proposed language to illustrate its expressiveness:

1. Find all patterns with less than 5 spatial features, spatial predicate pairwise distance less than 5 miles, and interestingness measure participation-index with threshold 0.7.

   FIND t
   WHERE t.size $\leq 5 \bigwedge$ t.pairwise(close(5))
   WITH $participation\_index \geq 0.7$

2. Find all patterns which contains 3 features and one feature is Chromium 6 polluted water. The Chromium 6 polluted water contains the other two features and the other two features are close to each other. The interestingness is participation-index and the threshold is 0.7. In this example, the pattern size is three with one as feature Chromium 6 polluted water, which belongs to set s. The other two features, which are the components of set $(t - s)$, are close to each other and inside the Chromium 6 polluted water.

   FIND t
   WHERE t.size $= 3 \bigwedge$ (Chromium 6 polluted water) $\subseteq$ t $\bigwedge$
   (t - s)[1] *containedBy* chromium 6 polluted water $\bigwedge$
   (t - s)[2] *containedBy* chromium 6 polluted water $\bigwedge$
   (t - s)[1] *close_to* (t - s)[2]
   WITH $participation - index \geq 0.7$

3. Find all patterns with three features two of which are forest and river. The other feature is to the north of the forest and close to the river. The interestingness is support and the threshold is 100. The pattern size in this example is three with two known features. So set s contains forest and river. The other feature has relationships with both of the two features through *north of* and *close* respectively.

   FIND t
   WHERE t.size $= 3 \bigwedge$ forest, river) $\subseteq$ t $\bigwedge$
   (t - s)[1] $northofforest \bigwedge$
   (t - s)[1] $close(defaultlength)$ river
   WITH $support \geq 100$

4. Find all the patterns of sizes from 3 to 5 and contain feature volcano. One feature is close to volcano in the pattern. The interestingness is *support* and

the threshold is 25. The pattern size is between three to five with volcano in set s. One feature in the set $(t - s)$ must be close to the feature volcano.

FIND t
WHERE t.size $\geq 3 \bigwedge$ t.size $\leq 5 \bigwedge$ (volcano) $\subseteq$ t $\bigwedge$ (t - s)[1] *close_to* volcano
WITH *support* ≥ 25

4 Visual Pattern Expression

There are three kinds of spatial features in spatial databases, namely geo-point, polyline and geo-area [13]. Geo-point is a zero dimensional point to represent events such as location of an accident. Polylines are used to represent spatial objects such as roads and rivers. Geo-area can be used to represent forest stands or lakes. We consider the meaningful relationships among these three features. In all figures in this section, geo-point, polyline and geo-area are represented a dot, a straight line (we do not consider polylines that can turn around), and a rectangle (without holes) respectively. Due to space constraint, we could not describe the prototype system that we implemented based on JUMP [1] architecture and the four examples specified using our visual interface in this paper. Interested readers should refer to the full version of the paper available from the second author's website.

Geo-point. Figure 1(a) and Figure 1(b) represent a geo-point *on* and *close to* a polyline respectively. A buffer is created around the polyline to indicate the *close to* relationship. The point may stand for an accident while the polyline may stand for a road. Figure 1(a) shows the accident happened on the roads while Figure 1(b) shows that although geo-point is not on the polyline but is still within certain distance. We use the representation in Figure 1(c) to specify that we do not care about the relationship between the geo-point and polyline, i.e. we want all patterns with any spatial predicate, e.g. *on* or *close to*.

Figure 1(d), 1(e) and 1(f) are the three kinds of relationships between a geo-point and a geo-area. They are geo-point *in, on* and *close to* the geo-area respectively. If a user does not care exactly what kind of relationship is between the geo-point and the geo-area, users could use the representation of figure 1(g) to find the pattern whose features may be any one of the three relationships.

Polyline. The relationship of a polyline and goe_area is based on the interiors, boundaries and exteriors of the objects [5]. A spatial relationship can be specified

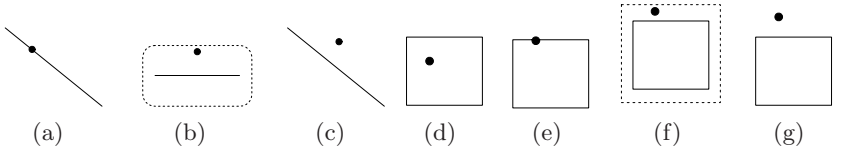


Fig. 1. The Expression of Relationships for Geo-point Objects

by a matrix representing if the interior(A°)/boundary(∂A)/exterior(A^-) of one object A intersects with the interior/boundary/exterior of another object B. Thirty-three possible relationships between two simple polylines were considered in [5]. We only consider straight lines and do not distinguish the boundaries from the interiors of the polyline, only five possible relationships between two simple polyline remain with four of them shown in figure 2 and the symmetric relationship of 2 (d) omitted due to space constraint.

The relationships specified in 2(b) to 2(d) can be further abstracted into a high level relationship called *intersect*, depending if the two polyline have any common points or not as shown in Figure 3(b). Users may define the a intersection relationship by using the specification as in figure 3(a). Otherwise, users may define specify the relationship by that in Figure 3(a). Figure 3(c) means the two polylines do not intersect while Figure 3(c) is used to express any kind of the relationships between two polylines.

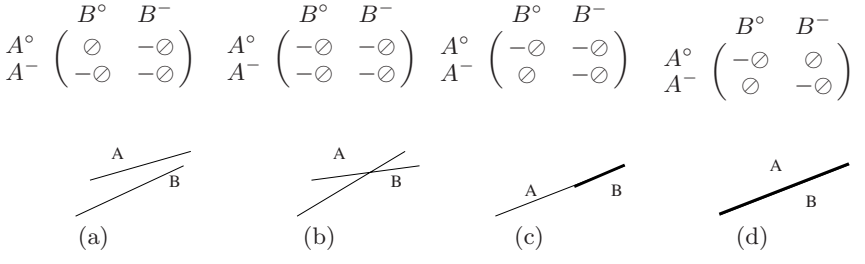


Fig. 2. Four Intersection Model between Two Straight Polylines

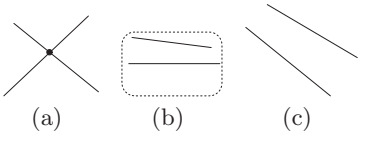


Fig. 3. The Expression of Relationships between Two Polylines

Based on the twenty relationships between a polyline and a geo-area [5], we found that the exteriors of a polyline always intersect with the interiors, boundaries and exteriors of a geo-area. We do not distinguish the boundary from the interior of a polyline, so we only consider the relationships between the interiors of a polyline with a geo-area. Figure 4 shows the matrix representation and the five relationships between a polyline and a geo-area. Figure 4(b) and 4(c) can be combined into a *touch* relationship as show in Figure 5(b).

The polyline could be *in, touch, across,* and *out of* a geo-area which are shown in Figure 5. If a user wants to find all relationships involving a polyline and a geo-area without providing a certain kind of relationship between them, they could just use the specification in figure 5(e).

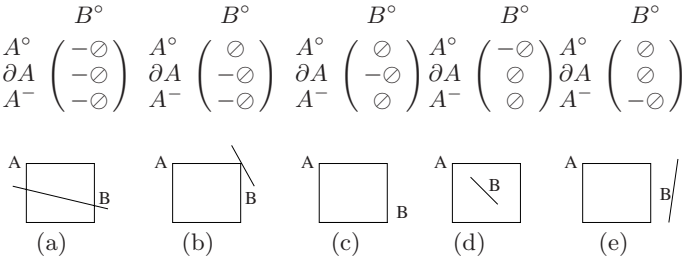


Fig. 4. Three intersection Model between a Polyline and a Geo-area

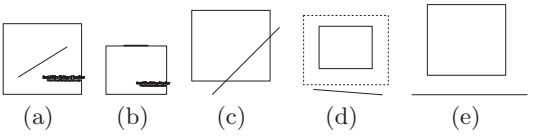


Fig. 5. The Expression of Relationships between a Polyline and a Geo-area

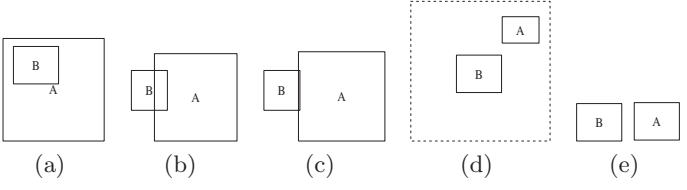


Fig. 6. The Expression of Relationships between Two Geo-areas

Geo-area. We allow the specification of 5 relationships between two geo-areas out of the 8 relationships specified in [5] as shown from Figure 6(a) to Figure 6(d) (the symmetric relationship of Figure 6(a) is omitted). If the users do not care the exact relationship between two features, they could specify the relationship as in figure 6(e).

5 Conclusion and Future Work

In this paper, we designed and developed a pattern mining language called CSPML for specifying complex spatial patterns. CSPML allows users to choose between exhaustive pattern search and targeted pattern mining constrained by prior knowledge about the datasets. We also proposed a pictorial language which has the equivalent expressive power to the proposed CSPML language. Efficient mining algorithms utilizing as much as the JTS tools of JUMP [1] will be implemented in the future.

References

1. The JUMP project. http://www.jump-project.org/index.php.
2. B. Arunasalam, S. Chawla, P. Sun, and R. Munro. Mining complex relationships in the sdss skyserver spatial database. In *COMPSAC '04*, pages 142–145, Washington, DC, USA, 2004.
3. S. Chakravarthy and D. Mishra. Snoop: An expressive event specification language for active databases. *Data Knowledge Engineering*, 14(1):1–26, 1994.
4. M. J. Egenhofer. Query processing in spatial-query-by-sketch. *Journal of Visual Languages and Computing*, 8(4):403–424, 1997.
5. M. J. Egenhofer and J. Herring. Categorizing binary topological relationships between regions, lines and points in geographic databases, 1991.
6. M. Ester, H.-P. Kriegel, and J. Sander. Algorithms and applications for spatial data mining, 2001.
7. V. Estivill-Castro and A. Murray. Discovering Associations in Spatial Data - An Efficient Medoid Based Approach. In *Proc. of the Second Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1998.
8. J. Han, Y. Fu, W. Wang, K. Koperski, and O. Zaiane. Dmql: A data mining query language for relational databases. *SIGMOD'96 Workshop*, 1996.
9. Y. Huang, S. Shekhar, and H. Xiong. Discovering Co-location Patterns from Spatial Datasets:A General Approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(12), 2004. (A summary of results was published at SSTD 2001).
10. C. Luo, H. Thakkar, H. Wang, and C. Zaniolo. A native extension of sql for mining data streams. In *SIGMOD '05*, 2005.
11. D. Malerba, A. Appice, and N. Vacca. Sdmoql: An oql-based data mining query language for map interpretation tasks. In *Proc. of the Workshop on Database Technologies for Data Mining (DTDM'02),in conjunction with the VIII International Conference on Extending Database Technology (EDBT'02)*, 2002.
12. R. Munro, S. Chawla, and P. Sun. Complex spatial relationships. *ICDM*, 00:227, 2003.
13. E. Pourabbas and M. Rafanelli. A pictorial query language for querying geographic databases using positional and olap operators. *SIGMOD Rec.*, 31(2):22–27, 2002.
14. J. Roddick and M. Spiliopoulou. A Bibliography of Temporal, Spatial and Spatio-temporal Data Mining Research. *ACM SIGKDD Explorations*, 1999.
15. S. Shekhar and Y. Huang. Co-location Rules Mining: A Summary of Results. In *Proc. 7th Intl. Symposium on Spatio-temporal Databases*, 2001.
16. A. P. Sistla and C. Yu. Reasoning about qualitative spatial relationships. *J. Autom. Reason.*, 25(4):291–328, 2000.
17. X. Zhang, N. Mamoulis, D. W. Cheung, and Y. Shou. Fast mining of spatial collocations. In *KDD '04*, pages 384–393, 2004.

Clustering Ensembles Based on Normalized Edges*

Yan Li[1], Jian Yu[2], Pengwei Hao[1,3], and Zhulin Li[1]

[1] Center for Information Science, Peking University,
Beijing, 100871, China
{yanli, lizhulin}@cis.pku.edu.cn
[2] Inst. of Computer Science & Engineering, Beijing Jiaotong Univ.,
Beijing, 100044, China
jianyu@center.njtu.edu.cn
[3] Dept. of Computer Science, Queen Mary, Univ. of London,
London, E1 4NS, UK
phao@dcs.qmul.ac.uk

Abstract. The co-association (CA) matrix was previously introduced to combine multiple partitions. In this paper, we analyze the CA matrix, and address its difference from the similarity matrix using Euclidean distance. We also explore how to find a proper and better algorithm to obtain the final partition using the CA matrix. To get more robust and reasonable clustering ensemble results, a new hierarchical clustering algorithm is proposed by developing a novel concept of normalized edges to measure the similarity between clusters. The experimental results of the proposed approach are compared with those of some single runs of well-known clustering algorithms and other ensemble methods and the comparison clearly demonstrates the effectiveness of our algorithm.

1 Introduction

Cluster analysis is an important tool for exploratory data analysis, aiming to find homogeneous groups in a data set of unlabeled objects. Numerous algorithms have been and are being developed [2], [9], [10], such as the K-means (KM), the single-linkage (SL) or the average-linkage (AL) and the spectral clustering algorithms [14], [15].

However, clustering is inherently an ill-posed problem. All the previous methods are designed with certain assumptions and favor some type of biases, and no single one is universally suitable for solving all the problems [19]. Hoping to exploit the strength of many individual clustering algorithms, people turn to

* This work was partially supported by the National Natural Science Foundation under Grant No. 60303014, the Fok Ying Tung Education Foundation under Grant No. 101068, the Specialized Research Found of Doctoral Program of Higher Education of China under Grant No. 20050004008, and the Foundation for the Authors of National Excellent Doctoral Dissertation, China, under Grant 200038.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 664–671, 2007.

clustering ensembles, seeking improvement over a single clustering algorithm in such aspects as *robustness, novelty* and *scalability, et al.*

Besides formal arguments on the effectiveness of cluster ensembles [18], many combining algorithms have been proposed, and their good performance further justified the use of cluster ensembles. A few examples are: methods based on hypergraph (CSPA, HGPA, and MCLA) [16] or bipartite graph partitioning [3], evidence accumulation using the CA matrix (EAC-SL and EAC-AL) [6], mixture model using a unified representation for multiple partitions [17], bagged clustering [11], and combination by plurality voting [1], [4], [7], [20].

Despite the primary success achieved by those algorithms, they are far from ideal. To make the clustering ensembles practical and helpful for us, an effective algorithm is crucial, which is the focus of this paper. We first analyze the CA matrix, and discuss the problem of designing a proper algorithm for it in Sect. 2. Based on a novel concept of *normalized edges* as we have defined in this paper, we propose a hierarchical algorithm to find the final partition in Sect. 3. In Sect. 4, experiment results demonstrate the effectiveness of our proposed method.

2 Analysis of the Co-Association Matrix

2.1 The Co-Association (CA) Matrix

In order to combine the multiple partitions of the data, one can first map the data to a new feature space as a way to accumulate the information provided by each partition. The CA matrix[1] C [6] is a newly constructed similarity matrix from multiple partitions of the original data. It takes the co-occurrence of pairs of patterns in the same clusters as votes for their association, with elements

$$C(i,j) = \frac{n_{ij}}{N}, \tag{1}$$

where n_{ij} is the number of times the pair x_i and x_j is assigned to the same cluster among the N partitions. In fact, the CA matrix records the frequency that every pair of points is in the same cluster.

The CA matrix ($N = 30$, and the number of clusters is fixed to 60[2]) of the *2-spirals* data[3] (Fig. 1a) is shown in Fig. 1c. For comparison, the similarity matrix (normalized into the 0-1 scale) based on the Euclidean distance for the original patterns is plotted in Fig. 1b. Evidently, the similarity of pairs of points from different clusters is mostly much smaller in the CA matrix than that in the original similarity matrix, showing the CA matrix captures the global structure of the data. Thus, it is not surprising that the evidence accumulation method operating on the CA matrix [6] performs well and that is why we use the CA matrix to combine the multiple partitions in our method.

[1] This matrix is also used in the CSPA algorithm [16]; and in [12] but by the name of *consensus matrix.*

[2] This strategy, initially splitting the data into a large number of small clusters and then combining them, is the so-called *split-and-merge* approach [5].

[3] To make comparison easy, we reorder the data, so the first 100 points are from one cluster and the last 100 points from the other cluster.

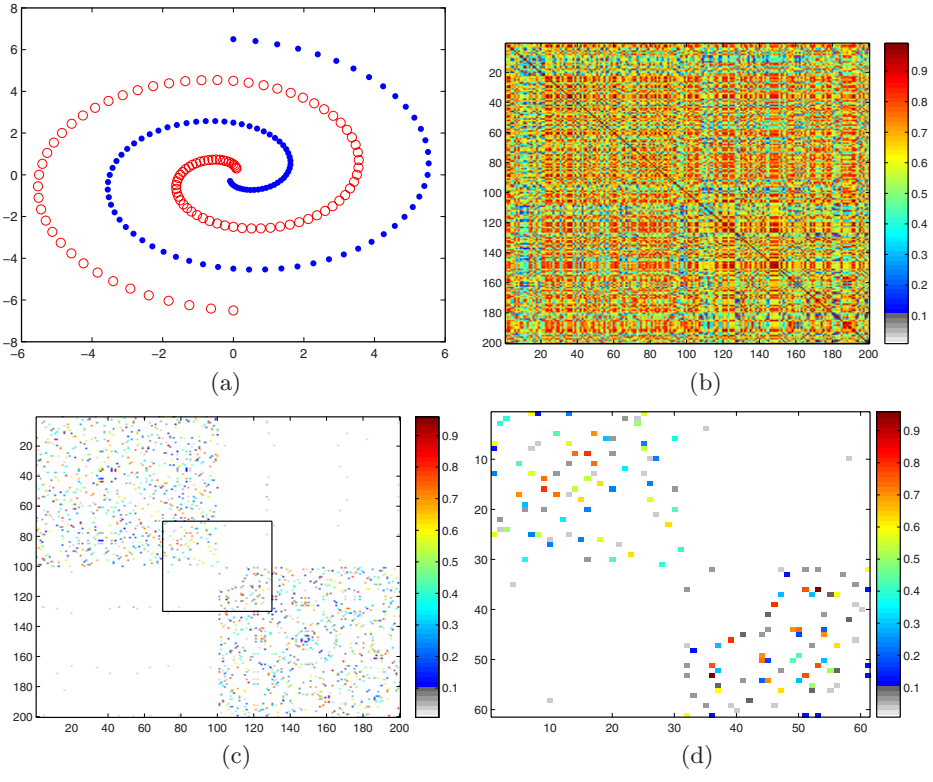


Fig. 1. The *2-spirals* data set and its similarity matrices. (a) *2-spirals* data. (b) Similarity matrix using Euclidean distance. (c) CA matrix: new similarity matrix using multiple partitions. (d) Enlarged part of (c).

2.2 A Proper Algorithm

Despite the good discrimination ability of the CA matrix, improper clustering algorithms can still lead to bad results. For example, in [6] (e.g. table 2) and [17] (e.g. Fig. 9 and 10b), the authors found that, based on the CA values, the results of the consensus functions (i.e., SL, AL and CL) differ significantly, and the choice of a good consensus function is sensitive to the choice of the data set. So, does there exist a better consensus function? How can we find it?

Compared with ordinary similarity matrix using Euclidean distance, the CA matrix has special characteristics. Without loss of generality, we take the CA matrix (Fig. 1c) of the *2-spirals* data as an example, part of which is highlighted in Fig. 1d. The similarity matrix of the data using Euclidean distance is shown in Fig. 1b. To construct a good algorithm, we believe that the following characteristics should be taken into account: 1) points from different clusters are always dissimilar, 2) a large percentage of pairs of points from the same cluster have very low similarity, and 3) if two points from the same cluster are dissimilar, then there always should be a path of some points (or just one point) between them

who are successively similar. Referring to these features, we can also explore the reasons why the SL, AL and CL algorithms would have such performance in [6].

3 Normalized Edges and the Algorithm

Based on the above analysis, we will customize a hierarchical clustering algorithm to operate on the CA matrix for combining multiple partitions, in hopes of discovering the true structure of the data more successfully and robustly.

3.1 Normalized Edges

Our proposed hierarchical clustering algorithm is based on a novel concept of *normalized edges* for measuring the similarity between clusters. Treating all points of the data as a set of vertices, we can define an undirected and unweighted graph. An edge exists between two points (or vertices), x_i and x_j, if and only if their similarity is larger than a threshold θ. In fact, this defines a *threshold graph*. For simplicity, we define a function *edge* between two points x_i and x_j,

$$edge(x_i, x_j) = \begin{cases} 1 \text{ if } \mathrm{sim}(x_i, x_j) > \theta, \\ 0 \text{ otherwise.} \end{cases} \tag{2}$$

The notion of *edges* between two clusters C_i and C_j, $edges(C_i, C_j)$, is just the number of distinct edges connecting these two groups. Essentially, this can be used as a measure of the *goodness* of merging them in an agglomerative hierarchical clustering algorithm. However, this naive approach may work well only for well-separated and approximately equal-sized clusters.

A proper way to fix this problem is to normalize the number of edges between two clusters $edges(C_i, C_j)$, by dividing it by the (estimated) expected number of edges between them, which is inspired by *goodness measure* used in ROCK [8]. Hence, the number of *normalized edges (NE)* between two clusters, C_i and C_j, is

$$NE(C_i, C_j) = \frac{edges(C_i, C_j)}{(n_i + n_j)^{1+f(\theta)} - n_i^{1+f(\theta)} - n_j^{1+f(\theta)}}, \tag{3}$$

where n_i and n_j are the number of points, $n_i^{1+f(\theta)}$ and $n_j^{1+f(\theta)}$ are the estimated expected number of edges, in the clusters C_i and C_j respectively.

As in [8], we assume that every point in C_i has $n_i^{f(\theta)}$ edges with other points in the cluster, then the total number of edges between points in the cluster is $n_i^{1+f(\theta)}$ (each edge is counted twice). Thus the expected number of edges between pairs of points (each point from a different cluster) becomes $(n_i + n_j)^{1+f(\theta)} - n_i^{1+f(\theta)} - n_j^{1+f(\theta)}$. Intuitively, the function $f(\theta)$ is introduced to measure the influence of θ on the number of edges. Based on the analysis of Guha et al. [8], it is also defined as $(1-\theta)/(1+\theta)$ in this paper.

3.2 Our Clustering Algorithm

With the definition of *normalized edges* to measure the similarity between two clusters, we can use this measure to construct a new agglomerative hierarchical algorithm. To reduce the workload of calculation, we need not re-start calculating the similarity between clusters after each merging step. We just have to update the similarity between the merged and the other clusters. That is, if clusters C_i and C_j are merged into a new cluster C_k, then we have (for $l \neq i, j$ and k) $edges(C_l, C_k) = edges(C_l, C_i) + edges(C_l, C_j)$, and $n_k = n_i + n_j$.

Thus we can calculate the *normalized edges* $NE(C_l, C_k)$ by definition.

The proposed algorithm can be summarized as follows:

The Algorithm

Input:
The similarity matrix (the CA matrix in this paper), the threshold θ, and the number of clusters k.
Initialization:
To calculate *edge* between all pairs of points based on the similarity matrix.
Repeat:
1. To merge two clusters, among all possible pairs of clusters, with the largest *normalized edges*;
2. To update the *edges* and *normalized edges* between the merged clusters and the other clusters.
Until:
Only k clusters left or the number of edges between every pair of the remaining clusters becomes zero.

Certainly, we can speed up the algorithm by many methods used in the traditional agglomerative clustering algorithms. The *edge* between every pair of points can be computed in $O(n^2)$ time, and the worst time complexity of the clustering algorithm is $O(n^2 \log n)$, just as the ROCK algorithm.

For the threshold θ in our algorithm, we are still seeking a general way to determine it. Empirically, the algorithm worked well with θ in the interval [0.1,0.4] for many data sets, and was not very sensitive to it (e.g., for all the data sets in the experiments of this paper, we fixed θ to 0.30).

4 Experiments

We compared our algorithm with single runs of some well-known clustering algorithms and other ensemble ones, and the experiment results demonstrate the effectiveness of our method.

4.1 Data Sets, Algorithms and Parameters Selection

We summarized the details of the data sets in Table 1, which had been adopted by other authors to test their ensemble algorithms [4], [6], [13], [17].

Table 1. Characteristics of the Data Sets

Data set	No. of features	No. of classes	No. of points/class(noise)	Total no. of points
2-spirals	2	2	100-100	200
Complex image	2	7	200-200-100-100-50-50-33-(10)	743
3-paths	2	3	200-200-200-(200)	800
Wisconsin Breast-cancer	9	2	239-444	683
Std Yeast	17	5	67-135-75-52-55	384

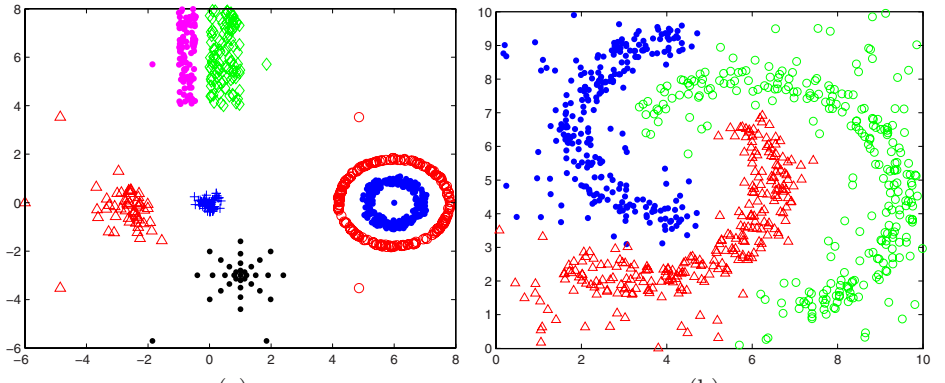


Fig. 2. The *complex image* and *3-paths* data sets. Clusters are indicated by different colors/patterns. Here show the clustering results of our ensemble algorithm. (a) *Complex image*, the 10 points of the outer circle are considered as noise. (b) *3-paths*, 3 path-like clusters (200 points for each), corrupted by 200 noise points.

We compared the experiment results of our ensemble algorithm (denoted by CA-HNE) with single runs of following algorithms: KM, SL, AL, and spectral clustering algorithm (SC) [14], and following ensemble methods: CSPA, HGPA and MCLA [16], Boost-KM [7], EAC-SL and EAC-AL [6], and Latent-EM [17].

For the algorithms proposed by other authors, the parameters selection and other settings are the same as suggested in their papers. The multiple partitions of the data were obtained by running KM algorithm with random initialization of cluster centers.

We found that, using only 10 or 20 component partitions, our algorithm worked well for many data sets, while many other authors used a (much) larger number for their algorithm, e.g. 50 [6], 100 [4], 500 [12]. For simplicity, we generated 30 partitions for our algorithm and 50 for all other ensemble ones. For our algorithm, k was chosen as a constant, usually larger than the true number of clusters of the data. The 'true' number of clusters of the data, was assumed to be known, as in [6], [17].

4.2 Results, Comparison and Analysis

Table 2 summarizes the mean error rates and standard deviations from 20 independent runs of the different methods on the data sets. The error rates are obtained by matching the clustering results with the ground-truth information, taken as the known labeling of the real-world data sets or the perceptual grouping of the artificial ones. Since all the clustering algorithms considered here do not detect outliers in the data, we ignore the noise points when calculating the error rates. Notice that the SL and AL algorithms give unvaried clustering results for each data set.

Table 2. Mean Error Rates and Standard Deviations of Different Algorithms

Data set	KM	SL	AL	SC	CSPA	HGPA
2-spirals	.399±.011	0	.480	**0±0**	.418±.059	.394±.078
Complex image	.550±.073	.523	.478	.250±.074	.404±.076	.389±.048
3-paths	.327±.005	.667	.350	.046±.001	.212±.058	.251±.068
Breast-cancer	.039±.001	.349	.057	**.029±0**	.167±.020	.147±.017
Std Yeast	.358±.057	.638	.341	**.320±0**	.442±.011	.431±.014
Data set	MCLA	Boost -KM	EAC -SL	EAC -AL	Latent -EM	CA -HNE
2-spirals	.365±.075	.429±.007	**0±0**	.325±.051	.418±.030	**0±0**
Complex image	.397±.078	.533±.032	.136±.125	.546±.043	.540±.055	**.013±.023**
3-paths	.214±.138	.328±.000	.617±.122	.375±.046	.349±.049	**.006±.017**
Breast-cancer	.131±.030	.039±.001	.319±.093	.047±.007	.039±.001	.030±.004
Std Yeast	.422±.022	.332±.021	.594±.062	.349±.023	.390±.072	.333±.030

We can see that the evidence accumulation clustering (EAC-SL or EAC-AL) can sometimes discover the structure of the data successfully. However, which method will succeed depends heavily on the choice of the data sets. In general, the AL (SL) consensus function based on CA matrix is appropriate if standard AL (SL) agglomerative clustering method works well for the data, and vice versa [17]. This may be problematic since sometimes the characteristic of the data is difficult to know, or is complex for standard agglomerative clustering (e.g. *3-paths*). However, our algorithm tackles this problem well. For our method, the mean error rates presented in Table 2 are all the best or comparable to the best, and the partitions of the *complex image* and *3-paths* data sets are as good as we expected, see Fig. 2a and 2b. The experiments show the effectiveness of our algorithm: it gives the best (or comparable to the best) overall performance for all the data sets. It clusters all the chosen data sets reasonably, though they have hybrid or complex characteristic or are corrupted by noise. The CSPA algorithm does not perform well for these chosen data sets, though it is also based on the CA matrix. Again, this demonstrates the importance of the choice of algorithms for the final partition based on the CA matrix. Other ensemble methods HGPA, MCLA, Boost-KM, latent-EM still favor some type of biases, and do not perform well for other kinds of data sets.

Single runs of KM, SL, and AL algorithms result in good partitions when the data sets are suitable for them, but fail drastically otherwise (e.g. noise, hybrid structure). The SC algorithm gives reasonable partitions for some of the data sets, but fails for *complex image* and *3-paths*.

References

1. Dudoit, S., Fridlyand, J.: Bagging to Improve the Accuracy of a clustering procedure. Bioinformatics, 19 (2003) 1090–1099
2. Everitt, B.S., Landau, S., Leese, M.: Cluster Analysis (4e). Hodder Arnold (2001)
3. Fern, X., Brodley, C.: Solving cluster ensemble problems by bipartite graph partitioning. Proc. 21st Int'l Conf. Machine Learning (2004) 281–288
4. Fischer, B., Buhmann, J.M.: Bagging for Path-Based Clustering. IEEE Trans. Patt. Anal. Machine Intell., 25(11) (2003) 1411–1415
5. Fred, A.L.N., Jain, A.K.: Data Clustering Using Evidence Accumulation. Proc. 16th Int'l Conf. Pattern Recognition, 280 (2002) 276–280
6. Fred, A.L.N., Jain, A.K.: Combining Multiple Clusterings Using Evidence Accumulation. IEEE Trans. Patt. Anal. Machine Intell., 27(6) (2005) 835–850
7. Frossyniotis, D., Likas, A., Stafylopatis, A.: A Clustering Method Based on Boosting. Pattern Recognition Letters, 25 (2004) 641–654
8. Guha, S., Rastogi, R., Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. Information systems, 25(5) (2000) 345–366
9. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
10. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons (1990)
11. Leisch, F.: Bagged Clustering. Working Papers SFB Adaptive Information Systems and Modeling in Economics and Management Science, Institut fur Information, Abt. Produktionsmanagement, Wien, Wirtschaftsuniv, 51 (1999)
12. Monti, S., Tamayo, P., Mesirov, J.P., Golub, T.R.: Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. Machine Learning, 52(1-2) (2003) 91–118
13. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases, Univ. of California, Dept. of Info. and Computer Science (1998)
14. Ng, A.Y., Jordan, M.I., Weiss, Y.: On Spectral Clustering: Analysis and an Algorithm. Advances in Neural Information Processing Systems, 14 (2002)
15. Shi, J., Malik, J.: Normalized Cuts and Image Segmentation. IEEE Trans. Patt. Anal. Machine Intell., 22(8) (2000) 888–905
16. Strehl, A., Ghosh, J.: Cluster Ensembles — A Knowledge Reuse Framework for Combining Partitionings. J. Machine Learning Research, 3 (2002) 583–617
17. Topchy, A., Jain, A.K., Punch, W.: Clustering Ensembles: Models of Consensus and Weak Partitions. IEEE Trans. PAMI., 27(12) (2005) 1866–1881
18. Topchy, A.P., Law, M.H.C., Jain, A.K., Fred, A.L.: Analysis of Consensus Partition in Cluster Ensemble. Proc. 4th IEEE Int'l Conf. Data Mining (2004) 225–232
19. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Trans. Neural Networks, 16(3) (2005) 545–678
20. Zhou, Z.H., Tang, W.: Clusterer Ensemble. Knowledge-Based Systems, 19(1) (2006) 77–83

Quantum-Inspired Immune Clonal Multiobjective Optimization Algorithm

Yangyang Li and Licheng Jiao

Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China
lyy_791@yahoo.com.cn

Abstract. Based on the concept and principles of quantum computing, a quantum-inspired immune clonal multiobjective optimization algorithm (QICMOA) is proposed to solve extended 0/1 knapsack problems. In QICMOA, we select less-crowded Pareto-optimal individuals to perform cloning, recombination update. Meanwhile, the Pareto-optimal individual is proliferated and divided into a set of subpopulation groups. Individual in a subpopulation group is represented by multi-state gene quantum bits. For the novel representation, qubit individuals in subpopulation are updated by applying a new chaos update strategy. The proposed recombination realizes the information communication among individuals so as to improve the search efficiency. We compare QICMOA with SPEA, NSGA, VEGA and NPGA in solving nine 0/1 knapsack problems. The statistical results show that QICMOA has a good performance in converging to true Pareto-optimal fronts with a good distribution.

1 Introduction

The multiobjective optimization problems have attracted more attentions from researchers in various fields. Evolutionary algorithms (EAs) have been recognized to be well suited to multiobjective optimization since early in their development because they deal simultaneously with a set of possible solutions. The ability to handle complex problems, involving features such as discontinuities, multimodality, disjoint feasible spaces and noisy function evaluations, reinforces the potential effectiveness of EAs in multiobjective optimization, which is perhaps a problem area where Evolutionary Computation really distinguishes itself from its competitors [1].

Artificial Immune System (AIS) is a new hotspot following the neural network, fuzzy logic and evolutionary computation [2]. Its research production refers to many fields like control, data processing, optimization learning and trouble diagnosing. Based on immunological principles, new computational techniques are being developed, aiming not only at a better understanding of the system, but also at solving engineering problems [3]-[6].

Unlike other research areas, there has been relatively little work done in applying quantum computing to AIS. In [7], quantum-inspired computing was proposed. A quantum-inspired immune clonal algorithm (QICA) was firstly introduced in [3] for solving the high dimensional function optimization problems. It should be noted that although QICA is based on the concept of quantum computing, QICA is not a quantum algorithm, but a novel optimization algorithm for a classical computer. In this

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 672–679, 2007.

paper, we propose a novel multiobjective algorithm, called a quantum-inspired immune clonal multiobjective optimization algorithm (QICMOA). In QICMOA, individuals (antibodies) in a population are represented by quantum bits (qubits) [3]; the fitness value of each Pareto-optimal individual is assigned as the average distance of two Pareto-optimal individuals on either side of this individual along each of the objectives, which is called crowding-distance and proposed in NSGA-II [8]. According to the fitness values, only less-crowded Pareto-optimal individuals are selected to do cloning, recombination update. So in a single generation, QICMOA pays more attention to the less-crowded regions in the trade-off front.

2 The Multiobjective 0/1 Knapsack Problems

A 0/1 knapsack problem is a typical combinatorial optimization problem, yet the problem itself is difficult to solve (NP-hard). This single-objective problem can be extended directly to the multiobjective case by allowing an arbitrary number of knapsacks. Multiobjective 0/1 Knapsack Problems with n knapsacks (i.e. n objectives and n constraints) and m items in [10] can be defined as follows:

$$\begin{aligned} &\text{Maximize} \quad F(x) = (f_1(x), f_2(x), ..., f_n(x)) \\ &\text{subject to} \quad \sum_{j=1}^{m} w_{ij} x_j \le c_i \quad i = 1, 2, ..., n \end{aligned} \tag{1}$$

Where $f_i(x) = \sum_{j=1}^{m} p_{ij} x_j \quad i = 1, 2, ..., n$ and $x_j = 1 (j = 1, ..., m)$ if and only if item j is selected. In the above problem (1), $x = (x_1, x_2, ..., x_m) \in \{0,1\}^m$ is an $m-$dimensional binary vector; p_{ij} =profit of item j according to knapsack i; w_{ij} =weight of item j according to knapsack i; c_i =capacity of knapsack i.

Therefore, the optimization goal of the multiobjective 0/1 knapsack problem is to find a set of Pareto-optimal solutions approximating the true Pareto-optimal front.

3 QICMOA

3.1 Representation

In this study, an antibody represents a search point in decision space. In [3], we give a new representation, a qubit for the probabilistic representation that it can represent a linear superposition of states and has a better characteristic of population diversity than other representations [3], which are defined below.

Definition 1. The probability amplitude of one qubit is defined with a pair of numbers (α, β) as

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{2}$$

Satisfying

$$|\alpha|^2 + |\beta|^2 = 1 \tag{3}$$

Where $|\alpha|^2$ gives the probability that the qubit will be found in the '0' state and $|\beta|^2$ gives the probability that the qubit will be found in the '1' state.

Definition 2. A qubit antibody as a string of *m* qubits is defined as:

$$\begin{Bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_m \\ \beta_1 & \beta_2 & \dots & \beta_m \end{Bmatrix} \tag{4}$$

where $|\alpha_l|^2 + |\beta_l|^2 = 1, (l = 1, 2, ..., m)$.

3.2 Description of the Algorithm

In this section, we describe a novel multiobjective optimization algorithm, termed as quantum-inspired immune clonal multiobjective optimization algorithm (QICMOA). The main loop of QICMOA is as follows.

Algorithm 1: quantum-inspired immune clonal multiobjective optimization algorithm

Input: *Gmax* (maximum number of generations)
N_D (maximum size of Dominant Population)
N_A (maximum size of Active Population)
N_C (size of Clone Population)
Output: $\boldsymbol{D}_{\boldsymbol{Gmax+1}}$ (final Pareto-optimal set)

Step1: ***Initialization:*** Generate an initial qubit antibody population $\boldsymbol{QB}_0$ with the size N_D. Set t=0.

Step2: ***Observing Operator:*** Generate classical antibody population $\boldsymbol{B}_t$ by observing the states of $\boldsymbol{QB}_t$.

Step3: ***Update Dominant Population:*** Identify dominant antibodies in $\boldsymbol{B}_t$; Copy all the dominant antibodies to form the temporary *dominant population* $\boldsymbol{DT}_{t+1}$; If the size of $\boldsymbol{DT}_{t+1}$ is no larger than N_D , let $\boldsymbol{D}_{t+1}$=$\boldsymbol{DT}_{t+1}$. Otherwise, calculate the crowding-distance values of all individuals in $\boldsymbol{DT}_{t+1}$, sort them in descending order of crowding-distance, choose the first N_D individuals to form $\boldsymbol{D}_{t+1}$. Modify $\boldsymbol{D}_{t+1}$ with the greedy repair method as reported in reference [9]. Record the corresponding qubit of $\boldsymbol{D}_{t+1}$ to get qubit dominant population $\boldsymbol{QD}_{t+1}$.

Step4: ***Termination:*** If $t \geq Gmax$ is satisfied, export $\boldsymbol{D}_{t+1}$ as the output of the algorithm, Stop; Otherwise, t=t+1.

Step5: ***Generate active antibodies:*** If the size of $\boldsymbol{D}_t$ is no larger than N_A, let $\boldsymbol{A}_t$=$\boldsymbol{D}_t$. Otherwise, calculate

the crowding-distance values of all individuals in $\boldsymbol{D_t}$, sort them in descending order of crowding-distance, choose the first N_A individuals to form $\boldsymbol{A_t}$.

Record the corresponding qubit of $\boldsymbol{A_t}$ to get qubit active population $\boldsymbol{QA_t}$.

Step6: ***Cloning Operator:*** Get the qubit clone population $\boldsymbol{QC_t}$ by applying the cloning $\boldsymbol{T^C}$ to $\boldsymbol{QA_t}$.

Step7: ***Recombination update:*** Perform recombination update on $\boldsymbol{QC_t}$ and set $\boldsymbol{QC_t'}$ to the resulting population.

Step8: Get the qubit antibody population $\boldsymbol{QB_t}$ by combining the $\boldsymbol{QC_t'}$ and $\boldsymbol{QD_t}$; go to Step2.

The major elements of QICMOA are presented as follows.

1. Observing Operator

At Step2, in the act of observing a quantum state, it collapses to a single state (namely classical representation). The process is described as follows: (a) generate a random number $p \in [0,1]$; (b) if it is larger than $\left|\alpha_l^t\right|^2$, the corresponding bit in $P'(t)$ takes '1', otherwise it takes '0'.

2. Cloning Operator

In immunology, Cloning means asexual propagation so that a group of identical cells can be descended from a single common ancestor, such as a bacterial colony whose members arise from a single original cell as the result of mitosis. In this study, the cloning operator T^C on qubit active population QA is defined as:

$$T^C(QA) = \{T^C(qa_1), T^C(qa_2), ..., T^C(qa_{|QA|})\} \tag{5}$$

Where $T^C(qa_i) = I_i \times qa_i, i = 1,2,...,|QA|$, and I_i is C_i dimension row vectors. C_i is a self-adaptive parameter, and $\sum_{i=1}^{|QA|} C_i = N_C$, N_C is a given value of the size of the clone population. Then the values of C_i are calculated as

$$C_i = \left\lceil N_c * \frac{i_{\text{distance}}}{\sum_{i=1}^{N_A} i_{\text{distance}}} \right\rceil, \tag{6}$$

Where i_{distance} denotes the crowding-distance values of the i-th active antibodies. After clone, the population becomes:

$$QC = \{\{qc_1^1, qc_1^2, ..., qc_1^{C_1}\}, \{qc_2^1, qc_2^2, ..., qc_2^{C_2}\}, ..., \{qc_{|QA|}^1, qc_{|QA|}^2, ..., qc_{|QA|}^{C_{|QA|}}\}\} \tag{7}$$

Where $qc_i^j = qa_i$, $j = 1,2,...,C_i$. In fact, cloning on antibody qa_i is to make multiple identical copies of qa_i. The aim is that the larger the crowding-distance value of an individual, the more times the individual will be reproduced. So there exist more chances to do search in less-crowded regions of the trade-off fronts.

3. Recombination Update

In this paper, we use a new qubit representation, and perform cloning operator, recombination update on corresponding qubit antibody population so as to improve the search efficiency.

If $QC = \{qc_1, qc_2, ..., qc_{N_C}\}$ is the resulting qubit population from applying the cloning to $QA = \{qa_1, qa_2, ..., qa_{|QA|}\}$, then the recombination update T^R on clone population QC is defined as

$$\begin{aligned} T^C(QC) &= \{T^R(qc_1), T^R(qc_2), ..., T^R(qc_{N_C})\} \\ &= \{recom(qc_1, A), recom(qc_2, A), ..., recom(qc_{N_C}, A)\} \end{aligned} \tag{8}$$

Where $recom(qc_i, A)$ denotes selecting one individual from the offspring generated by a chaos update strategy on clone qc_i and an active antibody selected randomly from A.

For the chaos character of qubit [10], we propose a new chaos update strategy on qubit clone antibody population QC (see equation 7) as follows.

$$\begin{aligned} q_{guide} &= \omega \times a_i + (1-\omega) \times (1-a_i) \\ qc_i^j &= q_{guide} + \upsilon \times Logistic(j), \quad (i = 1,2,...,|QA|, j = 1,2,...,C_i) \end{aligned} \tag{9}$$

Where a_i, q_{guide} and $Logistic(j)$ are the i-th classical antibody in A_t, the guide qubit antibody and j-th value of Logistic sequence [11] whose length is C_i, respectively. qc_i^j is the j-th updated qubit antibody in i-th subpopulation. ω is the guide factor of q_{guide} and υ is the spread variance. Often we let $\omega \in [0.1, 0.5]$, $\upsilon \in [0.05, 0.15]$.

3.3 Computational Complexity

In this section, we only consider population size in computational complexity. Assuming that the maximum size of dominant population is N_D, the maximum size of active population is N_A, the clone population size is N_C, and then the time complexity of one generation for the algorithm can be calculated as follows:

The time complexity for observing operation is $O(N_D + N_C)$; the time complexity for identifying Pareto-optimal solutions in population is $O((N_D + N_C)^2)$; the worst time complexity for update the dominant population is $O((N_D + N_C)\log(N_D + N_C))$;

the worst time complexity for generate active antibodies is $O(N_D \log(N_D))$; the time complexity for cloning is $O(N_C)$ and the time complexity for recombination operation is $O(N_C)$. So the worst total time complexity is

$$O(N_D + N_C) + O((N_D + N_C)^2) + O((N_D + N_C)\log(N_D + N_C)) + O(N_D \log(N_D)) + O(N_C) + O(N_C) \tag{10}$$

According to the operational rules of the symbol O , the worst time complexity of one generation for QICMOA can be simplified as

$$O((N_D + N_C)^2) \tag{11}$$

So the cost of identifying the Pareto optimal individuals in population dominates the computational complexity of QICMOA.

4 Performance Comparison

In this section, we compare SPEA [9], NSGA [12], VEGA [13] and NPGA [14] with QICMOA in solving nine multiobjective 0/1 knapsack problems [9].The test data sets are available from Zitzler's homepages (http://www.tik.ee.ethz.ch/~zitzler/testdata.html/), where two, three and four knapsacks with 250, 500, 750 items are taken under consideration.

In the following experiments, we performed 30 independent runs on each test problem. Table 1 shows the direct comparison of QICMOA with SPEA, NSGA, VEGA and NPGA based on the performance metrics - Coverage.

For 2 knapsacks with 250 items, the results of Coverage show that the solutions obtained by QICMOA in a certain extent weakly dominate the solutions obtained by SPEA and clearly weakly dominate the solutions obtained by NSGA, VEGA and NPGA, which indicates that the behavior of QICMOA is better than that of the other four algorithm for this test problem. Especially for 2 knapsacks with 500 items and with 750 items, all the other four algorithms' solutions are clearly weakly dominated by QICMOA's ones over 30 independent runs. For 3 knapsacks with 250 items, the results show that solutions obtained by QICMOA in a certain extent weakly dominate the solutions obtained by SPEA and clearly weakly dominate the solutions obtained by NSGA, VEGA and NPGA. For 3 knapsacks with 500 items and 750 items, the results show that solutions obtained by QICMOA in a certain extent weakly dominate the solutions obtained by the other four algorithms where for 3 knapsacks with 500 items all the NPGA's and VEGA's solutions are clearly weakly dominated by QICMOA's ones over 30 independent runs. For 4 knapsacks with 250 items, the minimum, the mean and the maximum of $I(Q,S)$ are all smaller than that of $I(S,Q)$, which indicates in a sense SPEA performs better than QICMOA for this test problems. For 4 knapsacks with 500 items and 750 items, the results show that solutions obtained by QICMOA also in a certain extent weakly dominate the solutions obtained by SPEA NSGA and weakly dominate the solutions obtained by VEGA and NPGA.

Table 1. The Coverage of Two Sets for the Nine 0/1 Knapsack problems

	Coverage	$I(Q,S)$	$I(S,Q)$	$I(Q,NS)$	$I(NS,Q)$	$I(Q,V)$	$I(V,Q)$	$I(Q,NP)$	$I(NP,Q)$
2-250	Min	0.64707	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
	Mean	0.98634	0.04965	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
	Max	1.00000	0.10588	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
2-500	Min	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
	Mean	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
	Max	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
2-750	Min	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
	Mean	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
	Max	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
3-250	Min	0.8611	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
	Mean	0.9866	0.10733	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
	Max	1.0000	0.58000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
3-500	Min	0.3342	0.03470	0.98947	0.00000	1.00000	0.00000	1.00000	0.00000
	Mean	0.56271	0.24133	0.99965	0.00000	1.00000	0.00000	1.00000	0.00000
	Max	0.83541	0.32000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
3-750	Min	0.57674	0.00000	0.69795	0.00000	0.89703	0.00000	0.95070	0.00000
	Mean	0.78607	0.10124	0.84577	0.00145	0.97861	0.01596	0.98741	0.00124
	Max	0.90785	0.24158	0.96831	0.08000	0.99682	0.07250	0.99527	0.00278
4-250	Min	0.1101	0.13333	0.66515	0.00000	0.74903	0.00000	0.80731	0.00000
	Mean	0.32948	0.50808	0.85047	0.02866	0.93261	0.00033	0.92765	0.02000
	Max	0.55200	0.69697	0.97831	0.28000	0.99682	0.01000	0.99701	0.03120
4-500	Min	0.25324	0.12698	0.94851	0.00000	0.97849	0.00000	0.98893	0.00000
	Mean	0.59467	0.29629	0.98606	0.00000	0.99871	0.00000	0.99939	0.00000
	Max	0.80176	0.49206	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000
4-750	Min	0.58624	0.03500	0.99435	0.00000	1.00000	0.00000	0.99725	0.00000
	Mean	0.78526	0.12866	0.99965	0.00000	1.00000	0.00000	0.99991	0.00000
	Max	0.92958	0.38000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000

5 Conclusion

In this paper, we have introduced a new multiobjective optimization algorithm- quantum-inspired immune clonal multiobjective optimization algorithm (QICMOA) to solve 0/1 knapsack problems. When compared with SPEA, NSGA, VEGA and NPGA, QICMOA is more effective for multiobjective optimization problems in the popular metrics-Coverage.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 60372045), the Graduate Innovation Fund of Xidian University (No. 05004) and the National Basic Research Program of China (No. 2001CB309403).

References

1. Fonseca, C. M., Fleming, P. J.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation, 3(1), pp.1-16, 1995.
2. Dasgupta, D., Forrest, S.: Artificial immune systems in industrial applications. IPMM'99. Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials. IEEE press (1999) 257-267.
3. Li, Y.Y., Jiao, L.C.: Quantum-inspired Immune Clonal Algorithm. ICARIS'05. Proceedings of the 4th International Conference on Artificial Immune Systems, Banff, Alberta, Canada, Aug. 2005, pp.304 – 317.
4. De Castro, L. N., Von Zuben, F. J.: Learning and Optimization Using the Clonal Selection Principle. IEEE Trans. Evol. Comput., vol. 6, pp.239–251, Jun. 2002.
5. Jiao, L.C., Wang, L.: A novel genetic algorithm based on immune. IEEE Trans. Syst., Man, Cybern. A, vol. 30, pp. 1–10, Sept. 2000.
6. Gong, M.G., Du, H.F., Jiao, L.C.: Optimal approximation of linear systems by artificial immune response. Science in China: Series F Information Sciences. Science in China Press, co-published with Springer-Verlag GmbH. vol. 49, no.1. 2006, pp. 63–79.
7. Moore, M., Narayanan, A.: Quantum-Inspired Computing. Dept. Comput. Sci., Univ. Exeter, U.K., 1995.
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, 2002.
9. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. Evolutionary Computation. vol. 3, no. 4, pp.257–271 Nov. 1999.
10. Jordan, A. N.: Topics in Quantum Chaos. PHD paper, university of California Santa Barbara. 2002.
11. Li, Y.Y., Jiao, L.C., Liu, F.: Self-adaptive chaos quantum clonal evolutionary programming. ICSP '04. Proceedings of 7th International Conference on Signal Processing, Beijing, China, Sept. 2004, pp.1550 – 1553.
12. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput., vol. 2, no. 3, pp. 221–248, 1994.
13. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: the Proceedings of the International Conference on Genetic Algorithms and Their Applications. Pittsburgh, PA, 1985, pp.93–100.
14. Horn, J., Nafpliotis, N., Goldberg, D. E.: A niched pareto genetic algorithm for multiobjective optimization. in Proc. 1st IEEE Conf Evolutionary Computation, IEEE World Congr. Computational Computation, Piscataway, NJ, vol. 1, Jun. 27–29, 1994, pp. 82–87.

Phase Space Reconstruction Based Classification of Power Disturbances Using Support Vector Machines

Zhiyong Li and Weilin Wu

College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China
eewuwl@zju.edu.cn

Abstract. Using Phase Space Reconstruction (PSR) and Support Vector Machines (SVMs), a novel approach for power disturbance classification is presented. The types of concerned disturbances include voltage sags, voltage swells, voltage interruptions, impulsive transients, harmonics and flickers. PSR is applied for disturbance feature extraction. Based on PSR, power disturbance trajectories are constructed and then converted into binary images through encoding. Four distinctive features are proposed as the inputs of SVM classifier. Simulation results show that the classification method is effective and it requires less training samples.

Keywords: Power quality, Disturbance classification, Phase space reconstruction, Support vector machines.

1 Introduction

Power quality (PQ) is an increasingly important issue to electricity consumers along with the proliferation of highly sensitive computerized equipments and deregulation of the electric power industry. But unfortunately, the contamination of electromagnetic environment is getting worse because extensive applied power electronic technologies lead to a wide diffusion of nonlinear, time-variant loads in power distribution network.

Poor quality is attributed to the various power disturbances such as voltage sags, swells, interruptions, impulsive transients, harmonics, and voltage flickers, etc [1]. PQ disturbance detection and classification are therefore necessary in identifying the causes and sources of such disturbances so that their effects can be neutralized using suitable corrective and preventive measures. Great efforts have been made on the disturbance signals processing and subsequent decision making algorithms to realize disturbance automatic detection and classification [2-9].

Disturbance classification algorithm is always composed by two sequential processes: feature extraction and classification. In feature extraction stage, Wavelet Transform (WT) is widely used. Based on WT, different features like number of peaks of the wavelet coefficients [2], energy information at each decomposition level [3, 4], and energy difference between the distorted signal and the pure one [5] were chosen to construct the feature vectors for subsequent training and testing. Besides WT, Fourier transform [2], S-transform [6], Walsh transform [7] have been proved feasible in disturbance feature extraction. For classifier design, fuzzy-expert system [2, 3],

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 680–687, 2007.

dynamic time warping [7], wavelet neural network [8], and self organizing map [9] have been studied.

In this paper, we propose another novel method. Power disturbance signals are always measured by digital meters and saved as time series. So we apply a time series analysis tool (namely PSR) and define several indices to represent features of different disturbance patterns. SVM has received a great deal of attention recently proving itself a very effective approach in a variety of pattern classification tasks. It is adopted here to realize the automatic classification of PQ disturbance events.

The paper is organized as follows. Section 2 introduces the basic concept of PSR and illustrates how to extract disturbance features using PSR. Section 3 presents the concept and scheme of SVM classifier. Section 4 shows the simulation results using the method proposed, and finally, section 5 draws the conclusion.

2 Feature Extraction Based on Phase Space Reconstruction

2.1 Phase Space Reconstruction

PSR [10] was first utilized to reconstruct the motion on strange attractors in chaotic systems. Borrowing the idea of constructing signal trajectories from time series, Ref. [11] introduced PSR into PQ research field.

The basic concept of PSR is to convert a scalar sequence of measurements into state vectors. The values of variables at a certain moment and those values after τ, 2τ..., $(m-1)\tau$ time intervals are treated as coordinates of a special point in m-dimension phase space. Thus, for a single variable sequence $x_1, x_2..., x_N$, a delay reconstruction in m dimensions can be formed by the vector $\mathbf{X_i}$, given as:

$$\mathbf{X_i} = [x_i, x_{i+\tau}..., x_{i+(m-2)\tau}, x_{i+(m-1)\tau}] \ . \tag{1}$$

where i = 1, 2…, L, and L = 1, 2…, N-(m-1) τ.

From equation (1), we get the value matrix that carries the coordinates of points forming the trajectory:

$$\begin{aligned} \mathbf{X_1} &= [x_1, x_{1+\tau}, \ldots, x_{1+(m-1)\tau}] \\ \mathbf{X_2} &= [x_2, x_{2+\tau}, \ldots x_{2+(m-1)\tau}] \\ &\vdots \\ \mathbf{X_L} &= [x_L, x_{L+\tau}, \ldots x_{L+(m-1)\tau}] \end{aligned} \ . \tag{2}$$

Take the sinusoid sequence as an example. From a normalized sine waveform with *50Hz* power frequency (shown in Fig. 1(a)), which is sampled at *4800Hz*, we obtain time series of *96* points for every period. We construct the trajectory in 2-D phase plane (m=2, τ=20). The values of x_i and x_{i+20} indicate the position of the i-th point in trajectory. Therefore, a sequence sampled from sine wave in a period can be mapped into 96 points which compose a sinusoid trajectory shown in Fig. 1(b).

Similarly, Fig.2 shows several typical power disturbances and their corresponding PSR-based trajectories.

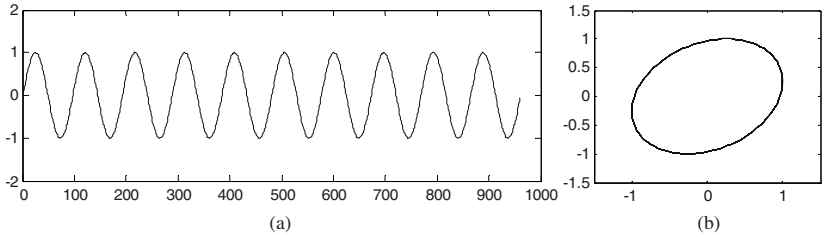


Fig. 1. A pure sine wave and its corresponding trajectory using PSR

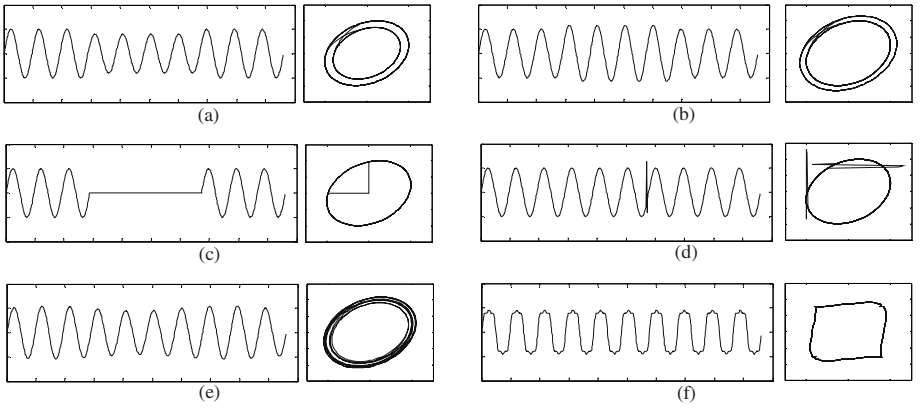


Fig. 2. Several typical power disturbance waveforms and their corresponding PSR-based trajectories. (a) voltage sag, (b) voltage swell, (c) voltage interruption, (d) impulsive transient, (e)harmonic, (f) flicker

2.2 Feature Extraction of Power Disturbances

2.2.1 Encoding

As shown in Fig. 1(b), under per-unit system, the trajectory of sine wave is restricted in the square [-1, 1] × [-1, 1]. Taking potential over voltage into account, we pick a larger area of [-1.5, 1.5] × [-1.5, 1.5] for further analysis. This area is divided into 300×300 segments. Each segment indicates a 2-value pixel (*black pixel*: value 1 and *white pixel*: value 0). The pixels are assigned value 1 if any point in trajectory falls into corresponding segment, the remnant pixels remain value 0. After such encoding process, the graph in Fig. 1(b) will be transformed into a binary image.

In such image, the power system voltage waveform can be regarded as a combination of a carrier (e.g. pure sine wave) and disturbance component imposed onto that waveform. The sinusoidal component is not the informative part in terms of an event detection or classification, but its existence perturbs some statistical parameters. For this purpose, we remove sinusoidal component (also called stable limit circle, as shown in Fig. 1(b)) during feature extraction procedure.

Fig.3 illustrates the processing procedure of a voltage sag waveform.

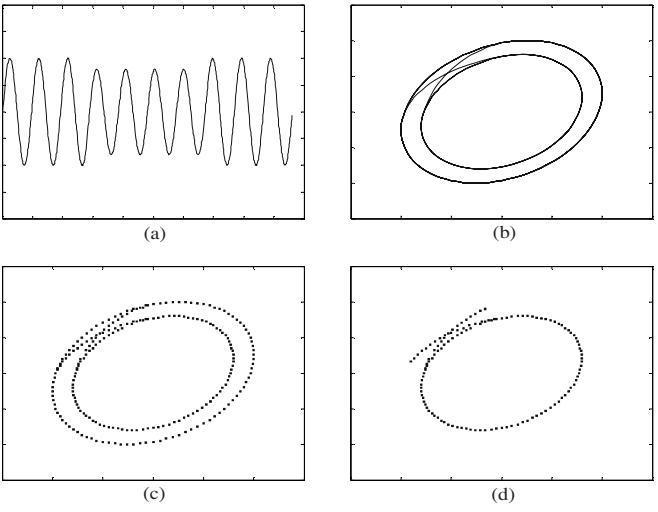


Fig. 3. (a) voltage sag waveform, (b) voltage sag trajectory based on PSR, (c) binary image of voltage sag trajectory, (d) binary image of trajectory after carrier component remove

2.2.2 Feature Definition

Having illustrated the image processing procedure, we further formulate how to extract distinctive parameters from characteristics carried by black pixels.

Definition 1. *Maximum Adjacent Distance (MAD)*

For each black pixel in binary image (as shown in Fig.3 (c) and (d)), we calculate minimum distance to its adjacent black pixel denoted by *AD*:

$$ADi = \min(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}) \ . \tag{3}$$

where (x_i, y_i) and (x_j, y_j) are coordinates of the i-th and j-th black pixel in binary image. $i, j = 1,2\ldots,N$ and $i \neq j$.

MAD is the maximum of *ADi* ($i, j = 1,2\ldots,N$):

$$MAD = \max(ADi) \ . \tag{4}$$

Definition 2. *Carrier Component Similarity (CCS)*

The binary image of disturbance trajectory is compared with stable limit cycle. A black pixel in disturbance trajectory is marked as sinusoidal pixel (SP) if a relevant point can be found in sine wave trajectory to satisfy:

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq \varepsilon \ . \tag{5}$$

where (x_i, y_i) are coordinates of the i-th black pixel in disturbance trajectory, and (x_j, y_j) indicate the relevant pixel in stable limit cycle. ε is a small value greater than zero, considering ubiquitous noise.

CCS is defined as the number of SP:

$$CCS = n(SP) \ . \tag{6}$$

Definition 3. *Overlay Area (OA)*
After removing previous SP, the rest black pixels describe how densely disturbance component occupy the binary image. Hence *OA* is represented by the number of remnant pixels (RP):

$$OA = n(RP) \ . \tag{7}$$

Definition 4. *Mean Amplitude (MA)*
The definition of *MA* is given as follows:

$$MA = \sum_{i=1}^{n(RP)} \sqrt{x_i^2 + y_i^2} \ / \ n(RP) \ . \tag{8}$$

where (x_i, y_i) are coordinates of the i-th black pixel.

These definitions represent characteristics of different disturbance patterns. Large *MAD* suggests potential impulse because the duration of impulse is short but the magnitude always large. The carriers of harmonic and flicker disturbance are distorted, which results in less *CCS*. In other words, *CCS* is feasible to separate long term disturbances such as harmonics or flickers from short term disturbances such as voltage sags, swells or interruptions.

Because of the oscillating envelope, each cycle of flicker waveform construct a different trajectory in phase plane, so its overlay area is much larger than other types. But the duration of impulsive transient is typically within a millisecond. This implies few black pixels apart from stable limit cycle in binary image, therefore the value of its *OA* tends to zero. *MA* indicates the magnitude of disturbance event, so it is effective to distinguish among voltage sags, swells and interruptions.

3 Support Vector Machines Classifier

3.1 Support Vector Machines

SVMs deliver state-of-the-art performance in real-world applications, and are now established as one of the standard tools for machine learning and data mining [12]. The basic objective is to find a hyperplane which best separate the positive/negative data in the feature space.

Consider a binary classification task with a set of linearly separable training samples $S = \{(\mathbf{x}_1, y_1)\ldots, (\mathbf{x}_n, y_n)\}$ where $\mathbf{x}$ is the input vector such that $\mathbf{x} \in R^d$ (in d-dimensional input space) and y_i is the class label such that $y_i \in \{-1, 1\}$. The goal of training is to create a suitable discriminating function:

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \ . \tag{9}$$

where $\mathbf{x}$ is input vector, $\mathbf{w}$ is weight vector which determines the orientation of the hyperplane $f(\mathbf{x})=0$, and b is the bias or offset.

For a linear SVM, the construction of discriminating function shown in (9) results in a convex optimization problem formed as:

$$\begin{aligned} &\min_{w,b} \ \frac{1}{2}\|\mathbf{w}\|^2 \\ &S.T. \quad y_i(\langle \mathbf{w}\cdot\mathbf{x}_i \rangle + b) \geq 1, i = 1,2,...,n \end{aligned} \quad . \tag{10}$$

The minimization of linear inequalities is typically solved by application of Lagrange duality theory. By introducing Lagrange multipliers αi, (10) can be converted into corresponding dual problem, and finally the classifier function becomes:

$$f(x) = \mathrm{sgn}[\sum_{i=1}^{n} \alpha_i^* y_i x_i \cdot x + b^*] \ . \tag{11}$$

3.2 Classifier Design

A single SVM is constructed to respond binary to the testing data. It has to be augmented with other strategies to achieve multicase classification. In this work, we adopted a one-against-all scheme by which each class can be ranked seriatim.

The sorting logic is shown in Fig.4. The text in square brackets indicates the candidates for analysis, and the features in parentheses are the inputs to corresponding SVM. For example, values of *OA* of flickers are always greater than that of other disturbances. So *OA* is chosen as the input feature of SVM1, and through SVM1, flickers are excluded from other samples. By proceeding iteratively, all kinds of disturbances will be distinguished one by one.

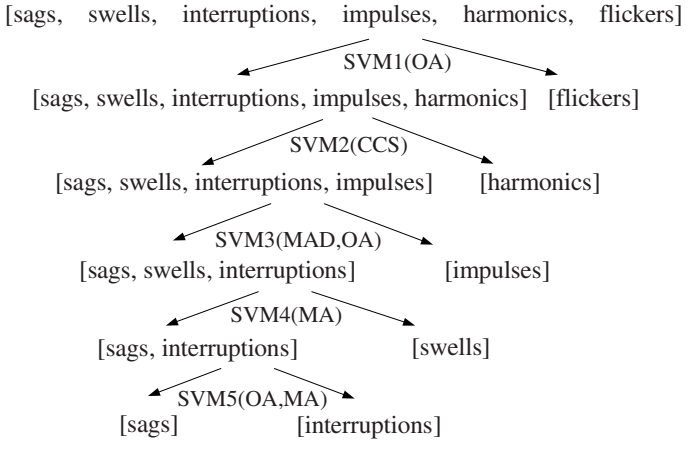


Fig. 4. Tree diagram of sorting logic to classify six different power disturbances

4 Classification Results

To obtain representative signals that possess the inherent characteristics of most common PQ disturbances, disturbance signals are initially generated in MATLAB 7.0. Some unique attributes for each disturbance are allowed to change randomly, within specified limits, in order to create different disturbance cases. The randomness in the generated signals is intended to test the universal validity of classification method proposed in this paper.

Normal frequency of disturbance signals was 50 Hz. Sampling frequency was 4.8 kHz (96 points/cycle). Length of sequence was ten power frequency cycles (960 points). Training samples and testing samples with different parameters were generated according to models introduced in Ref. [4].

We used 120 training samples to optimize the parameters of the SVMs shown in Fig.4, and tested the feasibility of classifier using 1200 testing samples. Considering ubiquitous noise, we added Signal-to-Noise Ratio (SNR) of 40 dB white noise into the testing samples. Table 1 shows the experimental results. It can be found that the constructed and trained SVM classifier results in a correct identification and classification rate of 99.9%, which shows that the proposed PQ disturbance classification method based on PSR and SVM is effective, accurate and reliable.

Table 1. Experimental results of testing samples added with 40 dB white noise

Patterns	Classification results					
	Sags	Swells	Interruptions	Impulses	Harmonics	Flickers
Sags	199	0	1	0	0	0
Swells	0	200	0	0	0	0
Interruptions	0	0	200	0	0	0
Impulses	0	0	0	200	0	0
Harmonics	0	0	0	0	200	0
Flickers	0	0	0	0	0	200
Accuracy	99.9%					

5 Conclusion

Combining with a time series analysis tool, a novel power disturbance classification method is presented. This method consists of two parts: PSR-based signal processing and SVM-based classifier. The function of the former is to extract features from disturbance signals, and the latter is to recognize and classify different types of power disturbances. Several typical disturbances are taken into consideration. The evaluation results with 120 training samples and 1200 testing samples demonstrate that proposed method can effectively classify different kinds of PQ disturbances in poor SNR and training sample size conditions.

Our proposed method reduces calculation burden in feature extraction stage using PSR-based approach instead of mathematical manipulation such as wavelet transform or S-transform. The size of the feature set is also greatly reduced compared with other existing techniques. Furthermore, linear SVM classifier function is computationally

much simpler. These characteristics make the proposed method a proper candidate for on-line disturbance recognition. And the idea of the combining PSR with SVM classifier could potentially be used in other domains, such as audio data analysis, automatic target recognition, etc.

References

1. IEEE standards coordinating committee 22 on power quality: IEEE Recommended Practice for Monitoring Electric Power Quality (1995)
2. Liao, Y., Lee, J.B.: A Fuzzy-Expert System for Classifying Power Quality Disturbances. International Journal of Electrical Power and Energy Systems 26 (3) (2004) 199-205
3. Tiwari, A.K., Shukla, K.K.: Wavelet Transform Based Fuzzy Inference System for Power Quality Classification. Lecture Notes in Computer Science 2275 (2002) 148-155
4. He, H.B., Starzyk J.A.: A Self-Organizing Learning Array System for Power Quality Classification Based on Wavelet Transform. IEEE Transaction on Power Delivery 21 (1) (2006) 286-295
5. Gaouda, A.M., Kanoun, S.H., Salama, M.M.A.: On-Line Disturbance Classification Using Nearest Neighbor Rule. Electric Power Systems Research 57 (1) (2001) 1-8
6. Chilukuri, M.V., Dash, P.K.: Multiresolution S-Transform-Based Fuzzy Recognition System for Power Quality Events. IEEE Transaction on Power Delivery 19 (1) (2004) 323-330
7. A. M. Youssef, T. K. Abdel-Galil, E. F. El-Saadany, et al. Disturbance classification utilizing dynamic time warping classifier. IEEE Transaction on Power Delivery 19 (1) (2004) 272-278
8. Tong, W.M., Song, X.L., Zhang, D.Z.: Recognition and Classification of Power Quality Disturbances Based on Self-adaptive Wavelet Neural Network. Lecture Notes in Computer Science 3972 (2006) 1386-1394
9. Germen, E., Ece, D.G., Gerek, Ö.N.: Self Organizing Map (SOM) Approach for Classification of Power Quality Events. Lecture Notes in Computer Science 3696 (2005) 403-408
10. Kantz, H., Schreiber, T.: Nonlinear Time Series Analysis. Cambridge University Press (1997)
11. Li, Z.Y., Wu, W.L.: Detection and Identification of Power Disturbance Signals Based on Nonlinear Time Series. Proceedings of the WCICA2006 9 (2006) 7646-7650
12. Nello, C., John, S.T.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press (2000)

Mining the Impact Factors of Threads and Participators on Usenet Using Link Analysis

Hongbo Liu, Jiaxin Wang, Yannan Zhao, and Zehong Yang

State Key Lab of Intelligent Technology and System,
Department of Computer Science and Technology, Tsinghua University,
Beijing, P.R. China
liuhb1@gmail.com

Abstract. Usenet is a world-wide distributed discussion system, and it is one of the representative resources on Internet. The structure of newsgroup on Usenet forms gradually along with the evolution of the newsgroup and could provide helpful information for the users. In this paper, we present a method to evaluate the impact factors of participators and threads based on the structure of newsgroup. Some analysis and experimental results on real data sets are also provided. The impact factors could provide useful intrinsic information of the newsgroup and can be used in some applications to help access information more efficiently on the Usenet. The method can be also applied on other discussion systems, such as web forums, bulletin board systems, and so on.

1 Introduction

Usenet is world-wide discussion system on various topics. It provided a convenient way for the communication and organization of discussions. Comparing with web pages, the content of postings on Usenet is generally more informal, brief and personalized. It contains rich information and ideas contributed by the participators. It is difficult to access the information needed efficiently from all the postings in a group due to its huge size and loose organization. People can only subscribe a few groups and generally read a small fraction of the postings. It may take quite much time for a newbie to familiar with a group and its participators to use the group sensibly.

On WWW, because of the intrinsic hyperlink property of web pages, link analysis based on ideas of social networks has been used in the ranking system of some search engines [1,2]. The simplicity, robustness and effectiveness of link-based ranking method have been witnessed with the success of Google, whose basis of ranking system is PageRank. Social networks have been applied in other domains, such as marketing [3], email relationship [4] and so on.

As most postings on the Usenet do not contain hypertexts, they can not be benefited from these link-based algorithms of WWW directly. The briefness and casualty of newsgroup postings make it difficult for conventional text mining techniques. Some investigations based on social networks have also been done to extract useful information from the Usenet [5,6].

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 688–695, 2007.

On the Usenet, person is judged only by his postings. Some people are influential and popular in the newsgroup owing to their character and knowledge background. Some threads are more helpful and valuable than other threads due to their posting contents. The calculation of impact factors of participators (IFP) and impact factors of threads (IFT) can offer useful references for the newsgroup users and could be used in the ranking system of Usenet search in organizing the search results. Therefore, good IFP and IFT can provide intrinsic properties of a group and make the information on Usenet more accessible.

In this paper, according to the characteristics of Usenet, a link-based method to calculate the impact factors of participators and threads on Usenet is proposed. Some mathematical analysis of this method is discussed and experimental results on real data sets are also given.

2 IFP, IFT and Their Calculation

2.1 Usenet Group and Its Representation

The postings on a newsgroup of Usenet were organized by threads. Each thread is invoked by one seed posting and followed by several response postings. The structure of a newsgroup can be represented with weighted bipartite graph $G(P, T, E)$. In the graph, there are two classes of nodes $p_i \in P, t_i \in T$, which represent participators and threads respectively. Considering a newsgroup containing n_p participators and n_t threads, there are totally $n_\mathrm{p}+n_\mathrm{t}$ nodes in G. If one participator p_i posted some postings in thread t_j, there is an edge $\{p_i, t_j\} \in E$ connecting node p_i and t_j in G, whose weight is the number of postings by participator p_i in the thread t_i.

The graph of a newsgroup can be represented with posting matrix $\mathbf{M}$, whose row and column represent the participator p_i and thread t_j respectively and the elements of $\mathbf{M}$ equals to the weight of edge $e = \{p_i, t_j\}$. Therefore, $\mathbf{M}$ is a $n_\mathrm{p} \times n_\mathrm{t}$ dimensional nonnegative matrix.

Let n_a be the total number of postings in the newsgroup. A small newsgroup containing 4 participators, 3 threads and 12 postings was shown in Fig. 1. It can be represented as posting matrix

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 4 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix}.$$

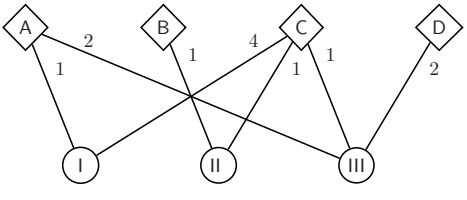


Fig. 1. graph representing newsgroup containing 12 postings

2.2 The Calculation of IFP and IFT

One intuitive idea to calculate IFP is that if a participator posted more postings, he is more influential. Thus, the number of total postings of one participator can

be a candidate of his IFP. Let n_p dimensional row vector $\boldsymbol{f}_\mathrm{p}$ represent the IFP vector of all participators. The IFP value of p_i is $f_{\mathrm{p},i}$. Therefore, the IFP vector can be calculated with $\boldsymbol{f}_\mathrm{p} = (\mathbf{M}\boldsymbol{e}_\mathrm{p})^T$, where $\boldsymbol{e}_\mathrm{p}$ is a n_p dimensional column vector with all ones. Let $\boldsymbol{e}_\mathrm{t}$ be a n_t dimensional column vector. Using $\boldsymbol{f}_\mathrm{t} = \boldsymbol{e}_\mathrm{t}\mathbf{M}$, IFT row vector $\boldsymbol{f}_\mathrm{t}$ can be calculated in similar way.

This method seems to be simple and feasible. However, this calculation works with the assumption that all postings have equal contribution to IFP and IFT, which is not reasonable enough. All postings are not created equal. A posting of more influential participator may have more impact than a posting by unknown one. This posting can improve the impact of the thread containing it. So it is better to consider the difference of the postings when calculating the $\boldsymbol{f}_\mathrm{t}$. This effect of the opposite orientation may also work. A thread with high IFT makes the participators more visible for others. Considering this, the IFP vector $\boldsymbol{f}_\mathrm{p}$ and IFT vector $\boldsymbol{f}_\mathrm{t}$ can be retrieved from the Usenet structure recursively, and we can find that the results have an interesting interpretation with later mathematical analysis.

In the following discussions, by $\mathscr{M}, \mathscr{V}$ we denote matrix space and vector space respectively. Transformation $N : \mathscr{M} \to \mathscr{M}$ is defined as the normalization of the row vectors of matrix based on their l_1 norms. That is, for $\mathbf{A} \in \mathscr{M}$, $N_{ij}(\mathbf{A}) = \mathbf{A}_{ij} / \sum_{j=1}^{n} \mathbf{A}_{ij}$.

After transformed with N, the posting matrix $\mathbf{M}$ was normalized to a stochastic matrix. The elements of normalized posting matrix $N_{ij}(\mathbf{M})$ represent the proportion of postings posted by participator p_i on thread t_j and can be viewed as vote from p_i to t_j. Similarly, $N(\mathbf{M}^T)$ is a $n_\mathrm{t} \times n_\mathrm{a}$ dimensional stochastic matrix and represents the votes of the reversed direction.

The IFP vector $\boldsymbol{f}_\mathrm{p}$ and IFT vector $\boldsymbol{f}_\mathrm{t}$ can be calculated as follow.

$$\boldsymbol{f'}_\mathrm{p}(k) = \boldsymbol{f'}_\mathrm{t}(k-1)N(\mathbf{M}) \tag{1}$$

$$\boldsymbol{f'}_\mathrm{t}(k) = \boldsymbol{f'}_\mathrm{p}(k-1)N(\mathbf{M}^T). \tag{2}$$

Since $\mathbf{M}$ is not a square matrix, the selection of the dimension of initial vector will effect the finial results, so we use two initial vectors and the results need to be merged together. Starting from initial vectors $\boldsymbol{f'}_\mathrm{p}(0) = \boldsymbol{e}_\mathrm{p}$ and $\boldsymbol{f'}_\mathrm{t}(0) = \boldsymbol{e}_\mathrm{t}$, Eq. (1) and (2) can converge to their stable values $\boldsymbol{f'}_\mathrm{p1}, \boldsymbol{f'}_\mathrm{t1}$ and $\boldsymbol{f'}_\mathrm{p2}, \boldsymbol{f'}_\mathrm{t2}$ respectively. Then $\boldsymbol{f}_\mathrm{p}$ and $\boldsymbol{f}_\mathrm{t}$ can be obtained with

$$\boldsymbol{f}_\mathrm{p} = \frac{\boldsymbol{f'}_\mathrm{p1} + \boldsymbol{f'}_\mathrm{p2}}{n_p + n_t}, \quad \boldsymbol{f}_\mathrm{t} = \frac{\boldsymbol{f'}_\mathrm{t1} + \boldsymbol{f'}_\mathrm{t2}}{n_p + n_t}. \tag{3}$$

The properties and convergence of our calculation will be discussed in section 2.3.

The implementation of the IFP and IFT calculation is matrix-free and only $nnz(\mathbf{M})$ multiplications are needed for each iteration, where $nnz(\mathbf{M})$ is the number of non-zeros in $\mathbf{M}$. Since $\mathbf{M}$ is a sparse matrix, $O(nnz(\mathbf{M})) \approx O(n)$. Only the storage of one vector $\boldsymbol{f'}_\mathrm{p}(k)$ or $\boldsymbol{f'}_\mathrm{t}(k)$ is required at each iteration. Thus, this algorithm is suitable for the size and sparsity of posting matrix. We performed some experiments to evaluate the impact factors on realistic datasets. Some experimental results achieved will be discussed in section 3.2.

2.3 Analysis on the Calculation

Let us define $n_\mathrm{p}\times n_\mathrm{p}$ dimensional square matrix $\mathbf{R}_{(\mathrm{p})} = N(\mathbf{M})N(\mathbf{M}^T)$ and $n_\mathrm{t}\times n_\mathrm{t}$ dimensional square matrix $\mathbf{R}_{(\mathrm{t})} = N(\mathbf{M}^T)N(\mathbf{M})$. The following properties of $\mathbf{R}_{(\mathrm{p})}$ and $\mathbf{R}_{(\mathrm{t})}$ can be deduced.

Property 1. *$\mathbf{R}_{(\mathrm{p})}$ and $\mathbf{R}_{(\mathrm{t})}$ are stochastic.*

Proof. Omitted. □

Property 2. *Let $B : \mathscr{M} \to \mathscr{M}$ represents the binary transformation, that is, for $\mathbf{A} \in \mathscr{M}$,*

$$B_{ij}(\mathbf{A}) = \begin{cases} 0 & \text{if } \mathbf{A}_{ij} = 0 \\ 1 & \text{if } \mathbf{A}_{ij} \neq 0. \end{cases}$$

$B(\mathbf{R}_{(\mathrm{p})}), B(\mathbf{R}_{(\mathrm{t})})$ are symmetric, and the diagonals of $B(\mathbf{R}_{(\mathrm{p})}), B(\mathbf{R}_{(\mathrm{t})})$ are $\boldsymbol{e}_\mathrm{p}, \boldsymbol{e}_\mathrm{t}$ respectively.

Proof. Let $n_\mathrm{p} \times n_\mathrm{p}$ dimensional square matrix $\mathbf{S} = \mathbf{M}\mathbf{M}^T$, we have

$$\mathbf{S}_{ij} = \sum_{k=1}^{n_t} \mathbf{M}_{ik}\mathbf{M}^T_{kj} = \sum_{k=1}^{n} \mathbf{M}^T_{ki}\mathbf{M}_{jk} = \mathbf{S}_{ji}.$$

Hence $\mathbf{S}$ is symmetric. Then we can get

$$\begin{aligned} B(\mathbf{R}_{(\mathrm{p})}) &= B(N(\mathbf{M})N(\mathbf{M}^T)) = B(\mathbf{S}) = B(\mathbf{S}^T) \\ &= B(N(\mathbf{S}^T)) = B(N^T(\mathbf{S})) = B^T(\mathbf{R}_{(\mathrm{p})}), \end{aligned} \tag{4}$$

so $B(\mathbf{R}_{(\mathrm{p})})$ is symmetric.

According to definition in section 2.2, there is at least one link for the participator, so $\mathbf{S}_{ii} = \sum_{k=1}^{n_t} \mathbf{M}_{ik}\mathbf{M}^T_{ki} > 0$. Therefore the diagonal of $B(\mathbf{R}_{(\mathrm{p})})$ is $\boldsymbol{e}_\mathrm{p}$. The property of $B(\mathbf{R}_{(\mathrm{t})})$ can be proved in the same way. □

From Eq. (1) and Eq. (2), we can get $\boldsymbol{f'}_\mathrm{p}(k) = \boldsymbol{f'}_\mathrm{p}(k-1)\mathbf{R}_{(\mathrm{p})} = \boldsymbol{f'}_\mathrm{p}(0)\mathbf{R}^k_{(\mathrm{p})}$. When $k \to \infty$, $\boldsymbol{f'}_\mathrm{p}(k) \to \boldsymbol{f'}_\mathrm{p}$, so $\boldsymbol{f'}_\mathrm{p}$ can be calculated using power method and $\boldsymbol{f'}_\mathrm{t}$ can also be obtained with $\boldsymbol{f'}_\mathrm{t} = \boldsymbol{f'}_\mathrm{p}N(\mathbf{M}^T)$. Although $\mathbf{R}_{(\mathrm{p})}$ is a stochastic matrix, it maybe not irreducible. Hence the convergence of power method can not be proved directly and this will be discussed next.

If $\mathbf{R}_{(\mathrm{p})}$ is not irreducible, some nodes can not be accessible from other node in graph G. From the definition of graph G, it means G is not connected and can be decomposed to serval connected components. Suppose G can be decomposed to s connected components, and each component contains n_pccm participators and n_tccm threads where $m = 1, 2, \ldots, s$. There exists permutation matrix $\mathbf{H}$ with which $\mathbf{R}_{(\mathrm{p})}$ can be permuted to a block diagonal matrix $\mathbf{D}_{(\mathrm{p})}$. Thus, $\mathbf{H}$ satisfies

$$\mathbf{D}_{(\mathrm{p})} = \mathbf{H}\mathbf{R}_{(\mathrm{p})}\mathbf{H} = \begin{pmatrix} \mathbf{D}_{(\mathrm{pcc1})} & & 0 & \\ & \mathbf{D}_{(\mathrm{pcc2})} & & \\ & & \ddots & \\ 0 & & & \mathbf{D}_{(\mathrm{pccs})} \end{pmatrix}, \tag{5}$$

where the rows and columns of block $\mathbf{D}_{(\mathrm{pcc1})}, \mathbf{D}_{(\mathrm{pcc2})}, \ldots, \mathbf{D}_{(\mathrm{pccs})}$ represent the participators of connected components. From the definition and discussion above, we can get that each block $\mathbf{D}_{(\mathrm{pccm})}, m = 1, 2, \ldots, s$, is stochastic and its spectral radius equals to 1. Therefore, according to Property 2, we can infer that the Markov chain with transition matrix $\mathbf{D}_{(\mathrm{pccm})}$ is irreducible and ergodic. Suppose row vector $\boldsymbol{v}_{(\mathrm{pccm})}$ satisfies

$$\boldsymbol{v}_{(\mathrm{pccm})}\mathbf{D}_{(\mathrm{pccm})} = \boldsymbol{v}_{(\mathrm{pccm})}, \quad m = 1, 2, \ldots, s. \tag{6}$$

From Eq. (6), $\boldsymbol{v}_{(\mathrm{pccm})}$ is the left eigenvector of $\mathbf{D}_{(\mathrm{pccm})}$, and its normalization $\boldsymbol{v}_{(\mathrm{pccm})}/\|\boldsymbol{v}_{(\mathrm{pccm})}\|_1$ is a probability vector which is the stable solution of Markov chain with transition matrix $\mathbf{D}_{(\mathrm{pccm})}$. $\boldsymbol{v}_{(\mathrm{pccm})}$ can be calculated using power method with any non-zero initial vector and the rate of convergence corresponding to the second largest eigenvalue of $\mathbf{D}_{(\mathrm{pccm})}$.

Since $\mathbf{H}$ is permutation matrix, $\mathbf{H} = \mathbf{H}^T = \mathbf{H}^{-1}$. According to Eq. (5),

$$\begin{aligned} \boldsymbol{f'}_{\mathrm{p}}(k) &= \boldsymbol{f'}_{\mathrm{p}}(0)\mathbf{R}^k_{(\mathrm{p})} = \boldsymbol{f'}_{\mathrm{p}}(0)(\mathbf{H}\mathbf{D}_{(\mathrm{p})}\mathbf{H})^k = \boldsymbol{f'}_{\mathrm{p}}(0)\mathbf{H}\mathbf{D}^k_{(\mathrm{p})}\mathbf{H} \\ &= \hat{\boldsymbol{f}}_{\mathrm{p}}(0) \begin{pmatrix} \mathbf{D}^k_{(\mathrm{pcc1})} & & & 0 \\ & \mathbf{D}^k_{(\mathrm{pcc2})} & & \\ & & \ddots & \\ 0 & & & \mathbf{D}^k_{(\mathrm{pccs})} \end{pmatrix} \mathbf{H}, \end{aligned} \tag{7}$$

where $\hat{\boldsymbol{f}}_{\mathrm{p}}(0) = \boldsymbol{f'}_{\mathrm{p}}(0)\mathbf{H}$. When $k \to \infty$, Eq. (7) yields

$$\boldsymbol{f'}_{\mathrm{p}} = \lim_{k\to\infty} \boldsymbol{f'}_{\mathrm{p}}(k) = [\boldsymbol{v}_{(\mathrm{pcc1})}\boldsymbol{v}_{(\mathrm{pcc2})} \ldots \boldsymbol{v}_{(\mathrm{pccs})}]\mathbf{H}.$$

Since $\mathbf{R}_{(\mathrm{p})}$ and $\mathbf{R}_{(\mathrm{t})}$ are a stochastic square matrices, the l_1 norms of $\boldsymbol{f'}_{\mathrm{p}}(k)$ and $\boldsymbol{f'}_{\mathrm{t}}(k)$ keep unchanged after each iteration. Therefore, $\|\boldsymbol{f'}_{\mathrm{p}1}\|_1 = \|\boldsymbol{f'}_{\mathrm{t}1}\|_1 = \|\boldsymbol{f'}_{\mathrm{p}}(0)\|_1 = n_{\mathrm{p}}$, and $\|\boldsymbol{f'}_{\mathrm{p}2}\|_1 = \|\boldsymbol{f'}_{\mathrm{t}2}\|_1 = \|\boldsymbol{f'}_{\mathrm{t}}(0)\|_1 = n_{\mathrm{t}}$. According to Eq. (3), the l_1 norm of $\boldsymbol{v}_{\mathrm{pccm}}$ in $\boldsymbol{f}_p$ is

$$\|\boldsymbol{v}_{\mathrm{pccm}}\|_1 = \frac{n_{\mathrm{pccm}} + n_{\mathrm{tccm}}}{n_{\mathrm{p}} + n_{\mathrm{t}}}, \quad m = 1, 2, \ldots, s.$$

Therefore $\boldsymbol{f}_{\mathrm{p}}$ equals to the permuted eigenvectors of the $\mathbf{D}_{(\mathrm{p})}$ and the IFP of each participator corresponds to the value of stable solution of Markov chain with transition matrix $\mathbf{D}_{(\mathrm{pccm})}$, which derives from the connected components in G. Similarly, IFT equals to the eigenvector value of block matrix in $\mathbf{D}_{(\mathrm{t})}$, which is a block diagonal matrix transformed from $\mathbf{R}_{(\mathrm{t})}$. Therefore, IFP and IFT in our calculation are intrinsic properties of the posting matrix and could reflect the nature features of G, so they might be good measures of the participators and threads in a newsgroup.

3 Experiments and Their Results

3.1 Datasets Preparation

We wrote a bot program in Perl to download the postings from the nntp server. The bot program communicates with nntp server using socket connection

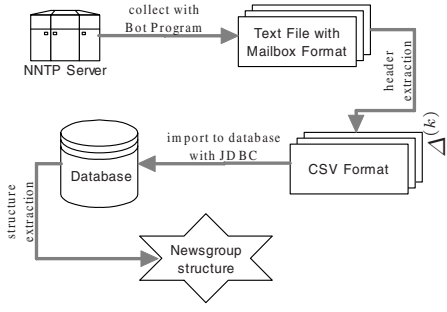


Fig. 2. The process of data collection and structure extraction of Usenet newsgroup

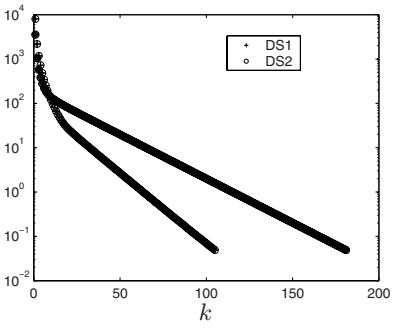


Fig. 3. The convergence rates of experiments on DS1 and DS2

following RFC 977 specification [7] and save the postings in text file with Mailbox format. Since only the headers are needed in our calculation, the headers were separated from the postings from the Mailbox file, and they are stored using csv format after some text treatment. The contents in csv file were ordered and imported to database. SQL statements were performed on the database by a Java program through JDBC interface to construct and extract the structure of newsgroup based on the header information. The process of data sets collection and newsgroup structure extraction was shown in the diagram of Fig. 2.

Experiments were preformed on two data sets collected from comp.lang.perl.misc and comp.lang.python, which are two active newsgroups about computer languages on Usenet. The datasets are called DS1 and DS2 in the following.

DS1 contains 10532 postings including 1286 participators and 1774 threads of comp.lang.perl.misc from Mar 5, 2006 to Jun 27, 2006. DS2 contains 18821 postings including 2463 participators and 3408 threads of comp.lang.python from Mar 5, 2006 to Jun 27, 2006.

3.2 Experimental Results

We measure the rates of convergence using the l_1 norm of the residual vector, i. e.,

$$\Delta^{(k)} = \|\boldsymbol{f}(k) - \boldsymbol{f}(k-1)\|_1.$$

The convergence rates of in our experiments of DS1 and DS2 were plotted on semi-log graph shown in Fig. 3. Our method could converge rapidly, which follows $O(\alpha^k)$ where $\alpha \in (0, 1)$.

According to the analysis in section 2.3, we can get $\|f_\mathrm{p}\|_1 = \|f_\mathrm{t}\|_1 = 1$. Since the impact factors are small, the logarithms of IFT and IFP of DS1 and DS2 are shown in the histogram Fig. 4 and Fig. 5. In these histograms, most of the impact factors are quite small and a few impact factors are relatively high. Heavy tail features are exhibited in these distributions of IFP and IFT. In DS1, the participator with highest IFP is fredrik@pythonware.com whose IFP is 4.195×10^{-2} and the highest IFT thread is "What is Expressiveness in a

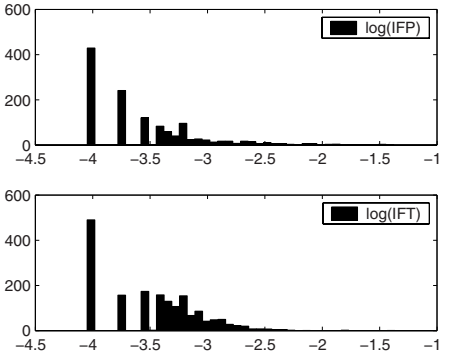


Fig. 4. Histograms of the logarithm of IFT and IFP of DS1

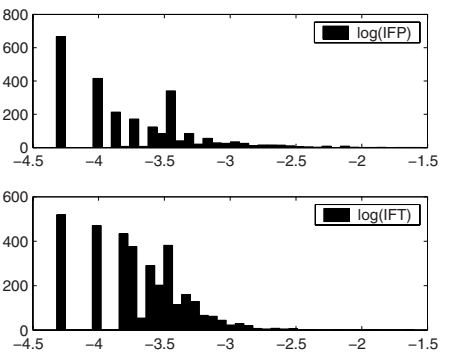


Fig. 5. Histograms of the logarithm of IFT and IFP of DS2

Computer Language" with IFT 4.297×10^{-2}. In DS2, tadmc@augustmail.com got the highest IFP of 2.911×10^{-2} and the highest IFT thread is also "What is Expressiveness in a Computer Language" with IFT 2.481×10^{-2}.

IFP and IFT results can be used in some mining tasks on Usenet. For example, we can obtain the characteristics of each participator by considering IFP and IFT simultaneously. Let $\mathcal{T}(p_i)$ be the set of threads participated by p_i. We can define $f_{\mathrm{t}}^{\mathrm{ave},i}$ as the average IFT of $\mathcal{T}(p_i)$, that is

$$f_{\mathrm{t}}^{\mathrm{ave},i} = \sum_{t_j \in \mathcal{T}(p_i)} f_{\mathrm{t},j}/m_{\mathrm{t}}^i,$$

where m_{t}^i is size of $\mathcal{T}(p_i)$. The relationship of vector $\boldsymbol{f}_{\mathrm{t}}^{\mathrm{ave}}$ and $\boldsymbol{f}_{\mathrm{p}}$ in DS1 was shown in Fig. 6. In this figure, each symbol represents a participator. Only a few participators have both high $f_{\mathrm{t}}^{\mathrm{ave},i}$ and high $f_{p,i}$. The IFP of p_i is influenced by the $f_{\mathrm{t}}^{\mathrm{ave},i}$ and the number of postings. A participator can improve his IFP by participating threads with high IFT or by posting a lot of postings. The average IFT of participator who owns very high IFP is generally not very high.

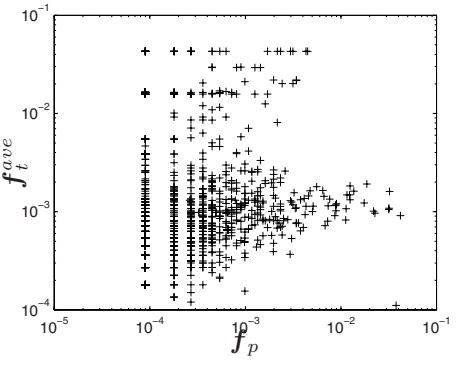


Fig. 6. The relationship of $\boldsymbol{f}_t^{ave}$ and $\boldsymbol{f}_p$

4 Conclusions and Discussions

In this paper, we proposed a method to mine the impact factors of the participators and threads according to the characteristics of Usenet. From the analysis, we can see that our method can converge rapidly. Its results correspond to the

eigenvectors of block matrices and the stationary vectors of Markov chains of connected components.

Our method is link-based and content independent, and it can be computed offline using only the posting headers. Therefore, it can be implemented on the servers of newsgroup services or on the newsgroup client softwares. The results could provide useful intrinsic information of the newsgroup and can be used in many applications including helping organizing the search results, investigating hot topics and their evolutions for some period and so on.

Our method provides an essential way to determine the impact factors. Some improvements can be done based on it to adjust the results according to the requirements. For example, the seed posting invokes the whole thread and plays special roles in the newsgroup, so it is better to give extra bonus to the seed postings other than the response postings in some situations.

On the WWW, it has been confirmed that link carries less noisy information than text, and the effectiveness of link analysis has been testified by some web search engines. Similar with web structure, the structure of newsgroup forms gradually along with the evolution of newsgroup. It represents the judgments and choices of participators. Therefore, it could provide rich information for the mining assignments on Usenet. Together with the IR methods based on text contents, link analysis can be used in the clustering, topic discovery, etc..

References

1. Brin, S., Page, L., Motwanl, R., Winogard, T.: The pagerank citation ranking: Bring order to the web. Technical report, Stanford University (1999) Available at http://dbpubs.stanford.edu:8090/pub/1999-66.
2. Kleinberg, J.: Authoritative sources in a hyperlinked environment. Journal of the ACM **46**(5) (1999) 604–632
3. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proc. of The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York: ACM Press (2001)
4. Schwartz, M.F., Wood, D.C.M.: Discovering shared interests using graph analysis. Communications of the ACM **36**(8) (1993) 78–89
5. Borgs, C., Chayes, J.T., Mahdian, M., Saberi, A.: Exploring the community structure of newsgroups. In: Proc. of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. (2004) 783–787
6. Agrawal, S.D., Rajagopaian, S., Srikant, R., Xu, Y.: Mining newsgroups using networks arising from social behavior. In: Proc. of the Twelfth International World Wide Web Conference, New York: ACM Press (2003)
7. w3.org: Network news transfer protocol. (Intenet(WWW)) http://www.w3.org/Protocols/rfc977/rfc977.
8. Langville, A.N., Meyer, C.D.: Deeper inside pagerank. Internet Mathmatics **1** (2003) 335–380

Weighted Rough Set Learning: Towards a Subjective Approach

Jinfu Liu, Qinghua Hu, and Daren Yu

Harbin Institute of Technology, 150001 Harbin, China
{liujinfu, huqinghua, yudaren}@hcms.hit.edu.cn

Abstract. Classical rough set theory has shown powerful capability in attribute dependence analysis, knowledge reduction and decision rule extraction. However, in some applications where the subjective and apriori knowledge must be considered, such as cost-sensitive learning and class imbalance learning, classical rough set can not obtain the satisfying results due to the absence of a mechanism of considering the subjective knowledge. This paper discusses problems connected with introducing the subjective knowledge into rough set learning and proposes a weighted rough set learning approach. In this method, weights are employed to represent the subjective knowledge and a weighted information system is defined firstly. Secondly, attribute dependence analysis under the subjective knowledge is performed and weighted approximate quality is given. Finally, weighted attribute reduction algorithm and weighted rule extraction algorithm are designed. In order to validate the proposed approach, experimentations of class imbalance learning and cost-sensitive learning are constructed. The results show that the introduction of appropriate weights can evidently improve the performance of rough set learning, especially, increasing the accuracy of the minority class and the AUC for class imbalance learning and decreasing the classification cost for cost-sensitive learning.

Keywords: weighted rough set; knowledge reduction; rule extraction; class imbalance learning; cost-sensitive learning.

1 Introduction

In many practical machine learning problems, such as class imbalance learning and cost-sensitive learning, in order to gain better performance, some subjective and aprior knowledge about applications, such as class distribution and misclassification cost, are necessary. Recently, these kinds of learning problems have been recognized as a crucial problem in machine learning and data mining because such problems are encountered in a large number of domains, such as fraud detection [1], medical diagnosis [2] and text classification [3]. In some cases, the absence of consideration of the subjective knowledge causes seriously negative effects on the performance of machine learning methods. In order to introduce the subjective and aprior knowledge into machine learning, many learning methods have been developed [4], [5], [6].

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 696–703, 2007.

However, most of the research efforts are devoted to making decision trees, neural networks and SVM subjective [7], [8], [9].

Rough set theory, proposed by Pawlak [10], has shown powerful capability dealing with inconsistent information in attribute dependence analysis, knowledge reduction and decision rule extraction. However, there are only a few studies to consider the subjective knowledge in rough set. Through assigning each attribute an appropriate weight in the reduction process, Xu C.-Z. introduced some subjective knowledge about attributes into attribute reduction [11]. But the subjective knowledge about objects related with class distribution and misclassification costs can't be considered in this method yet. In [12] and [13], probability rough set was introduced and each object was associated with a probability $p(x)$, which may include some subjective knowledge about objects. However, how to determine the probability under the subjective knowledge was not given. What's more, specific knowledge acquiring algorithms were not presented and systemic experimental analyses were not carried out in their works.

In order to make rough set subjective, we propose a weighted rough set approach in this paper. The rest of the paper is organized as follows. Weighted rough set learning is proposed and discussed in section 2. Experimental studies of class imbalance learning and cost-sensitive learning based on weighted rough set are carried out in section 3. Section 4 concludes this paper.

2 Weighted Rough Set Learning

In this section, we will introduce weights as a representation of the subjective knowledge and proposed a weighted rough set learning method.

A weighted information system is formally denoted by $WIS =< U, W, A, V, f >$, where U is a finite set of objects, W is a weight distribution on U, which can be given by expert or be estimated from data, A is a finite set of attributes, V is the value domain of A, and f is an information function $f : U \times A \to V$. If $A = C \bigcup D$, where C is the condition attribute set and D is the decision attribute set, WIS is called a weighted decision table.

In WIS, weights provide some necessary and additional information about applications, which can not be given by data. Since the equivalence class is the elementary information granule that expresses knowledge, it is necessary to obtain the weight of a set of objects. Let $w(X)$ be the weight of $X \subseteq U$, $w(Y)$ be the weight of $Y \subseteq U$ and $w(X \bigcup Y)$ be the weight of $X \bigcup Y$, if $X \bigcap Y = \varnothing$, then

$$w(X \bigcup Y) = \frac{w(X)p(X) + w(Y)p(Y)}{p(X \bigcup Y)}, \tag{1}$$

where $p(X)$, $p(Y)$ and $p(X \bigcup Y)$ represent respectively the probability of the set of objects X, Y and $X \bigcup Y$.

Since the family of equivalence classes associated with the attribute set A in WIS is the same as that in classical IS, lower and upper approximation of the decision class in WIS are also the same as that in classical IS. However, the quality of classification in

WIS and classical *IS* will be quite different. For $WIS =< U, W, A = C \bigcap D, V, f >$, if $B \subseteq C$ is the condition attribute set, D is the decision attribute set and $Pos_B(D)$ is the B-positive region of classification induced by D, the weighted quality of classification, denoted by $\gamma_B^W(D)$, is defined as

$$\gamma_B^W(D) = \frac{w(pos_B(D))\left|pos_B(D)\right|}{w(U)\left|U\right|}. \tag{2}$$

When the weight of each object of U is equal, the weighted quality of classification will degenerate into the classical quality of classification.

Attribute reduction is a core problem in rough set. Based on the weighted quality of classification, we design a heuristic attribute reduction algorithm under the subjective knowledge as Algorithm 1. This algorithm adds the attribute with the greatest weighted quality of classification to attribute subset $B \subseteq C$ in sequence until $\gamma_B^W(D) = \gamma_C^W(D)$. In order to guarantee $B \subseteq C$ is a reduct of C, at the end of process this algorithm checks whether B is a minimal subset that has the same knowledge representation ability as C, i.e. whether the elimination of any attribute a from B remains $\gamma_B^W(D) = \gamma_C^W(D)$. For the real world applications, in order to restrain the noise, a threshold ε is introduced to stop the reduction process in the above algorithm, i.e. until $\gamma_C^W(D) - \gamma_B^W(D) \le \varepsilon$, instead of until $\gamma_B^W(D) = \gamma_C^W(D)$.

Algorithm 1. Weighted attribute reduction

Input: $WIS =< U, W, A = C \bigcup D, V, f >$ and a threshold ε.

Output: a D-reduct B of C.

1. **begin**
2. compute the maximal weighted quality of classification $\gamma_C^W(D)$;
3. $B \leftarrow \varnothing$;
4. **while** $B \subset C$ **do**
5. **begin**
6. **for** each $a \in C - B$ **do**
7. compute $\gamma_{B\cup\{a\}}^W(D)$;
8. select a_{max} such that $\gamma_{B\cup\{a_{max}\}}^W(D)$ is maximum;
9. $B \leftarrow B \bigcup \{a_{max}\}$;
10. **if** $\gamma_C^W(D) - \gamma_B^W(D) \le \varepsilon$ **then** exit the loop;
11. **end**
12. **for** each $a \in B$
13. **if** $\gamma_C^W(D) - \gamma_{B-\{a\}}^W(D) \le \varepsilon$ **then** $B \leftarrow B - \{a\}$;
14. return B;
15. **end**

Another important problem which can be solved using rough set is rule extraction. For $WIS =< U, W, A = C \bigcap D, V, f >$, if $Q \subseteq A$ and $E \in U / IND(Q)$, then $Des(E, Q) = \wedge(a = f_a(E))$, where $a \in Q$, is called the description of class E with

respect to Q. If $B \subseteq C$, $X \in U / IND(B)$ and $Y \in U / IND(D)$, a decision rule r can be represented as an assertion of the form

$$Des(X, B) \rightarrow Des(Y, D), \tag{3}$$

where $Des(X, B)$ is the condition part of r and $Des(Y, D)$ is the decision part of r.

Nowadays, there are many known rule extraction algorithms inspired by the rough set theory. Among these algorithms, LEM2 algorithm, proposed by Grzymala in [14], is one of the most used rough set based rule induction algorithm. In LEM2, a generalized decision is defined firstly, which may be a decision, or may be the conjunction of more than one decision. According to the generalized decisions, the objects is partitioned into a family of disjoint subsets of objects, denoted by $\tilde{Y}$. Each of $\tilde{Y}$ may be the lower approximation of a decision class $Y \in U / IND(D)$, or may be one of the disjoint subsets of the boundary of a decision class. For instance, assume that three decision classes, Y_1, Y_2, Y_3, are roughly defined in the decision table, then the boundary of Y_1 will consist of three disjoint subsets, i.e. $BND(Y_1) = (\overline{B}Y_1 \cap \overline{B}Y_2 - \overline{B}Y_3) \cup (\overline{B}Y_1 \cap \overline{B}Y_3 - \overline{B}Y_2) \cup (\overline{B}Y_1 \cap \overline{B}Y_2 \cap \overline{B}Y_3)$, where $\overline{B}Y_1$, $\overline{B}Y_2$ and $\overline{B}Y_3$ represent respectively the upper approximation of Y_1, Y_2 and Y_3. Obviously, $\tilde{Y}$ is consistent with respect to the generalized decisions. For each consistent object set $K \in \tilde{Y}$, LEM2 uses a heuristic strategy to generate a minimal rule set of K.

On the basis of LEM2, we design a rule extraction algorithm under the subjective knowledge as Algorithm 2. In Algorithm 2, c is an element of the description of class with respect to the condition attribute set and Φ is a conjunction of such element c being a candidate for condition part of a decision rule. Additionally, Φ_G denotes the set of elements currently considered to be added to the conjunction Φ and $[\Phi]$ denotes the cover of Φ.

In order to evaluate discovered rules and predict the unseen objects, the weighted support coefficient and confidence coefficient are defined at first. For $WIS =< U, W, A = C \cap D, V, f >$, if $B \subseteq C$, $X \in U / IND(B)$ and $Y \in U / IND(D)$, the weighted support coefficient $\mu_s^W(r)$ and weighted confidence coefficient $\mu_c^W(r)$ of a decision rule $r : Des(X, B) \rightarrow Des(Y, D)$, are defined as

$$\mu_s^W(r) = \frac{w(X \cap Y)|X \cap Y|}{w(U)|U|}, \quad \mu_c^W(r) = \frac{w(X \cap Y)|X \cap Y|}{w(X)|X|}. \tag{4}$$

Suppose that there are N rules $r_1, r_2, \cdots, r_N$ matched with the description of the new objects and there are K decisions $d_1, d_2, \cdots, d_K$, then the overall weighted support coefficient of rules with decision d_k is computed as

$$\mu_s^W(d_k) = \sum_{r_i \in d_k} \mu_s^W(r_i), \tag{5}$$

where $r_i \in d_k$ means that the decision of rule r_i is d_k.

According to the principle of majority voting, the decision algorithm can give the decision of an unseen object with the following guideline

$$d : \mu_s^W(d) = \max_k \mu_s^W(d_k). \tag{6}$$

Algorithm 2. Weighted rule extraction

Input: a set of objects $K \in \widetilde{Y}$
Output: rule set R of K

1. **begin**
2. $G \leftarrow K$, $R \leftarrow \varnothing$;
3. **while** $G \neq \varnothing$ **do**
4. **begin**
5. $\Phi \leftarrow \varnothing$, $\Phi_G \leftarrow \{c : [c] \bigcap G \neq \varnothing\}$;
6. **while** ($\Phi = \varnothing$) or (not($[\Phi] \subseteq K$)) **do**
7. **begin**
8. for each $c \in \Phi_G$, select c_{max} such that $w([c] \bigcap G) \mid [c] \bigcap G \mid$ is maximum;
9. $\Phi \leftarrow \Phi \bigcup \{c_{\max}\}$, $G \leftarrow [c_{max}] \bigcap G$;
10. $\Phi_G \leftarrow \{c : [c] \bigcap G \neq \varnothing\}$, $\Phi_G \leftarrow \Phi_G - \Phi$;
11. **end**
12. **for** each $c \in \Phi$ **do**
13. **if** $[\Phi - c] \subseteq K$ **then** $\Phi \leftarrow \Phi - \{c\}$;
14. create rule r basing on the conjunction Φ;
15. $R \leftarrow R \bigcup \{r\}$, $G \leftarrow K - \bigcup_{r \in R}[r]$;
16. **end**
17. **for** each $r \in R$ **do**
18. **if** $\bigcup_{s \in R-r}[S] = K$ **then** $R \leftarrow R - r$;
19. **end**

3 Experimental Studies

In order to validate the proposed algorithm, in this section, we will carry out the experimental studies of class imbalance learning and cost-sensitive learning based on weighted rough set.

In the experiments, 20 UCI data sets[15], which consist of 10 two-class data sets(echocardiogram, Hepatitis, heart_s, breast, horse, votes, credit, breast_w, tictoc, german) and 10 multi-class data sets(zoo, lymphography, wine, machine, glass, audiology, heart, solar, soybean, anneal), are used. In these data sets, class distribution is imbalanced. Especially, the size ratio of the majority class to the minority class is from 1.25 to 3.84 for the two-class data sets. For the multi-class data sets, the size ratio of the maximum class to the minimum class is from 1.48 to 85.5, and the size of most minimum class is less than 10 objects. In order to perform the experiment based on weighted rough set, the preprocessing on data sets is done at first. In each data set, missing values on continuous attributes are set to the average value while those on

nominal attributes are set to the majority value, and all the continuous attributes are discretized via entropy (MDLP) [16].

3.1 Class Imbalance Learning

For class imbalance learning, the introduction of the subjective knowledge of class distribution is necessary to achieve the satisfying results. In order to improve the classification accuracy of minority classes, we need assign greater weights to the objects of minority classes. Here we use the inverse class probability as the weight of each object. Formally, the inverse class probability weight of x is computed by $1/p(D(x))$, where $D(x)$ denotes the decision equivalence class containing object x.

Via 10-fold cross-validations, the experimental results obtained by classical rough set (RS) and weighted rough set (WRS) are shown in Table 1. It can be found that weighted rough set learning improves evidently the accuracy of the minimum class by averagely 0.0689 on all data sets. At the same time, the accuracy of the maximum class decreases by averagely 0.0463 and the overall accuracy decreases by averagely 0.0140. The AUC is a popular classification performance measure for class imbalance learning. In the experiment, the AUC achieved by WRS is bigger than that achieved by RS on most of data sets and increases by averagely 0.0161 on all data sets. Through the consideration of the subjective knowledge of class distribution, WRS improves greatly the accuracy of the minimum class and increases the AUC.

Table 1. Detail results in class imbalance learning

Data set	Accuracy of minimum class		Accuracy of maximum class		Overall accuracy		AUC	
	RS	WRS	RS	WRS	RS	WRS	RS	WRS
echocardiogram	0.7100	0.7350	0.8056	0.7722	0.7780	0.7626	0.7578	0.7536
hepatitis	0.7167	0.7833	0.9083	0.9026	0.8700	0.8779	0.8125	0.8429
heart_s	0.7583	0.7750	0.7733	0.7733	0.7667	0.7741	0.7892	0.7908
breast	0.2833	0.4125	0.7412	0.6967	0.6049	0.6118	0.5845	0.6261
horse	0.9560	0.9632	0.9699	0.9699	0.9648	0.9675	0.9738	0.9773
votes	0.9221	0.9221	0.9437	0.9513	0.9357	0.9403	0.9385	0.9404
credit	0.7719	0.7947	0.7781	0.8146	0.7754	0.8058	0.8115	0.8321
breast_w	0.9210	0.9252	0.9520	0.9564	0.9414	0.9457	0.8307	0.8458
tictoc	0.8099	0.8127	0.8978	0.9010	0.8674	0.8705	0.8856	0.8947
german	0.0567	0.8733	0.9771	0.4071	0.7010	0.5470	0.5169	0.6402
zoo	0.8000	0.8000	1.0000	1.0000	0.9409	0.9518	0.9312	0.9462
lymphography	0.2690	0.5214	0.8042	0.8139	0.7295	0.7286	0.6926	0.7036
wine	0.9600	0.9600	0.9714	0.9714	0.9778	0.9778	0.9829	0.9829
machine	1.0000	1.0000	0.8103	0.7192	0.6750	0.6788	0.7406	0.7646
glass	0.6500	0.7500	0.7054	0.6661	0.7338	0.6818	0.8218	0.8137
audiology	0.2000	0.1000	0.9100	0.8967	0.7617	0.7522	0.7868	0.7956
heart	0	0	0.8037	0.7493	0.5280	0.4985	0.5715	0.5562
solar	0	0.0333	0.9191	0.7271	0.8297	0.6716	0.5263	0.5257
soybean	1.0000	1.0000	0.8111	0.8667	0.8097	0.8667	0.9008	0.9463
anneal	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Average	0.6392	0.7081	0.8741	0.8278	0.8096	0.7956	0.7928	0.8089

3.2 Cost-Sensitive Learning

Many practical classification problems have different costs associated with different types of errors. For example, in medical diagnosis, the error committed in diagnosing someone as healthy when they have a life-threatening disease is usually considered to be far more serious (thus, higher cost) than the opposite type of error. In such applications, cost-sensitive learning is necessary.

In order to perform cost-sensitive learning using weighted rough set, it is necessary to know the weight of each object associated with misclassification cost. This can be solved according to [7], where misclassification cost matrix is divided into three types. Suppose that $Cost(i, j)$ denotes the cost of misclassifying an object of the ith class to the jth class and $w(i)$ denotes the weight of the ith class associated with misclassification cost, then $w(i)$ is defined respectively as follows:

(a) $1.0 < Cost(i, j) \le 10.0$ only for a single value of $j = J$ and $Cost(i, j \ne J) = 1.0$ for all $j \ne i$, then $w(i) = Cost(i, J)$ for $j \ne J$ and $w(J) = 1.0$.

(b) $1.0 \le Cost(i, j) = H_i \le 10.0$ for each $j \ne i$ and at least one $H_i = 1.0$, then $w(i) = H_i$.

(c) $1.0 \le Cost(i, j) \le 10.0$ for all $j \ne i$ and at least one $Cost(i, j) = 1.0$, then $w(i) = \sum_j Cost(i, j)$.

For each type of cost matrix, via 10-fold cross-validations with randomly generated cost matrices belonging to the same cost type, the average experimental results of cost-sensitive learning obtained by classical rough set (RS) and weighted rough set (WRS) on 20 data sets are shown in Table 2. The results show that weighted rough set decreases the number of errors of high cost class and the overall classification cost for all types of cost matrix, although it increases the overall number of errors.

Table 2. Average results in cost-sensitive learning

Cost matrix type	(a)		(b)		(c)	
	RS	WRS	RS	WRS	RS	WRS
Misclassification cost	19.6748	14.4895	25.6586	20.9979	28.4562	22.041
Number of errors	7.2250	8.6850	7.2250	8.4300	7.2250	8.6600
Number of errors of high cost class	3.0450	1.7100	4.2250	2.9700	4.4950	3.2300

4 Conclusions

In this paper, we have introduced a weighted rough set learning method to consider the subjective and aprior knowledge in machine learning. Some basic definitions of classical rough set are extended under the subjective weights, and weighted attribute reduction algorithm and weighted rule extraction algorithm are designed.

Our experimental results show convincingly that weighted rough set learning achieves the better performance than classical rough set in both class imbalance learning and cost-sensitive learning. The introduction of appropriate weights directly contributes to this improved performance, respectively, increasing the accuracy of the minority class and the AUC for class imbalance learning and decreasing the classification cost for cost-sensitive learning.

References

1. Fawcett R.E., Provost F.: Adaptive fraud detection. Data Mining and Knowledge Discovery 3 (1) (1997) 291–316
2. Japkowicz N., Stephen S.: The Class Imbalance Problem: A Systematic Study. Intelligent Data Analisis 6(5) (2002):429–450
3. Weiss G.M., Provost F.: The Effect of Class Distribution on Classifier Learning: an Empirical Study. Technical Report ML-TR-44, Rutgers University, Department of Computer Science (2001)
4. Japkowicz N.: Learning from Imbalanced Data Sets: A Comparison of Various Strategies. Working Notes of the AAAI'00 Workshop Learning from Imbalanced Data Sets (2000) 10–15
5. Weiss G., Provost F.: Learning When Training Data are Costly: The Effect of Class Distribution on Tree Iinduction. Journal of Artificial Intelligence Research 19(2003) 315–354
6. Maloof M.A.: Learning When Data Sets are Imbalanced and When Costs Are Unequal and Unknown. Proc. Working Notes ICML'03 Workshop Learning from Imbalanced Data Sets (2003)
7. Ting K.M.: An Instance-Weighting Method to Induce Cost-Sensitive Trees. IEEE Trans. Knowledge and Data Eng. 14(3)(2002) 659–665
8. Brefeld U., Geibel P., Wysotzki F.: Support Vector Machines with Example Dependent Costs. Proc. 14th European Conf. Machine Learning (2003) 23–34
9. Zhou Z.-H., Liu X.-Y.: Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. IEEE Trans. Knowledge and Data Eng. 18(1) (2006) 63–77
10. Pawlak Z.: Rough Sets. International Journal of Computer and Information Sciences 11 (1982) 341–356
11. Xu C.-Z., Min F.: Weighted Reduction for Decision Tables. Fuzzy systems and knowledge discovery, proceedings lecture notes in computer science (2006) 246–255
12. Ma T.-H., Tang M.-L.: Weighted Rough Set Model. Sixth International Conference on Intelligent Systems Design and Applications (2006) 481-485
13. Hu Q.-H., Yu D.-R., Xie Z.-X., Liu J.-F.: Fuzzy Probabilistic Approximation Spaces and Their Information Measures. IEEE Transactions on Fuzzy Systems 14 (2) (2006) 191–201
14. Grzymala-Busse J. W.: LERS - a System for Learning from Examples Based on Rough Sets. In R. Slowinski, (ed.) Intelligent Decision Support, Kluwer Academic Publishers (1992) 3–18
15. Blake C., Keogh E., Merz C.J.: UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, Univ. of California, Irvine, http://www.ics.uci.edu/~mlearn/MLRepository.html (1998)
16. Fayyad U., Irani K.: Discretizing Continuous Attributes While Learning Bayesian Networks. In Proc. Thirteenth International Conference on Machine Learning, Morgan Kaufmann (1996) 157–165

Multiple Self-Splitting and Merging Competitive Learning Algorithm

Jun Liu and Kotagiri Ramamohanarao

Department of Computer Science and Software Engineering
The University of Melbourne, Victoria, 3010, Australia
{junliu, rao}@csse.unimelb.edu.au

Abstract. The Self-Splitting Competitive Learning (SSCL) is a powerful algorithm that solves the difficult problems of determining the number of clusters and the sensitivity to prototype initialization in clustering. The SSCL algorithm iteratively partitions the data space into natural clusters without *a priori* information on the number of clusters. It starts with only a single prototype and adaptively splits it into multiple prototypes during the learning process based on a split-validity measure. It is able to discover all natural groups; each is associated with a prototype. However, one major problem of SSCL is the slow speed of learning process, because only one prototype can split each time. In this paper, we introduce multiple splitting scheme to accelerate the learning process and incorporates prototypes merging. Besides of these, Bayesian Information Criterion (BIC) score is used to evaluate the clusters. Experiments show that these techniques make the algorithm 5 times faster than SSCL on large data set with high dimensions and achieve better quality of clustering.

1 Introduction

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into subgroups (clusters). It has important applications in many problem domains, such as data mining, document retrieval, image segmentation and pattern classification. One of the well-known methods is the k-means algorithm [3], which iteratively reassigns each data point to the cluster whose center is closest to the data point and then recomputes the cluster centers.

Several algorithms have been proposed previously to determine cluster number (called k) automatically. Bischof *et al.* [2] use a Minimum Description Length (MDL) framework, where the description length is a measure of how well the data are fit by the model optimized by the k-means algorithm. Pelleg and Moore [4] proposed a regularization framework for learning k, which is called X-means. The algorithm searches over many values of k and scores each clustering model. X-means chooses the model with the best score on the data.

Recently, Zhang and Liu presented the SSCL algorithm [7] based on the One Prototype Takes One Cluster (OPTOC) learning paradigm. The OPTOC-based learning strategy has the following two main advantages: 1) it can find natural clusters, and 2) the final partition of the data set is not sensitive to initialization.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 704–711, 2007.

Although promising results have been obtained in some applications [7], the learning speed is slow due to that only one prototype can split at one time. This paper will introduce multiple splitting into SSCL to accelerate the learning speed.

The remainder of this paper is organized as follows. In Section 2, the original SSCL algorithm is introduced. Section 3 will describe the details of multiple splitting and merging. Their performance in identifying Gaussian clusters is compared in Section 4. Finally, Section 5 presents the conclusions.

2 SSCL Algorithm

Clustering is an unsupervised learning process [1]. Given a data set of N dimensions, the goal is to identify groups of data points that aggregate together in some manner in an N-dimensional space. We call these groups "natural clusters." In the Euclidean space, these groups form dense clouds, delineated by regions with sparse data points.

The OPTOC idea proposed in [7] allows one prototype to characterize only one natural cluster in data set, regardless of the number of clusters in the data. This is achieved by constructing a dynamic neighborhood using an online learning vector $\boldsymbol{A_i}$, called the Asymptotic Property Vector (APV), for the prototype $\boldsymbol{P_i}$, such that patterns inside the neighborhood of $\boldsymbol{P_i}$ contribute more to its learning than those outside. Let $|\boldsymbol{XY}|$ denote the Euclidean distance from $\boldsymbol{X}$ to $\boldsymbol{Y}$, and assume that $\boldsymbol{P_i}$ is the winning prototype for the input pattern $\boldsymbol{X}$ based on the minimum-distance criterion. The APV $\boldsymbol{A_i}$ is updated by

$$\boldsymbol{A}_i^* = \boldsymbol{A}_i + \frac{1}{n_{\boldsymbol{A}_i}} \cdot \delta_i \cdot (\boldsymbol{X} - \boldsymbol{A}_i) \cdot \Theta(\boldsymbol{P}_i, \boldsymbol{A}_i, \boldsymbol{X}) \tag{1}$$

where Θ is a general function given by

$$\Theta(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\omega}) = \begin{cases} 1 \text{ if } |\boldsymbol{\mu\nu}| \geq |\boldsymbol{\mu\omega}|, \\ 0 \text{ otherwise,} \end{cases} \tag{2}$$

and δ_i, within the range $0 < \delta_i \leq 1$, is defined as

$$\delta_i = \left(\frac{|\boldsymbol{P}_i \boldsymbol{A}_i|}{|\boldsymbol{P}_i \boldsymbol{X}| + |\boldsymbol{P}_i \boldsymbol{A}_i|} \right)^2. \tag{3}$$

$n_{\boldsymbol{A_i}}$ is the winning counter which is initialized to zero and is updated as follow:

$$n_{\boldsymbol{A}_i} = n_{\boldsymbol{A}_i} + \delta_i \cdot \Theta(\boldsymbol{P}_i, \boldsymbol{A}_i, \boldsymbol{X}). \tag{4}$$

The winning prototype $\boldsymbol{P_i}$ is then updated by

$$\boldsymbol{P}_i^* = \boldsymbol{P}_i + \alpha_i (\boldsymbol{X} - \boldsymbol{P}_i), \tag{5}$$

where,

$$\alpha_i = \left(\frac{|\boldsymbol{P}_i \boldsymbol{A}_i^*|}{|\boldsymbol{P}_i \boldsymbol{X}| + |\boldsymbol{P}_i \boldsymbol{A}_i^*|} \right)^2. \tag{6}$$

If the input pattern $\boldsymbol{X}$ is well outside the dynamic neighborhood of $\boldsymbol{P_i}$, i.e., $|\boldsymbol{P_iX}| \gg |\boldsymbol{P_iA_i}|$, it would have very little influence on the learning of $\boldsymbol{P_i}$ since $\alpha_i \to 0$. On the other hand, if $|\boldsymbol{P_iX}| \ll |\boldsymbol{P_iA_i}|$, i.e., $\boldsymbol{X}$ is well inside the dynamic neighborhood of $\boldsymbol{P_i}$, both $\boldsymbol{A_i}$ and $\boldsymbol{P_i}$ would shift toward $\boldsymbol{X}$ according to Equations (1) and (5), and $\boldsymbol{P_i}$ would have a large learning rate α_i according to Equation (5). During learning, the neighborhood $|\boldsymbol{P_iA_i}|$ will decrease monotonically. When $|\boldsymbol{P_iA_i}|$ is less than a small quantity ε, $\boldsymbol{P_i}$ would eventually settle at the center of a natural cluster in the input pattern space.

Let $\boldsymbol{C_i}$ denote the center, i.e. arithmetic mean, of all the patterns that $\boldsymbol{P}_i$ wins according to the minimum-distance rule. The distance $|\boldsymbol{P_iC_i}|$ measures the discrepancy between the prototype $\boldsymbol{P}_i$ found by the OPTOC learning process and the actual cluster structure in the dataset. After the prototypes have all settled down, a large $|\boldsymbol{P_iC_i}|$ indicates the presence of other natural clusters in the dataset. A new prototype would be generated from the prototype with the largest distance $|\boldsymbol{P_iC_i}|$ when this distance exceeds a certain threshold ξ.

When cluster splitting occurs, the new prototype is initialized at the position specified by a Distant Property Vector (DPV) $\boldsymbol{R_i}$ associated with the mother prototype $\boldsymbol{P_i}$. The idea is to initialize the new prototype far away from its mother prototype to avoid unnecessary competition between the two. Initially, the DPV is set to be equal to the prototype to which it is associated with. Then each time a new pattern $\boldsymbol{X}$ is presented, the $\boldsymbol{R_i}$ of the winning prototype $\boldsymbol{P_i}$ is updated as follows:

$$\boldsymbol{R}_i^* = \boldsymbol{R}_i + \frac{1}{n_{\boldsymbol{R_i}}} \cdot \rho_i \cdot (\boldsymbol{X} - \boldsymbol{R}_i) \cdot \Theta(\boldsymbol{P}_i, \boldsymbol{X}, \boldsymbol{R}_i), \tag{7}$$

where

$$\rho_i = \left(\frac{|\boldsymbol{P}_i\boldsymbol{X}|}{|\boldsymbol{P}_i\boldsymbol{X}| + |\boldsymbol{P}_i\boldsymbol{R}_i|} \right)^2, \tag{8}$$

and $n_{\boldsymbol{R_i}}$ is the number of patterns associated with the prototype $\boldsymbol{P_i}$. Note that unlike $\boldsymbol{A_i}$, $\boldsymbol{R_i}$ always try to move away from $\boldsymbol{P_i}$. After a successful split, the property vectors ($\boldsymbol{A_i}$, $\boldsymbol{R_i}$) of every prototype $\boldsymbol{P_i}$ are reset and the OPTOC learning loop is restarted.

3 Multiple Self-Splitting and Merging Competitive Learning

3.1 Multiple Splitting

The SSCL algorithm as it was described up to this point can only allow one prototype split when it meets the convergence and splitting criteria. This has dramatically slowed down the learning process, especially for the data with large number of points or clusters. We proceed now to demonstrate how to efficiently search for the best number of clusters by letting more prototypes to split in two at the same time.

How can we decide how many prototypes to split at one time? Pelleg and Moore tried to split every centroid into two children and run the test locally

Initialization :

Set the number of clusters $K = 1$;
Set $\boldsymbol{P}_1 = \boldsymbol{R}_1$ at a random location in the input feature space;
Set $\boldsymbol{A}_1$ at a random location far from $\boldsymbol{P}_1$;
Set winning counters $n_{\boldsymbol{A}_1}$ and $n_{\boldsymbol{R}_1}$ to zero;

Learning Loop :

OPTOC Learning:

Repeat

1. Randomly read a pattern $\boldsymbol{X}$ from the data set;
2. Find the winner $\boldsymbol{P}_i$, where $|\boldsymbol{P}_i\boldsymbol{X}| = \min_l |\boldsymbol{P}_l\boldsymbol{X}|$, $l = 1, \ldots, K$. Label $\boldsymbol{X}$ with i;
3. Update the APV $\boldsymbol{A}_i$ using (11);
4. Update the Prototype $\boldsymbol{P}_i$ using (15);
5. Update the DPV $\boldsymbol{R}_i$ using (17);

Until $\max_l |\boldsymbol{P}_l\boldsymbol{A}_l| < \varepsilon$ or number of OPTOC iteration exceeds 10.
Record prototype set with best BIC score;

Multiple Split Stage:

1. For $i = 1 : K_{old}$
   If $|\boldsymbol{P}_i\boldsymbol{C}_i| > \xi$
   Increase K;
   Set $\boldsymbol{P}_K = \boldsymbol{R}_i$;
   End If
   If no splitting, quit the learning loop;
   End For

Reset Stage:

1. Set $\boldsymbol{R}_l = \boldsymbol{P}_l$, $l = K_{old} + 1, \ldots, K$;
2. Set $\boldsymbol{A}_l$ far from $\boldsymbol{P}_l$, $l = K_{old} + 1, \ldots, K$;
3. Set all the winning counters $n_{\boldsymbol{A}_l}$ and $n_{\boldsymbol{R}_l}$, $l = 1, \ldots, K$, to zero;

Merging Clusters :

Repeat

Find cluster i and cluster j that minimize $|\boldsymbol{C_i}\boldsymbol{C_j}| - (\sigma_i + \sigma_j)$ (see (9));
If $|\boldsymbol{C_i}\boldsymbol{C_j}| \leq (\sigma_i + \sigma_j)$
Merge cluster i and cluster j;
Decrease the number of clusters K by 1;
End If

Until no more clusters can be merged.
Calculate BIC score;
The result is the prototype set with best BIC score;

Fig. 1. Pseudo code for the proposed MSSMCL

to make decisions that increasing the number of centroids or not in the X-means algorithm [4]. Unfortunately, SSCL is not splitting prototypes locally. New prototype can be generated far from the parent prototype. The method of splitting locally and testing is not a good strategy for SSCL.

Recall that, the prototype with the largest distance $|\boldsymbol{P_iC_i}|$ which exceeds a certain threshold ξ will split. The new splitting strategy is that all the prototypes with $|\boldsymbol{P_iC_i}| > \xi$ will split, called Multiple Self-Splitting Competitive Learning (MSSCL). This allows an automatic choice of whether to increase the number of prototypes by very few (the current number is very close to the true number, i.e. the end of learning process) or very many (the beginning of learning process). With this strategy, the learning process of SSCL will be accelerated.

3.2 Prototypes Merging

In essence, the MSSCL algorithm starts form one prototype and continues to add prototypes where they are needed until the stop criterion is reached. However, sometimes more than one DPVs are attracted by the same cluster. This may cause more than one prototypes will split to a very close position. It is possible that a natural cluster in the data set would be split into two or more clusters.

One merging scheme is proposed to merge two clusters when these two clusters are close each other to the extent that their joint Probability Density Function (pdf) form a unimodal structure [6]. Let $\boldsymbol{C_i}$ be the center (i.e., mean) of cluster i and δ_i be its standard deviation. In the Multiple Self-Splitting and Merging Competitive Learning (MSSMCL), if two clusters satisfy the following condition, they should be merged into one:

$$|\boldsymbol{C_iC_j}| \leq \delta_i + \delta_j. \tag{9}$$

Only comparing prototypes, merging is a quite efficient way to overcome the potential over clustering problem brought by multiple splitting.

We then define measure of quality for a cluster μ:

$$distortion_\mu = \frac{1}{R} \cdot \sum_i d^2(i, \mu_{(i)}) \tag{10}$$

where i ranges over all input points.

During the MSSMCL learning process presented in this paper, the prototype set that achieves the best BIC score is recorded, and this is the one that is finally output. The pseudo code for the proposed MSSMCL is shown in Fig. 1.

4 Experimental Results

We have conducted experiments on randomly-generated data, as described in [5]. The synthetic experiments were conducted in the following manner. First, a data-set was generated using randomly-selected points (cluster centers). For each data-point, a cluster was first selected at random. Then, the point coordinates were chosen independently under a Gaussian distribution with mean at the cluster center.

In our first experiment, we tested the quality of the MSSMCL solution against that of SSCL. The test data set was 3-D data with 30 clusters, within the range

Table 1. Distortion of SSCL and MSSMCL

Points	80000	90000	100000	110000	120000
SSCL	0.001875	0.001874	0.001853	0.001874	0.001879
MSSMCL	0.001889	0.001873	0.001871	0.00188	0.001876

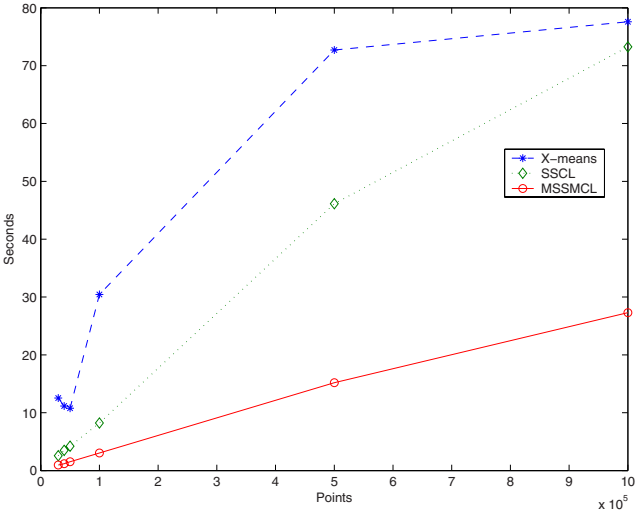


Fig. 2. Average run-times are shown for 2 dimensions and 20 clusters

(0, 1) for each dimension. We compared both algorithms by the distortion of their output. The convergence and splitting thresholds are both set as the same value of deviation σ, 0.025. The results shown on Tab. 1 are average of 30 runs. The two algorithms have achieved similar results in the mean of distortion.

As far as speed is concerned, MSSMCL scales much better than SSCL. One data set was generated as described above with 20 clusters on 2 dimensions contained different number of points, from 30000 to 1000000, respectively drawn this way. The deviation σ equals to 0.5 and each dimension data rage is (0, 10). The SSCL and MSSMCL are running on this data-set and measured for speed. The experiment is repeated 30 times and averages are taken. Fig. 2 shows the run-times of MSSMCL and SSCL with convergence threshold ε and splitting threshold ξ set as 0.5.

Two algorithms were also tested with the number of dimensions varied from 2 to 10 with deviation $\delta = 0.5$. This experiment is repeated 30 times and averages are taken. The convergence and splitting thresholds are both set to 0.5. The number of clusters each algorithm requested to find is 20 and the each cluster has 1000 points. The results are shown in Fig. 3, which shows that MSSMCL runs a half time of SSCL.

Fig. 2 and Fig. 3 also show that both SSCL and MSSMCL run faster than X-means. The run-times are shown on 2.4Ghz Pentium-4.

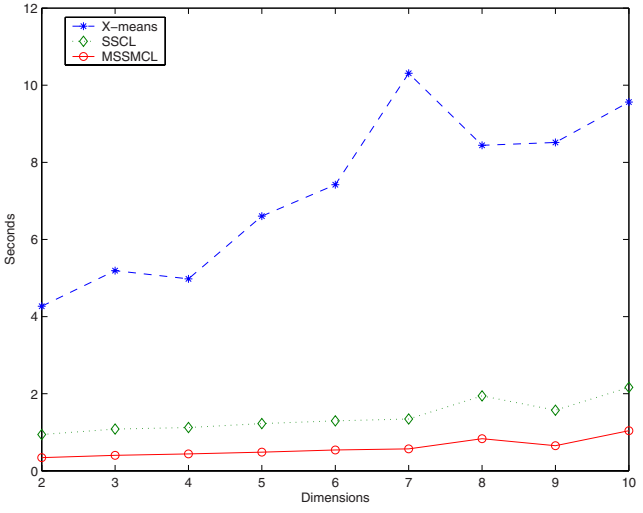


Fig. 3. Run-times shown as the number of dimensions varies, 20 clusters and 20000 points

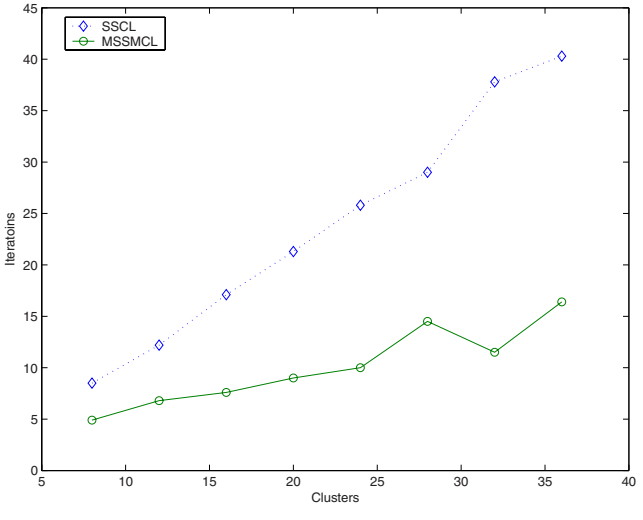


Fig. 4. The number of iterations needed to find the best number of clusters

In terms of run-time, MSSMCL is not only faster, but increases its advantages as the number of points (see Fig. 2), or the number of clusters increases (See Fig. 4) . The total number of points are 10000 with clusters from 8 to 36. Fig. 4 illustrates the iterations needed as a function of number of clusters. We can see that the iterations needed by SSCL is at least the number of the clusters if the natural number of clusters is successfully found. MSSMCL can benefit a lot from the multiple splitting especially for a data set with large number of clusters.

5 Conclusion

We have presented an efficient Multiple Self-Splitting Competitive and Merging Learning algorithm that incorporates BIC scoring and prototypes merging. It uses statistically-based criteria to maximize the model's posterior probabilities. This prevents missing the better prototypes set during the learning process. Our various experimental results on random generated data show that this new algorithm can perform 5 times faster than SSCL on large data set with hight dimensions and achieves better quality of clustering. This new algorithm can be used on large size of data set with high dimensions.

References

[1] Barlow, H.B.: Unsupervised learning. Neural Computation **1** (1989) 295–311
[2] Bischof, H., Leonardiš, A., Selb, A.: MDL principle for robust vector quantization. Pattern Analysis and Applications **2** (1999) 59–72
[3] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proc. 5th Berkeley Symposium on Mathematics, Statistics and Probability, Berkeley, CA: Univ.California Press (1967) 282–297
[4] Pelleg, D., Moore, A.: X-means: Extending K-means with efficient estimation of the number of clusters. In: Proceedings of the 17th International conference on Machine Learning, Morgan Kaufmann, San Francisco, CA (2000) 727–734
[5] Pelleg, D., Moore, A.: Accelerating exact k-means with geometric reasoning. Technical report, Carnegie Mellon University, Pittsburgh, PA. (2000)
[6] Wu, S.H., Liew, A., Yan, H. and Yang. M.S.: Cluster analysis of gene expression data based on self-splitting and merging competitive learning. IEEE Trans. On Information Technology in Biomedicine **8** (2004) 5–15
[7] Zhang, Y.J., Liu, Z.Q.: Self-splitting competitive learning: A new on-line clustering paradigm. IEEE Trans. on Neural Networks **13** (2002) 369–380

A Novel Relative Space Based Gene Feature Extraction and Cancer Recognition

Xinguo Lu[1], Yaping Lin[2,1], Haijun Wang[1], Siwang Zhou[1], and Xiaolong Li[1]

[1] School of Computer and Communication,
Hunan University, Changsha, 410082, China
[2] School of Software, Hunan University, Changsha, 410082, China
hnluxinguo@hotmail.com

Abstract. Recognizing patient samples with gene expression profiles is used to cancer diagnosis and therapy. In the high dimensional, huge redundant and noisy gene expression data the cancerogenic factor's locality is studied. Using gene feature transformation a relative space to a cancer is built and a least spread space with least energy to the cancer is extracted. And it is proven that the cancer is able to be recognized in the least spread space and a cancer classification with least spread space (CCLSS) is proposed. In the Leukemia dataset and Colon dataset the correlation between the recognition rate and the rank of least spread space is explored, then the optimal least spread spaces to AML/ALL and to tumor colon tissue (TCT)/normal colon tissue (NCT) are extracted. At last using LOOCV the experiments with different classification algorithms are conducted and the results show CCLSS makes better precision than traditional classification algorithms.

1 Introduction

Recently huge amount of large-scale gene expression data has been generated as the development of Microarray technique and conduces to cancer diagnosis and therapy [1] [2]. Gene expression profile is a typical high dimensional, huge redundant and noisy data. The research of cancer recognition using gene expression data is usually plagued with "curse of dimensionality" [3].

Dimension reduction is often used to solve the "curse of dimensionality" in cancer detection [4]. A method of dimension reduction is gene selection [5]. In Golub's approach neighborhood analysis based on signal to noise ratio was used to Leukemia cancer dataset and 50 discriminant genes were selected [6]. In Veer's approach, the genes were ranked by correlation against the disease outcome and 70 genes were identified [7]. Cho *et al.* systematically explored different gene selection methods including correlation coefficient method, information gain and mutual information on different cancer datasets. Due to different proximity measures and optimization criterions it's much different for selected genes [8]. Another method is feature transformation. Conde *et al.* presented a clustering based cancer classification and the average values of clusters were used for training a perceptron [9]. Raychaudhur *et al.* summarized the observed variability

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 712–719, 2007.

in two hidden factors with principal components analysis(PCA) in the yeast sporulation dataset [10]. Feature transformation is able to discover the underlying informative factors for classification.

The main idea of cancer classification is that many valuable gene features are acquired via dimension reduction, then these features are applied to train the classification models. However by now all the classification models are constructed on the same set of gene features without regarding of the locality of cancerogenic factor in biology. As Fig.1 shows there are two cancer patterns P and Q. The pattern P corresponds to the cancer samples in the *x*-*y* feature plane which are close to one another. The pattern Q corresponds to the cancer samples in the *x*-*z* feature plane which are also very close. Traditional cancer classifications with dimension reduction do not work for this case.

In this paper a novel relative space(RS) based gene feature extraction and cancer recognition is proposed. Relative space to a cancer is obtained using feature transformation and a least spread space(LSS) with least energy to the cancer is extracted. It is proven that the cancer is able to be recognized in LSS. Then a sample is projected to the LSS of the corresponding cancer. Via projection the gene expression data is compressed efficiently and the noise and redundancy are removed effectively. Also cancer classification with least spread space(CCLSS) is presented to classify patient samples. The experimental results on Leukemia dataset and Colon dataset show CCLSS makes better precision than traditional classification algorithms.

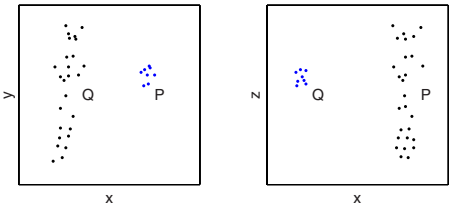


Fig. 1. The locality of cancerogenic factors

2 Pre-requirements

2.1 Related Definitions

If $\widetilde{X}$ is a set of objects and $\widetilde{X}=\{\overline{x}_1, \overline{x}_2, ..., \overline{x}_n\}$, let X be the numerical matrix for the objects in $\widetilde{X}$, $X = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n)$, where $\overline{x}_i = \{x_{i1}, x_{i2}, ..., x_{im}\}^T$.

Definition 1. *$M(\widetilde{X})$ is defined as the* ***mean*** *of $\widetilde{X}$, and $M(\widetilde{X}) = \frac{1}{n}\sum_{i=1}^{n}\overline{x}_i = \frac{1}{n}(\sum_{i=1}^{n} x_{i1}, \sum_{i=1}^{n} x_{i2}, \ldots, \sum_{i=1}^{n} x_{im})^T$*

Definition 2. *If X is a square matrix$(m = n)$, the* ***trace*** *of X is described with $TR(X)$, and $TR(X) = \sum_{i=1}^{n} x_{ii}$*

2.2 Gene Expression Data

The gene expression data is usually represented by expression matrix. Let X be a gene expression matrix, where rows represent genes, columns represent various samples and cell x_{ij} is the measured expression level of gene i in sample j.

3 RS Based Feature Extraction and Cancer Recognition

3.1 Gene Feature Extraction with RS/LSS

Assume in gene expression data there are k cancer classes and n samples. Let $\widetilde{C}_i$ be the set of samples in ith cancer, and there are n_i samples in $\widetilde{C}_i$, so $\sum_i n_i = n$. C_i is the $m \times n_i$ gene expression matrix for $\widetilde{C}_i$. Relative space ε and least spread space $\hat{\varepsilon}$ for $\widetilde{C}_i$ are produced as follow.

Let C_i^T denote the transpose of C_i, $C_i^T = (\overline{g}_1, \overline{g}_2, ..., \overline{g}_m)$, where $\overline{g}_j$ is the jth gene in $\widetilde{C}_i$. Then the covariance matrix $Cov(C_i^T)$ is decomposed as Equ.1.

$$Cov(C_i^T) = \sum \lambda_r \overline{p}_r \overline{p}_r^T = P \wedge P^T \tag{1}$$

where $\lambda_r (1 \leq r \leq m)$ is eigenvalue of $Cov(C_i^T)$ and $\overline{p}_r (1 \leq r \leq m)$ is corresponding eigenvectors, $\wedge$ is a diagonal matrix whose diagonal elements are $\lambda_r (1 \leq r \leq m)$ and $P = (\overline{p}_1, \overline{p}_2, ..., \overline{p}_m)$.

Definition 3. *Given $d \leq m$ as a rank to **relative space(RS)**, $\overline{p}_1, \overline{p}_2, ..., \overline{p}_d$ consist of d dimensional relative space ε for $\widetilde{C}_i$, $\varepsilon = \{\overline{p}_1, \overline{p}_2, ..., \overline{p}_d\}$. $\overline{p}_j (1 \leq j \leq d)$ is denoted as the jth direction, λ_j is its spread coefficient, and $P = (\overline{p}_1, \overline{p}_2, ..., \overline{p}_d)$ is relative space matrix for $\widetilde{C}_i$.*

Definition 4. *Assume $\hat{\varepsilon}$ is RS with rank of d for $\widetilde{C}_i$, and $\hat{\varepsilon}$ is denoted to be **least spread space(LSS)** iff $max(\lambda_1, \lambda_2, \ldots, \lambda_d) \leq min(\lambda_{d+1}, \lambda_{d+2}, \ldots, \lambda_m)$.*

Definition 5. *The set $\widetilde{C}_i$ can be expressed as $\widetilde{C}_i = \{\overline{s}_1, \overline{s}_2, \ldots, \overline{s}_{n_i}\}$, where $\overline{s}_l = (s_{l1}, s_{l2}, \ldots, s_{lm})^T$. The **feature extraction** of $\overline{s}_l$ is denoted as the projection of $\overline{s}_l$ on ε $P(\overline{s}_l, \varepsilon) = (\overline{s}_l \cdot \overline{p}_1, \overline{s}_l \cdot \overline{p}_2, \ldots, \overline{s}_l \cdot \overline{p}_d)^T$, where $\overline{p}_j = (p_{j1}, p_{j2}, \ldots, p_{jm})^T$ is the jth direction of ε, and $\overline{s}_l \cdot \overline{p}_j = \sum_{k=1}^{m} s_{lk} p_{jk}$.*

After feature extraction, the informative data can be compressed efficiently when $d \ll m$ and the features is accompanied with following properties.

Theorem 1. *The variances of variables projected to $\overline{p}_j (1 \leq j \leq d)$ in ε equal to the corresponding spread coefficients $\lambda_j (1 \leq j \leq d)$ and these variables are mutually uncorrelated.*

Proof. Given ith cancer expression matrix $C_i = (\overline{s}_1, \overline{s}_2, \ldots, \overline{s}_{n_i})$. The feature extraction of $\overline{s}_l (1 \leq l \leq n_i)$ on ε is denoted as $P(\overline{s}_l, \varepsilon) = (p_1, p_2, \ldots, p_d)^T$, where $p_j = \overline{s}_l \cdot \overline{p}_j = \overline{p}_j^T \overline{s}_l (1 \leq j \leq d)$. Without loss of generality, $P(\overline{s}_l, \varepsilon)$ can be viewed as normal random variable. So $Var(p_j) = \overline{p}_j^T Cov(C_i^T) \overline{p}_j = \overline{p}_j^T \lambda_j \overline{p}_j = \lambda_j$. $Cov(p_j, p_k)$ is denoted the correlation coefficient between variables p_j and p_k, $Cov(p_j, p_k) = Cov(\overline{p}_j^T \overline{s}_l, \overline{p}_k^T \overline{s}_l) = \overline{p}_j^T Cov(C_i^T) \overline{p}_k = \lambda_k \overline{p}_j^T \overline{p}_k = 0$. So p_j and p_k are unrelated.

Theorem 1 demonstrates that the gene features extracted by above method are mutually uncorrelated and the noise and redundancy are eliminated efficiently. Also the smaller λ_j is, the less difference between the samples in $\widetilde{C}_i$ after extraction is. The feature extraction owns following advantages: (1) local relative features for special cancer are extracted; (2) mutually uncorrelated between extracted features and the noise and redundancy are eliminated; (3) in a cancer the samples are most similar and the cancer pattern is detected.

3.2 Cancer Classification with LSS(CCLSS)

Definition 6. *Given sample* $\overline{s}_j$, $\overline{s}_{j'}$ *and* $\overline{s}_l = (s_{l1}, s_{l2}, \ldots s_{lm})^T (l = j, j')$, *the* ***relative space distance(RSD)*** *between* $\overline{s}_j$ *and* $\overline{s}_{j'}$ *in* ε *is denoted as* $D(\overline{s}_j, \overline{s}_{j'}, \varepsilon)$, *and* $D(\overline{s}_j, \overline{s}_{j'}, \varepsilon) = \|P(\overline{s}_j, \varepsilon) - P(\overline{s}_{j'}, \varepsilon)\| = \sqrt{\sum_{k=1}^{d} (\overline{s}_j \cdot \overline{p}_k - \overline{s}_{j'} \cdot \overline{p}_k)^2}$.

Assume a cancer dataset composed of k cancers and is divided into training set and test set. At first the LSS to either cancer $\widetilde{C}_i$ in training set is obtained via above method. Then for each sample $\overline{t}$ in test set the RSD between $\overline{t}$ and either sample $\overline{s}$ in training set in the LSS of $\widetilde{C}_i$ which $\overline{s}$ belongs to is computed and RSDs with the same cancer label of $\overline{s}$ are summarised as the weight for cancer identification. Finally $\overline{t}$ is identified as the cancer with least weight. The details of CCLSS is described as follows:

Inputting : training set, test set, rank d, assume ith cancer set is $\widetilde{C}_i (1 \leq i \leq k)$.
Begin :
For i=1 to k **do**
1. acquire the covariance matrix $Cov(C_i^T)$ of $\widetilde{C}_i$ from training.
2. compute the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ of $Cov(C_i^T)$ and corresponding eigenvectors $\overline{p}_1, \overline{p}_2, \ldots, \overline{p}_m$.
3. the d least $\lambda'_1, \lambda'_2, \ldots, \lambda'_d$ are selected and $\hat{\varepsilon}_i$ of $\widetilde{C}_i$ is formed with corresponding $\overline{p}'_1, \overline{p}'_2, \ldots, \overline{p}'_d$.

Next
For each $\overline{t}$ in test set **do**
the weight w_i is preassigned with $0 (i = 1 \ldots k)$.
For each $\overline{s}$ in training set **do**
1. assume $\overline{s} \in \widetilde{C}_i$, compute $D(\overline{s}, \overline{t}, \hat{\varepsilon}_i)$ in $\hat{\varepsilon}_i$.
2. calculate weight w_i of $\overline{t} \in \widetilde{C}_i$, $w_i = w_i + D(\overline{s}, \overline{t}, \hat{\varepsilon}_i) (i = 1 \ldots k)$.

Next
t is identified as $\widetilde{C}_{i'}$, if $w_{i'} = min(w_i) (i = 1 \ldots k)$.
Next
End

3.3 Algorithm Analysis

Definition 7. *The* ***class energy*** *of* $\widetilde{C}_i$ *in RS* ε *is described with* $E(\widetilde{C}_i, \varepsilon)$, $E(\widetilde{C}_i, \varepsilon) = \frac{1}{n_i} \sum_{\overline{s}_l \in \widetilde{C}_i, l=1 \ldots n_i} \{D(\overline{s}_l, M(\widetilde{C}_i), \varepsilon)\}^2$.

The class energy describes the distance between mean and sample in a class. The more smaller $E(\widetilde{C}_i, \varepsilon)$ is, the more similar and tighter between samples in $\widetilde{C}_i$ is, or the more different and sparser is. So $E(\widetilde{C}_i, \varepsilon)$ is used to evaluate ε.

Theorem 2. *Given d dimensional RS ε_i to the cancer set $\widetilde{C}_i$, then $E(\widetilde{C}_i, \varepsilon_i) = \sum_{r=1}^{d} \lambda_r$, where λ_r is direction spread coefficient.*

Proof. Assume $\widetilde{C}_i = \{\overline{s}_1, \overline{s}_2, \ldots, \overline{s}_{n_i}\}$, so $E(\widetilde{C}_i, \varepsilon_i) = \frac{1}{n_i} \sum_{s_l \in \widetilde{C}_i, l=1\ldots n_i} \{D(\overline{s}_l, M(\widetilde{C}_i), \varepsilon_i)\}^2$ where $M(\widetilde{C}_i)$ is mean of $\widetilde{C}_i$. From **Definition 6**, we get $D(\overline{s}_l, M(\widetilde{C}_i), \varepsilon_i) = \|p(\overline{s}_l, \varepsilon_i) - p(M(\widetilde{C}_i), \varepsilon_i)\| = \sum_{r=1}^{d} (\overline{s}_l \cdot \overline{p}_r - M(\widetilde{C}_i) \cdot \overline{p}_r)^2$. Then $E(\overline{C}_i, \varepsilon_i) = \frac{1}{n_i} \sum_{l=1}^{n_i} \sum_{r=1}^{d} (\overline{s}_l \cdot \overline{p}_r - M(\widetilde{C}_i) \cdot \overline{p}_r)^2 = \frac{1}{n_i} \sum_{r=1}^{d} \sum_{l=1}^{n_i} (\overline{s}_l \cdot \overline{p}_r - M(\widetilde{C}_i) \cdot \overline{p}_r)^2$. By **Theorem 1** we can demonstrate $\sum_{l=1}^{n_i} (\overline{s}_l \cdot \overline{p}_r - M(\widetilde{C}_i) \cdot \overline{p}_r)^2 = n_i \lambda_r$, so $E(\widetilde{C}_i, \varepsilon_i) = \sum_{r=1}^{d} \lambda_r$.

The trace $\sum_{r=1}^{d} \lambda_r$ is invariant under the transformation defined by the eigensystem P. If $d = m$ it reaches the peak value which means the least similarity between samples in $\widetilde{C}_i$. Also from **Definition 4** in the d dimensional LLS $\hat{\varepsilon}_i$ the class energy is minimized to $E(\widetilde{C}_i, \hat{\varepsilon}_i)$. So the cancer pattern $\widetilde{C}_i$ can be detected.

4 Experiments

4.1 Gene Expression Dataset and Preprocessing

In this paper two gene expression datasets including Leukemia dataset [6] and Colon dataset [8] are used. In Leukemia dataset 25 of 72 samples are acute myeloid leukemia(AML) and 47 samples are acute lymphoblastic leukemia(ALL). Each sample contains 7129 genes. In Colon dataset 40 of 62 samples are tumor colon tissues(TCT) and the remaining are normal colon tissues(NCT). Each sample contains 2000 genes.

Information index to classification(IIC) [11] is used to filter genes is defined by $\mathrm{IIC}(\overline{g}) = \frac{1}{2} \frac{|\mu_{\overline{g}+} - \mu_{\overline{g}-}|}{\sigma_{\overline{g}+} + \sigma_{\overline{g}-}} + \frac{1}{2} \ln[\frac{\sigma_{\overline{g}+}^2 + \sigma_{\overline{g}-}^2}{2\sigma_{\overline{g}+}\sigma_{\overline{g}-}}]$, where μ and σ are mean and standard deviation of gene expressions within corresponding class. In Leukemia dataset the threshold to IIC is assigned to 0.8 and 196 genes are selected for following experiments. In Colon dataset the threshold is assigned to 0.65 and genes are reduced to 93.

4.2 Results and Analysis

Firstly Leukemia dataset and Colon dataset are randomly divided into training set and test set respectively and the corresponding percentages are 70% and 30%. So in Leukemia dataset there are 51 training samples(18 AMLs, 33 ALLs) and 21 test samples(7 AMLs, 14 ALLs) and in Colon dataset there are 43 training samples(28 TCTs, 15 NCTs) and 19 test samples(12 TCTs, 7 NCTs).

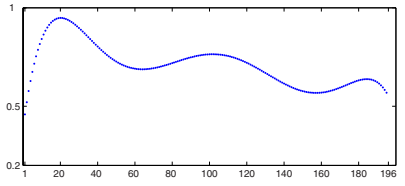


Fig. 2. The average precision distributed with rank d for Leukemia dataset

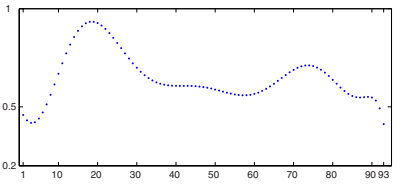


Fig. 3. The average precision distributed with rank d for Colon dataset

Then CCLSS is performed and the precision is calculated. The ranks of LSS for Leukemia are assigned to $1, 2, \ldots, 196$ respectively and the ranks for Colon are $1, 2, \ldots, 93$. The presented scheme is run 10 times for each rank and the average precisions are showed in Fig.2 and Fig.3. Fig.2 shows in Leukemia dataset the maximal precision is acquired when $d = 20$. Also the appropriate rank for Colon dataset is 18(Fig.3).

Then the locality of cancerogenic factor is discussed. In Leukemia the LSSs $\hat{\varepsilon}_{AML}$ and $\hat{\varepsilon}_{ALL}$ are built. The samples distribution is showed in Fig.4 and Fig.5. In Colon the LSSs $\hat{\varepsilon}_{TCT}$ and $\hat{\varepsilon}_{NCT}$ are also constructed. The samples distribution is showed in Fig.6 and Fig.7. In Fig.4 and Fig.5 the dimensions of

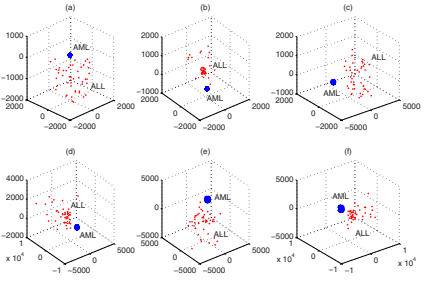


Fig. 4. For Leukemia dataset samples distribution in $\hat{\varepsilon}_{AML}$

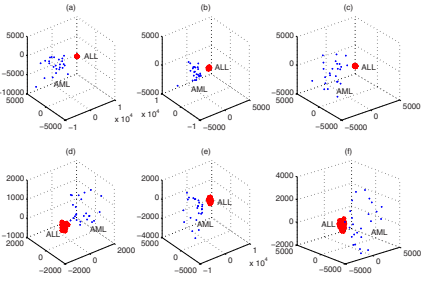


Fig. 5. For Leukemia dataset samples distribution in $\hat{\varepsilon}_{ALL}$

$\hat{\varepsilon}_{AML}$ and $\hat{\varepsilon}_{ALL}$ are ranked by their corresponding spread coefficients ascendingly and denoted as *dim1, dim2, . . . , dim20*. In their subfigures Fig.4(a) and Fig.5(a) *dim1*∼*dim3* are selected. And *dim5*∼*dim7*, *dim8*∼*dim10*, *dim11*∼*dim13*, *dim15*∼*dim17*, and *dim18*∼*dim20* are choosen for subfigures (b), (c), (d), (e) and (f). Fig.4 and Fig.5 show that in $\hat{\varepsilon}_{AML}$ AML samples are mutually close and ALL samples are distributed very sparsely while in $\hat{\varepsilon}_{ALL}$ ALL samples are distributed tightly and AML samples dispersively. So cancer patterns AML and ALL can be detected in $\hat{\varepsilon}_{AML}$ and $\hat{\varepsilon}_{ALL}$ respectively. Also in Fig.6 and Fig.7 the dimensions are denoted as *dim1, dim2, . . . , dim18*. Then *dim1*∼*dim3*, *dim4*∼*dim6*, *dim7*∼*dim9*, *dim10*∼*dim12*, *dim13*∼*dim15* and *dim16*∼ *dim18* are used for subfigures (a), (b), (c), (d), (e) and (f) in Fig.6 and Fig.7 respectively. In $\hat{\varepsilon}_{TCT}$ we can distinguish the TCT samples from NCT samples and in $\hat{\varepsilon}_{NCT}$ the NCT samples

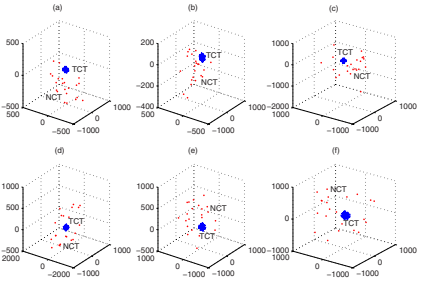


Fig. 6. For Colon dataset samples distribution in $\hat{\varepsilon}_{TCT}$

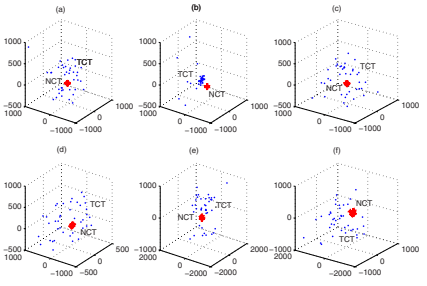


Fig. 7. For Colon dataset samples distribution in $\hat{\varepsilon}_{NCT}$

Table 1. Classification results comparison using LOOCV

	Leukemia		Colon	
Method	Features	Accuracy	Features	Accuracy
CCLSS	20	71	18	61
Weighted Voting	50	65	40	58
SVM	50	69	40	57
KNN	50	62	40	52
Clustering	50	63	40	53

also can been picked out. So the locality of cancerogenic factor is illustrated by these four figures.

Finally Leave-one-out cross-validation(LOOCV) is used. We pick up the first sample of the dataset(Leukemia or Colon) as a test sample, and the remaining samples as training set. Repeating through the first sample to the last one, we can get an accuracy, the number of samples which are correctly predicted. The accuracies are shown in Table 1. Also the accuracies acquired by other methods including *Weighted Voting* [6], the *Clustering* based perceptron model [9], *SVM* with RBF kernel and *KNN* (k=7) are presented. From Table 1, we can see that using *LSS* the gene expression data can be compressed efficiently. In Leukemia cancer dataset only 20 gene features are extracted for CCLSS and the best accuracy is acquired while larger features are used in other traditional algorithms. And in Colon cancer dataset the 18 extracted gene features are applied to CCLSS instead of original data and the accuracy is also better than any other one.

5 Conclusion

In this paper the cancerogenic factor's locality to a cancer is explored in the gene expression data. It is proven that a cancer pattern is able to be recognized in the *LLS*. In the Leukemia and Colon dataset the *LLS*s to AML/ALL and to TCT/NCT are extracted respectively. Then the CCLSS based on LSS is tested using LOOCV and compared with other algorithms and the results

show CCLSS makes better accuracy with smaller gene features. Using LLS the gene expression data can be compressed efficiently and CCLSS is appropriate to cancer recognition.

References

1. M. Kuramochi, and G. Karypis. "Gene classification using expression profiles:a feasibility study". International Journal on Artificial Intelligence Tools, 14(4):641-660, 2005.
2. X.G. Lu, Y.P. Lin, X.L. Yang, L.J. Cai, H.J. Wang, S.Gustaph. Using Most Similarity Tree Based Clustering to Select the Top Most Discriminating Genes for Cancer Detection. In the proceeding of The Eighth International Conference on Artificial Intelligence and Soft Computing (ICAISC). Zakopane, Poland, 931-940, 2006.
3. L. Parsons, E. Haque, and H. Liu. "Subspace clustering for high dimensional data: a review". SIGKDD Explorations, 6(1):90-105, 2004.
4. N. Kasabov. "Evolving Connectionist Systems, Methods and Applications in Bioinformatics". Brain Study and Intelligent Machines, Verlag Springer, 2002.
5. M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, and *et al.* "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning". Nature Medicine, 8:68-74, 2002.
6. T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, and *et al.* "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring". Science, 286:531-537, 1999.
7. L.J.V.T. Veer, H. Dai, M.J.V.D. Vijver, Y.D. He, and *et al.* "Gene expression profiling predicts clinical outcome of breast cancer". Nature, 415:530-536, 2002.
8. S.B. Cho, H.H. Won. "Machine Learning in DNA Microarray Analysis for Cancer Classification". In Proc. of the First Asia-Pacific Bioinformatics Conference(APBC 2003), 189-198, 2003.
9. L. Conde, A. Mateos, J. Herrero, and J. Dopazo. "Unsupervised Reduction of the Dimensionality Followed by Supervised Learning with a Perceptron Improves the Classification of Conditions in DNA Microarray Gene Expression Data". Neural Networks for Signal Processing XII. IEEE Press(New York). Eds. Boulard, Adali, Bengio, Larsen, Douglas. pp.77-86, 2002.
10. S. Raychaudhur, J.M. Stuar, R.B. Altma. "Principal Components Analysis to Summarize Microarray Experiments: Application to Sporulation Time Series". Pacific Symposium on Biocomputing Honolulu, Hawaii, 452-463, 2000.
11. Y.X. Li, and X.G. Ruan. "Feature Selection for Cancer Classification Based on Support Vector machine". Journal of Computer Research and Development, 42(10):1796-1801, 2005.

Experiments on Kernel Tree Support Vector Machines for Text Categorization

Ithipan Methasate[1,2] and Thanaruk Theeramunkong[1]

[1] Sirindhorn International Institute of Technology (SIIT), Thammasat University, 131 Moo 5, Tiwanont Road, Muang, Phathumthani 12000, Thailand
thanaruk@siit.tu.ac.th
[2] National Electronics and Computer Technology Center,112 Phahon Yothin Road, Klong Luang, Pathumthani 12120, Thailand
ithipan.methasate@nectec.or.th

Abstract. Text categorization is one of the most interesting topic, due to the extremely increase of digital documents. The Support Vector Machine algorithm (SVM) is one of the most effective technique for solving this problem. However, SVM requires the user to choose the kernel function and parameters of the function, which directly effect to the performance of the classifiers. This paper proposes a novel method, named Kernel Tree SVM, which represents the multiple kernel function with a tree structure. The functions are selected and formed by using genetic programming (GP). Moreover, the gradient descent method is used to perform fine tune on parameter values in each tree. The method is benchmarked on WebKB and 20Newsgroup datasets. The results prove that the method can find a bettr optimal solution than the SVM tuned with the gradient method.

1 Introduction

With the fast growth of the digital documents in internet, text categorization or classification (TC) becomes a key role in organizing and retrieving the massive online documents. Text categorization are applied in many application such as, information retrieval, news and e-mail categorization. During a few decades, there were many researchers attracted in this topic. Many statistical techniques were proposed to solve the TC problems, such as, expectation maximization [13], decision tree and rules [11], k-nearest neighbor [14], Bayesian classification and Centroid-Based method [16,13]. Among the statistical techniques, the SVM is one of the most efficient technique in solving classification problem. In SVM, the hyperplane is constructed to seperate two groups of data. Not only the accuracy of classification, the margin between classes is maximized also. However, SVM require the user to setup some parameters to the systems. Firstly, there are many kind of mapping function can be selected. Then, the value of parameters in mapping function and SVM itself need to set. The setting affects directly to the performance of SVM.

Recently, there are some studies focusing on parameter tuning for SVM. As an early work, Chapelle and Vapnik [4] presented a well-known method called the

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 720–727, 2007.

gradient descent algorithm for choosing multiple SVM parameters. To achieve invariance against linear transformations due to the problem of scaling and rotation in the space of SVM parameters, the covariance matrix adaptation evolution strategy (CMAES) was initiated by Friedrichs and Igel [5,**?**]. Howley and Madden proposed the genetic kernel for SVM, and evaluation technique of genetic kernel. Moreover, L1 and L2 SVM was proved to use Radius-Margin bound with BFGS Quasi-Newton method [6,10].

In this paper, the technique for tuning SVM parameter is described. The multiple kernel function is presented by the tree structure, called Kernel Tree. Kernel Tree is guaranteed to be positive semi-define function. In parameter tuning step, the genetic programming and BFGS Quasi-Newton method are used to search the optimal in global and local space respectively. In the rest of this paper, Section 2 describes the concept of SVM, kernel mapping and Mercer's theory. Section 3 presents the genetic programming and kernel tree. In section 4, the gradient method for tuning SVM parameter is shown. Section 5 shows the idea when genetic and gradient are combined together. In section 6, experimental results from 20Newsgroup and WebKB are given. Finally, the conclusion is made in section 7.

2 Overview of Support Vector Machines

2.1 Support Vector Machines (SVM)

SVM [1,2,3] is a linear binary learning system, unlikely with other linear classifier e.g. Linear Discriminant Analysis (LDA), SVM finds an optimal separable linear hyperplane by considering both minimum error and maximum margin conditions. Given a set of examples x_i with labels $y_i : \{(x_i, y_i) \mid x_i \in \mathbf{R}^N$ and $y_i \in \{-1, +1\}\}$. The hyperplane is constructed by solving the following constrained quadratic optimization problem:

$$\min_{w,\xi_i,b} \left\{ \frac{1}{2} \|w\|^2 + C \sum_i \xi_i \right\} \tag{1}$$

subject to

$$y_i \left(< x_i, w > + b \right) \geq 1 - \xi_i, \qquad i = 1, \ldots, n$$

and

$$\xi_i \geq 0, \qquad i = 1, \ldots, n$$

where w and b are the variables of a linear classifier. The regularization parameter C is the tradeoff between the empirical error and the complexity of model.

2.2 Kernel Functions, Kernel Properties and Mercer's Theory

The linear SVM can be applied to non-linear problems easily, by using mapping technique called *Kernel trick* . Normally, there are some common kernel functions that frequently used such as, Linear kernel, Polynamial kernel, RBF kernel and Sigmoid kernel. Based on the Mercer's theory, the following properties [2,7] are valid.

Here, $K_1(\cdot)$ and $K_2(\cdot)$ are two arbitrary kernel functions.

1. $K(x_i, x_j) = \alpha_1 K_1(x_i, x_j) + \alpha_2 K_2(x_i, x_j)$ where α_1, α_2 are positive scalar values.
2. $K(x_i, x_j) = K_1(x_i, x_j) K_2(x_i, x_j)$
3. $K(x_i, x_j) = \exp(K(x_i, x_j))$ where $\exp(x)$ is an exponential function of x.
4. $K(x_i, x_j) = x_i^T A x_j$ where A is $n \times n$ positive definite matrix.

With these four properties, a multiple kernel function can be created by applying these theories. For example: $K(x_i, x_j) = \alpha_1 K_1(x_i, x_j)^2 + \alpha_2 K_2(x_i, x_j)^2$ is Mercer's kernel by considering the first and the second properties. Anyway, the more number of functions we combine the more complicated the combined function is and the more parameters the function has.

3 Genetic Programming for Feature Selection

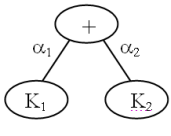
Genetic programming (GP) is one of the most well-known techniques in the field of genetic and evolutionary computation. The population are generate with some randomly citerias. In each generation, there are only good samples are survived and chosen to generate offspring (new generation). Unlike other evolutional method, GP algorithm works with tree structure data. In this paper, the tree structure is designed for representing a kernel function.

3.1 A Kernel Tree

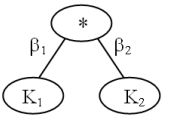
As mentioned above, basically the kernel function satisfies the properties described in Mercer's theory. The tree structure can be designed to fulfill these kernel properties. Therefore, two kinds of nodes can be used to represent a kernel function.

1. Operation Node: A node in a tree that presents an operation with the Mercer 's kernel properties. Two possible nodes are as follow.

- **Additional Node:** it represents the following additional properties
$K(x_i, x_j) = \alpha_1 K_1(x_i, x_j) + \alpha_2 K_2(x_i, x_j)$



- **Power/Multiply Node:** it represents the following properties $K(x_i, x_j) = K_1(x_i, x_j)^{\beta_1} K_2(x_i, x_j)^{\beta_2}$ where β_1 and β_2 are positive integer. This equation can be derived from second property.



2. Basis Kernel Node: A node in a tree that are constructed for representing common kernel functions. These nodes represent leaf nodes. Some common functions are linear, polynomial, and RBF kernels.

- **Linear Kernel Node:** $K(x_i, x_j) = (x_i \cdot x_j)$

- **Polynomial Kernel Node:** $K(x_i, x_j) = ((x_i \cdot x_j) + \theta)^d$

- **RBF Kernel Node:** $K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{c}\right)$

An operation node and a basis kernel node naturally satisfy Mercer's proved properties. Therefore, the function that is constructed as a tree with these nodes are guaranteed to be a Mercer kernel. Figure 1 shows an example of kernel trees represent the equation $K(x_i, x_j) = \alpha_1 \left(K_1(x_i, x_j)^{\beta_1} K_2(x_i, x_j)^{\beta_2}\right) + \alpha_2 (\alpha_3 K_3(x_i, x_j) + \alpha_4 K_4(x_i, x_j))$. In this paper, the sigmoid function is not chosen to be a basis function because the sigmoid function may not be a PSD function for all cases of parameters.

3.2 Genetic Programming Algorithm

In an initial setup, GP generates the tree population randomly. For each iteration, only good trees are survived and used to generate new offspring (new generation). The steps of GP algorithm are:

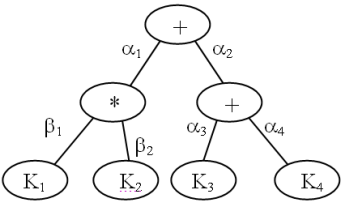


Fig. 1. The tree of function $K(x_i, x_j) = \alpha_1 \left(K_1(x_i, x_j)^{\beta_1} K_2(x_i, x_j)^{\beta_2}\right) + \alpha_2 (\alpha_3 K_3(x_i, x_j) + \alpha_4 K_4(x_i, x_j))$

1. Initial the first generation. In the first generation, p number of trees are generated randomly. Normally, the first generation is used *ramped-half-and-half* method, which half number of population are guaranteed to be maximum depth.
2. Evaluate of each individual. Each individual population is evaluated the fitness by averaging the five-fold cross-validation.
3. Select the best q and generate offspring. The best q individuals are chosen to be parents of new generation. The $p - q$ offsprings are created by using mutation and crossover with the same probability.
4. Repeat Step 2 and 3 until converged. There are two conditions for stop, the first condition is the iteration reachs the threshold. The second condition is the average of best s parents is not change more than t iterations.
5. Build SVM model using the n best tree. Finally, the best s trees can be obtained and used to create the SVM model.

4 Gradient Search for Feature Selection

The gradient method is one of the most effective tools for finding the local optimal solution. The advantages of the gradient method is the convergent speed, comparing to the evolutional strategy. However, this method can find only the optimal point that is close to the current position.

In applying gradient to SVM feature selection, the empirical risk is estimated from radius-magin bound [6,10], or Leave-One-Out (LOO) bound. The gradient of $R^2\|w\|^2$ can be calculated by following equations:

$$\frac{\partial f}{\partial C} = \frac{1}{m}\left[\frac{\partial \|w\|^2}{\partial C}R^2 + \|w\|^2\frac{\partial R^2}{\partial C}\right] \qquad \frac{\partial f}{\partial \theta_i} = \frac{1}{m}\left[\frac{\partial \|w\|^2}{\partial \theta_i}R^2 + \|w\|^2\frac{\partial R^2}{\partial \theta_i}\right] \quad (2)$$

The partial differtiation are shown here:

$$\frac{\partial \|w\|^2}{\partial C} = \sum_i \alpha_i / C^2 \qquad \frac{\partial R^2}{\partial C} = -\sum_i \beta_i \left(1 - \beta_i\right) / C^2 \quad (3)$$

$$\frac{\partial \|w\|^2}{\partial \theta_i} = -\sum_{i,j} \alpha_i \alpha_j y_i y_j \frac{\partial \widetilde{K}\left(x_i, x_i\right)}{\partial \theta_i} \qquad \frac{\partial R^2}{\partial \theta_i} = -\sum_{i,j} \beta_i \beta_j \frac{\partial \widetilde{K}\left(x_i, x_i\right)}{\partial \theta_i} \quad (4)$$

where θ_i is the parameter of the kernel function.

In the task of minimize the objective function, there are several gradient descent algorithms such as BFGS Quasi-Newton's method.

5 Perspective of the Combination Between Genetic and Gradient Method

The nature of genetic approach and gradient approach are not the same. The genetic approach is based on evolutional process that work well on finding the

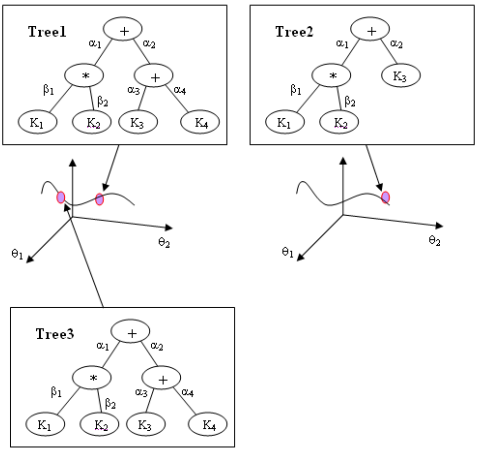


Fig. 2. Tree 1 and Tree 3 ,that have the same tree structure, are corresponding to two points in same parameter space. While tree 2 is in another space.

optimal in a search space. But the convergence speed of the approach is slow due to high computational. Normally, the improvement of GP becomes slow down when the iteration increases. On the other hand, the gradient approach finds an optimal solution near the initial within short period. But it cannot guarantee that the solution is global optimal. In combining both advantages of these two approach, the genetic programming is used in the first step to search in wide area. As shown in Figure 2, each tree is a sample when varying parameters in a space. The trees that have the same structure are sampled from the same space. The system selects only the points that locally give a good evaluation. The gradient method is used to find the optimal besides each selected point.

6 Experimental Results

6.1 Datasets

The experimental was performed on two benchmark datasets that are 20News-Group and Webkb. 20Newsgroups consists of 20 groups of news from Usenet article,which contains around 1,000 documents per group from overall 19,997 documents. WebKB consists of WebKB1 and WebKB2 that contain 7 and 5 classes respectively. The summarized information of these datasets are shown in Table 1 also.

In the experiment, we compares two methods. The first method is Gradient descent (BFGS Quisi-Newton) tuned by adjusting the parameter of single RBF kernel. The initial values are $log(C) = 1$ and $log(\gamma) = 1$. The second method is GP+Gradient method. On the part of GP, the maximum depth is 5. The number of populations for each generation is 30,and selects 10 parents. The GP run 5 iterations to get the best 10 trees. Then, the parameters in each selected tree

Table 1. Characteristic of Webkb and 20Newsgroup Dataset

Dataset	Type	#Docs.	#Cls	#Docs/#Cls
20Newsgroup	Plain Text + Header	19,997	20	1,000
WebKB1	HTML	8282	5	Table 2
WebKB2	HTML	8282	7	Table 2

Table 2. Accuracy (%) of RBF kernel and Tree kernel on 20NewGroup, Webkb1 and Webkb2

Datasets	RBF (Gr)	Tree (GP + Gr)
20NewGroup	88.2967	95.4765
WebKB1	76.8415	85.5897
WebKB2	86.90	94.633

will be finely tuned with BFGS Quisi-Newton. Moreover, 5-fold cross-validation is applied in all experiments.

From the result, the use of GP + Gr method over Tree kernel gets a better performance for all datasets. The main advantage of the proposed method over the gradient method is tree kernel able to represent more the complex function (multiple kernel) while the gradient method needs user to set the function. Also when many different shape tree are generated, it means that we can search more in parameter space. While simple gradient method cannot search out of the function space. However, tree kernel with GP + Gr method need more computational than simple gradient method. So, we need to set the number of GP iteration to limit the computational cost.

7 Conclusion

In the paper, the kernel tree is suggested to solve the problem of SVM parameter tuning. The tree kernel can represent multiple kernel and it still holds properties according to Mercer's theory. The GP the optimization method that used to find the solution of the problem. In GP, the cross-validate is the fitness function. To limit the problem of slow convergence, BFGS Quasi-Newton method is used to do a fine tuning for all selected tree.

Acknowledgement

This work has been supported by Thailand Research Fund (TRF).

References

1. Vapnik V.N.: Statistical Learning Theory. John Wiley and Sons, USA (1998)
2. Burges C.: A Tutorial on Support Vector Machines for Pattern. Data Mining and Knowledge Discovery, **2** (1998) 121–267

3. Cristianini N., Shawe-Taylor J.: An Introduction to Support Vector Machines. Cambridge University Press, (2000)
4. Chapelle O., Vapnik V., Bousquet O., Mukerjee S.: Choosing multiple parameters for support vector machines. Advances in Neural Information Processing Systems, (2001)
5. Friedrichs F., Igel C.: Evolutionary Tuning of Multiple SVM Parameters. Neurocomputing64 (C) (2005) 107-117
6. Chung K.M., Kao W.C.,Sun C.L., Wang L.L., Lin C.J.: Radius margin bounds for support vector machines with the RBF kernel. Neural computation, MIT Press **15** (2003) 2643–2681
7. Genton M.G.: Classes of Kernels for Machine Learning: A Statistics. Jour. of Machine Learning, **2** (2001) 299–312
8. Glamachers T., Igel C.: Gradient-based Adaptation of General Gaussian Kernels. Neural Computation, **17** (10) 2099-2105
9. Tom H., Michael G.M.: The Genetic Kernel Support Vector Machine: Description and Evaluation. Artificial Intelligence Review **10** (2005) 379–395
10. Keerthi, S.S.: Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms. IEEE Transaction on Neural Network, **13** (2002) 1225–1229
11. Apte C.d.,Damerau F.J., Weiss S.M.: Automated learning of decision rules for text categorization. Information Systems, (1994) 233–251
12. Weiss Y.,Schölkopf B.,Platt J.: A General and Efficient Multiple Kernel Learning Algorithm. Advances in Neural Information Processing Systems **18** (2005)
13. Nigam M.,McCallum A.K.,Thrun S.,Mitchell T.M.: Text classification from labeled and unlabeled documents using EM. Machine Learning **39** (2000) 103–134
14. Yang Y.,Liu X.: A re-examination of text categorization methods. Proceeding of SIGIR-99,ACM Press (1999) 42–49
15. Namburu S.M., Haiying T., Jianhui L., Pattipati K.R. : Experiments on Supervised Learning Algorithms for Text Categorization. Proceeding of AERO 2005 (2005) 1–8
16. Verayuth L.,Thanaruk T.: Effect of term distributions on centroid-based text categorization. Information Sciences **158** (2004) 89–115

A New Approach for Similarity Queries of Biological Sequences in Databases

Hoong Kee Ng, Kang Ning*, and Hon Wai Leong

Department of Computer Science, National University of Singapore
3 Science Drive 2, Singapore 117543
{nghoongk,ningkang,leonghw}@comp.nus.edu.sg

Abstract. As biological databases grow larger, effective query of the biological sequences in these databases has become an increasingly important issue for researchers. There are currently not many systems for fast access of very large biological sequences. In this paper, we propose a new approach for biological sequences similarity querying in databases. The general idea is to first transform the biological sequences into vectors and then onto 2-d points in planes; then use a spatial index to index these points with self-organizing maps (SOM), and perform a single efficient similarity query (with multiple simultaneous input sequences) using a fast algorithm, the multi-point range query (MPRQ) algorithm. This approach works well because we could perform multiple sequences similarity queries and return the results with just one MPRQ query, with tremendous savings in query time. We applied our method onto DNA and protein sequences in database, and results show that our algorithm is efficient in time, and the accuracies are satisfactory.

1 Introduction

Biological databases are becoming increasingly important, and their sizes are growing very rapidly, as this is the only plausible way to store newly sequenced protein or gene sequences. For example, GenBank already contains 56,037,734,462 base pairs and 52,016,762 DNA sequences as at 2005. For protein sequences, currently Release 50.8 of Oct 3rd, 2006 of UniProtKB/Swiss-Prot contains 234,112 sequence entries, comprising 85,963,701 amino acids abstracted from 146,463 references. The increasing size of biological databases is a boon for researchers in bioinformatics, but this is only true if they can retrieve information from these databases effectively (reasonable speed and accuracy). Two approaches exist for querying sequences.

The first is to perform database search based on sequence alignment [1]. Sequence alignment algorithms search for, in the database, all the sequences matching the query sequences. Since the time to perform sequence alignment increases proportionally to the size of the database, these algorithms become very slow for large sequence databases because databases size experienced exponential growth.

* Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 728–736, 2007.

The second approach is to use schemes similar to the BLAST algorithm [2, 3] or PatternHunter algorithm [4]. BLAST finds regions of local similarity between sequences. It compares nucleotide or protein sequences to sequences in databases and calculates the statistical significance of matches. It can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families. Though BLAST is fast, it does not guarantee to output all the matched sequences given the query sequences. At the same level of sensitivity as BLAST, PatternHunter is able to find homologies between sequences as large as human chromosomes, in mere hours on an entry-level PC. It can even approach the Smith-Waterman [1] exhaustive dynamic programming sensitivity at speed 3,000 times faster. However, the sensitivity of PatternHunter is still not as close as algorithms based on the first approach.

Accordingly, we ask the question of how to cope with the biological sequences database so that the query is efficient, and it can output as many matched sequences as possible. In this paper, we propose a new approach for sequences similarity search. The basic idea is to map all biological sequences in the database onto a plane as 2-d points through the self-organizing map (SOM) [5], where they are spatially indexed thereafter with a spatial data structure such as the widely-used R-tree. Then, we perform similarity search using our multi-point range query (MPRQ) algorithm [6, 7] that can support multiple simultaneous queries efficiently.

2 SOM, MPRQ and Similarity Query

In this paper, we transform the problem of searching for similar sequences in sequences database to the problem of spatial search of query points in 2-d planes. Our approach, as mentioned before, has two major components: the self-organizing map (SOM) and the multi-point range query (MPRQ) algorithm.

2.1 SOM for Sequences Transformation

The SOM algorithm can be applied on biological sequences to form a similarity "map" via unsupervised training. The map is a grid of artificial nodes which are adapted to closer match sequences from an input database *DB*. Each node is a vector of statistical values. The node that is most similar to an input sequence $S \in DB$, the "winner" node, is updated to be more similar to *S* while the winner's neighbors are also updated to be more similar to *S* but at a smaller degree. As a result, when a SOM is trained over a few thousand epochs, it gradually evolves into clusters whose data (sequences) are characterized by their similarity. Because of this, SOM could indicate relationships between clusters. Therefore, it is very suitable for analysis of the similarities among sequences and is widely used [8].

Biological sequences can be transformed to statistical vectors via SOM. In fact, entities that can be trained by SOM can be very general. For any entity *x* and *y*, a sufficient condition for them to be mapped into a SOM map is that some kind of symmetric distance function $dist \mid dist(x, y)$ is definable for all pairs (x, y). In our

implementation, we used the Euclidean (L_2) distance metric as similarity measure between any 2-d point representing sequences.

2.2 MPRQ for Query

Using the MPRQ algorithm from our earlier work, we are able to quickly perform similarity search of biological sequences in the database. The MPRQ algorithm accepts as input a set of query points and a search distance d, and returns all points (spatially indexed) that is within a distance d from any of the query points. Formally, given a spatial database DB, a set of points P, and a distance d, MPRQ(P, d) = { $p_i \in DB$, $p_j \in P \mid dist(p_i, p_j) \leq d$ }. The MPRQ algorithm supports any 2-d data structure including bulkloaded R-tree [9] which we use.

The general idea behind MPRQ algorithm is to perform only one pass of the R-tree while simultaneously process multiple query points (in this case, transformed from input sequences) in one shot. At each minimum bounding node R in the R-tree, the algorithm processes all the children of R against all the query points. MPRQ takes $O(\log_B n + k/B)$ time where B is the disk page size. Due to space constraints, we refer the readers to [6, 7] for more details. Previous experiments [7] showed that a large input (many points) *does not* increase the overall query time by a lot. This is due to the intelligent pruning rules embedded within MPRQ.

2.3 Similarity Query

The use of SOM is to achieve a high correlation between the proximity of 2-d points on the SOM map and the similarity between mapped sequences. The use of MPRQ algorithm is to achieve the best possible efficiency in finding similar sequences. Together, both SOM and MPRQ, as tools for similarity queries, present an alternative approach to studying the similarity query of biological sequences.

Both the query sequence(s) and sequences databases are transformed into statistical vectors, ready as input for SOM. Once the sequences in database are mapped to a 2-d plane with SOM, we transform the query sequence(s) into query point(s) in 2-d space and proceed to query. At this point, it is possible to use many sequences as the query, which translates to multiple points in 2-d space as the input for MPRQ.

Apart from a set of query points, MPRQ also accepts as input a parameter d that controls the radius of the search distance. The larger the value of d, the more similarity results will be returned.

When the algorithm has terminated, the query results (2-d points, each representing a similar sequence) obtained from MPRQ are collected for analysis. For each point in the result, we inspect the 2-d point object which carries a "tag id" that identifies the original sequence. Fig. 1 illustrates the whole process.

BLAST scores the sequences similarity based on a distance matrix. However, in our approach, we do not use a distance matrix; instead, the similarity measure of our approach is derived from the distance between 2-d points. To compute the number of matched sequences, we score and validate the candidates generated by MPRQ with the same distance matrix as BLAST (Fig. 2 (a)). By doing so, the results for our approach can be fairly compared with BLAST results.

3 Experiments and Results

3.1 Datasets and Experiment Settings

The DNA sequences database are based on the GenBank database (Release 155.0, ftp://ftp.ncbi.nih.gov/blast/db) [10] and the protein datasets are based on the Uni-ProtKB/Swiss-Prot database [11]. The specifications of these datasets are in Table 1.

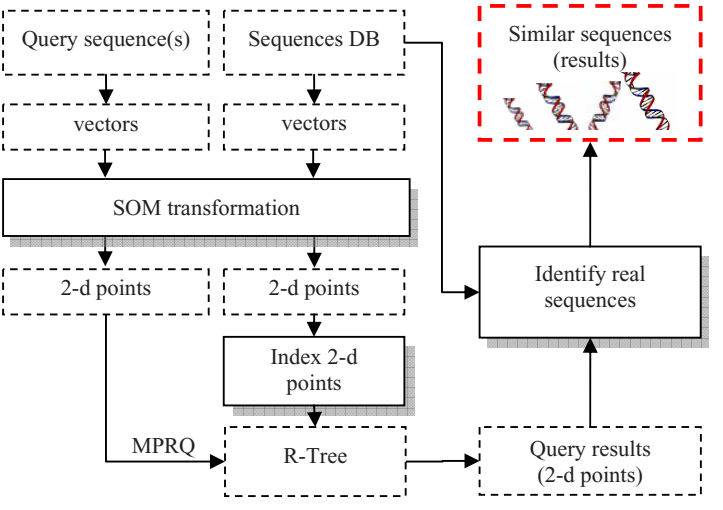


Fig. 1. An novel approach to process similarity sequences queries over the sequences database. Dashed rectangles represent data and solid rectangles represent method or algorithms.

Table 1. Specifications of datasets used for experiments

Type	Datasets	No. of seqs	Average seq length	Query length
DNA	*E. Coli*	400	11,801.8	100–2,000
	Yeast	17	723,939.5	
	Drosoph	1,170	106,144.8	
Protein	*Yeast*	6,298	478.6	100–200
	Drosoph	14,331	507.7	

Notice that for DNA sequences, every dataset has few sequences, but each DNA sequence is very long. On the other hand, for protein sequences the number of sequences is large, and the average sequence length is about 500. It is apparent that the search space is very large for DNA sequences, especially for *Drosoph* datasets. Thus, it is a challenge to intelligently prune the search space for sequences to maintain query efficiencies. Query sequences were randomly selected from their respective datasets.

For comparison with other database search algorithms, we used NCBI BLASTn [4] for sequences search on DNA sequences, and NCBI BLASTx [12, 13] for sequences search on protein sequences. These two algorithms are widely used. For mapping the

biological sequences onto points in planes, we used the SOM_PAK [14] package. We implemented our program in C++ and Perl. The experiments were performed on a Linux PC machine with 3.0GHz CPU and 1.0GB RAM.

The accuracy of the sequences obtained is of great importance. For a result ρ_i and the real sequence ρ of DNA sequences, the accuracy score was computed in the same way as BLASTn for DNA sequences, and BLASTx for protein sequences. We call this function BLAST_Score.

$$accuracy(\rho_i, \rho) = \text{BLAST\_Score}(\rho_i, \rho) \tag{1}$$

The accuracy measures the portions of the real sequence that are in the results. Current algorithms generally use the number of "matches" as an indicator of the query results quality. For a result ρ_i and real sequence ρ, if $accuracy(\rho_i, \rho)$ exceeds a threshold value t (Threshold_{acc}), then we say it is a *match*. The number of match results is defined as

$$|\text{match results}| = \text{No. of } \rho_i \mid accuracy(\rho_i, \rho) > t \tag{2}$$

Since the BLASTn algorithm treats DNA sequences with scores above 10.0 as good results, we had also set the threshold value t of 10.0 and take these DNA sequences as matches. For protein sequences, the threshold value t is also 10.0, and the protein sequences with score above these are considered matches.

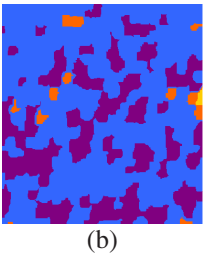
Based on the number of match results, we define the match ratio R_m.

$$R_m = \frac{|\text{match results}|}{|\text{DB search results}|} \tag{3}$$

where "DB search results" are the results given by database search with the combined SOM and MPRQ approach.

Type	DNA	Protein
match	1	BLOSUM62
mismatch	-3	
gap open cost	-5	-11
gap extend cost	-2	-1
Threshold_{acc}	10.0	10.0

(a)



(b)

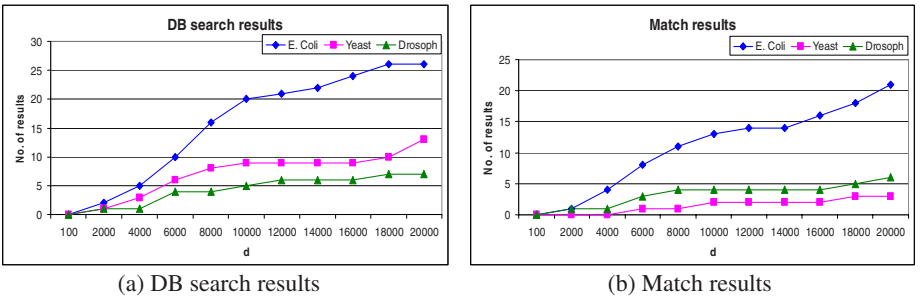
Fig. 2. (a) Scoring for DNA and protein sequences alignment, (b) A sample of SOM training of *E. Coli* for a 100x100 orthogonal grid being visualized. Similar colors (or shades of grey) represent similarity of trained sequences.

3.2 Results

We first analyzed the process of mapping biological sequences onto 2-d plane. The aim is to test how well the distances between points on the planes reflect the similarities between sequences. Results show that the closer the two points on the plane, the more similar the sequences that they represent (details not shown here due to space constraint, but Fig. 3 give some clues of it). And since sequences of high similarity were grouped together by SOM, we believe that SOM is suitable for clustering analysis. Fig. 2(b) shows a sample of the SOM map that we trained from the *E. Coli* dataset.

Next, we analyzed the performance of the spatial index used to index the points from the SOM so as to guarantee efficient querying. Extensive experiments have been conducted in [6, 7] and we had showed that (details not reproduced here) the query time for MPRQ is very fast (< 1 sec) even with a large number of query points.

In Fig. 3 we studied the effect of the search distance d on the number of results obtained from database search, as well as the number of matched results from database search. We observe that both the number of database search results and the number of matched results increase as the search distance d increases. This indicates a high correlation between the proximity of 2-d points on the SOM map and the similarity between the sequences. The number of matches do not increase greatly after d=10000 but the time increase significantly (details not shown here). Thus, in the following experiments, we had selected d=5000 and d=10000 for further analysis and comparison. From the same figure, we also infer that the match ratio R_m is very high. This empirically proves the effectiveness of our algorithm.



(a) DB search results (b) Match results

Fig. 3. The increase of (a) the number of database search results and (b) the number of match results as a function of the search distance d

We also compared our method with other existing biological sequences search algorithms. For DNA sequences from GenBank database, we compared with BLASTn. We tested the effect of the (i) size of inputs for simultaneous processing, (ii) search distance and (iii) total query time. We performed single input query (m = 1) and multiple input query (m = 10) using 10 sequences and we measured the total number of similar sequences. Table 2 depicts the results.

We noticed that our algorithm does not produce as many results as BLASTn. For single query, we think this is because BLASTn is more accurate than our algorithm. And for multiple queries, this is also partly due to the fact that BLASTn does not support multiple inputs; so for m = 10, we perform 10 separate queries, and thus the sequences reported includes some (though not many) overlaps.

The search distance parameter d influences the number of similarity results returned. We had selected d=5000 and d=10000 for experiments. As expected, a larger d value returns more results without significantly increasing the query time. Also, we observed that our algorithm performs very fast, especially for multiple inputs. Compared with BLASTn, our algorithm is faster by 1 to 2 orders of magnitude. The reason behind this is that our approach has transformed the similarity query problem for sequences into range queries of 2-d points.

Table 2. Comparison of the number of matched sequences and query time (secs) between our algorithm and the BLASTn algorithm

Datasets	Search distance (d)	Our algorithm (no. of match seqs)		BLASTn (no. of match seqs)		Our algorithm (secs)		BLASTn (secs)	
		$m = 1$	$m = 10$	$m = 1$	$m = 10$	$m = 1$	$m = 10$	$m = 1$	$m = 10$
E. Coli	5000	5	110	37	403	0.010	0.011	0.1	0.5
	10000	13	303			0.013	0.015		
Yeast	5000	1	32	5	115	0.009	0.012	0.2	1.6
	10000	2	90			0.011	0.011		
Drosoph	5000	2	301	14	869	0.016	0.021	0.3	5.1
	10000	4	500			0.019	0.022		

We also performed experiments on real protein sequences from UniProtKB/Swiss-Prot database. We compared our algorithm with BLASTx. The results are shown in Table 3.

Table 3. Comparison of the number of matched sequences and query time (secs) between our algorithm and the BLASTx algorithm on protein sequences

Datasets	Search distance (d)	Our algorithm (no. of match seqs)		BLASTx (no. of match seqs)		Our algorithm (secs)		BLASTx (secs)	
		$m = 1$	$m = 10$	$m = 1$	$m = 10$	$m = 1$	$m = 10$	$m = 1$	$m = 10$
Yeast	5000	1	16	4	25	0.010	0.012	0.2	1.3
	10000	3	20			0.010	0.014		
Drosoph	5000	3	12	10	36	0.012	0.015	0.1	0.7
	10000	8	31			0.016	0.019		

For single query, the number of matched protein sequences is about 20% to 75% of those by BLASTx on *Yeast* datasets; and are about 30% to 80% on *Drosoph* datasets. For batch query, large amount of matched protein sequences are also observed. This indicates that our algorithm also perform well on protein sequences queries.

Similar to Table 2, the query time of our algorithm is 1 to 2 magnitudes smaller than that of BLASTx. This is very significant, especially for large query sizes. Comparing the query time on *E. Coli*, *Yeast* and *Drosoph* DNA sequences datasets, we observe that the process time did not increase greatly with the increase of dataset size. The experiments on *Yeast* and *Drosoph* protein sequences datasets also show the same phenomenon. Furthermore, the query time is affected only slightly by query size and database size. This is a very good characteristic for larger scale database queries, in which query size is large, and database size is also large. Though we did not perform experiments on larger genome datasets such as BLAST nr dataset, we believe that the query time on large dataset will not increase too much.

We do, however, note that by using our algorithm, the original sequences database must be preprocessed before efficient queries can be performed. Most of the current sequences databases for queries are relatively stable (i.e. published databases are rarely changed radically) and most of the queries are performed on those stable sequences in the database. Therefore, preprocessing time can be seen as only a small one time overhead to the overall query efficiency. For example, the preprocessing time is about 1 minute for *Yeast* DNA dataset, and 3.5 hours for *Yeast* protein dataset. We also note that BLASTn and BLASTx also need preprocessing time. It takes

about 1 minute and a few minutes for these algorithms to preprocess *Yeast* DNA and *Yeast* protein datasets respectively.

Since the preprocessing time is proportional to the size of the datasets, for large datasets such as BLAST nr genome sequences, the preprocessing time will be a big overhead. However, as the number of queries performed increases, the total cost for each query actually decreases. As a result, if a sufficiently large number of queries are serviced, our algorithm can perform better than BLAST as our query time is 1 to 2 orders of magnitude faster.

4 Discussions and Conclusion

As the sizes of biological databases grow, we believe algorithms to effectively search biological sequences in the database will become the focus for many scientists. In this study, we proposed a new approach for indexing biological sequences in database so as to facilitate fast similarity queries. Essentially, this algorithm transforms sequences to vectors and then to 2-d points on SOM map, and use SOM and MPRQ for fast and accurate sequences query. Experiments show that our algorithm is not only efficient but also accurate in searching for similar sequences in database.

Since this is our first attempt on this problem, we know that there are still many areas in our current work that can be improved. For instance, the scale of our experiments is still relatively small, so larger scale experiments are needed to check this approach further. Also, it is possible to research better data structures and algorithms as spatial indexes that take into consideration some characteristics of biological sequences rather than mere 2-d points.

There are still many related problems waiting for further investigations. One of these interesting problems is how to extract biological information from such an indexing process. Since sequences similarity at different levels may indicate their functional relationships, we think research in such areas may be especially useful for the comprehensive comparison of sequences.

References

1. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of Molecular Biology 147 (1981) 195-197
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. Journal of Molecular Biology 215 (1990) 403-410
3. McGinnis, S., Madden, T.L.: BLAST: At the core of a powerful and diverse set of sequence analysis tools. Nucleic Acids Research 32 (2004) W20-W25
4. Ma, B., Tromp, J., Li, M.: PatternHunter: faster and more sensitive homology search. Bioinformatics 18 (2002) 440-445
5. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, New York (2001)
6. Ng, H.K., Leong, H.W., Ho, N.L.: Efficient Algorithm for Path-Based Range Query in Spatial Databases. IDEAS 2004 (2004) 334-343
7. Ng, H.K., Leong, H.W.: Multi-Point Range Queries for Large Spatial Databases. The Third IASTED International Conference on Advances in Computer Science and Technology (2007)

8. Bertone, P., Gerstein, M.: Integrative data mining: the new direction in bioinformatics. IEEE Engineering in Medicine and Biology Magazine 20 (2001) 33-40
9. Garcia, Y.J., Lopez, M.A., Leutenegger, S.T.: A Greedy Algorithm for Bulk Loading R-Trees. Proceedings of 6th ACM Symposium on Geographic Information Systems (ACM-GIS) (1998) 163-164
10. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Wheeler, D.L.: GenBank. Nucleic Acids Research 34 (2006) D21-D24
11. Bairoch, A., Apweiler, R., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Natale, D.A., O'Donovan, Redaschi, N., Yeh, L.S.: The Universal Protein Resource (UniProt). Nucleic Acids Research 33 (2005) D154-D159
12. http://www.ncbi.nlm.nih.gov/blast/producttable.shtml#pstab.
13. Gish, W., States, D.J.: Identification of protein coding regions by database similarity search. Nature Genetics 3 (1993) 266-272
14. Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J.: SOM_PAK: The Self-Organizing Map Program Package. Technical Report A31 (1996) FIN-02150 Espoo, Finland.

Anomaly Intrusion Detection Based on Dynamic Cluster Updating

Sang-Hyun Oh and Won-Suk Lee

Department of Computer Science, Yonsei University
{osh, leewo}@database.yonsei.ac.kr

Abstract. For the effective detection of various intrusion methods into a computer, most of previous studies have been focused on the development of misuse-based intrusion detection methods. Recently, the works related to anomaly-based intrusion detection have attracted considerable attention because the anomaly detection technique can handle previously unknown intrusion methods effectively. However, most of them assume that the normal behavior of a user is fixed. Due to this reason, the new activities of the user may be regarded as anomalous events. In this paper, a new anomaly detection method based on an incremental clustering algorithm is proposed. To adaptively model the normal behavior of a user, the new profile of the user is effectively merged to the old one whenever new user transactions are added to the original data set.

Keywords: Intrusion Detection, Anomaly Detection, Data Mining, Clustering.

1 Introduction

Due to the advance of computer and communication technologies, damages caused by the unexpected intrusions and crimes related to computers have been increased rapidly. Previously, most of the intrusion methods were very primitive and simple. However, they have been changed into more complicated forms, and eventually, they have turned to some kinds of new advanced intrusion methods. As a result, it is not enough to preserve security just handling the known intrusion methods individually.

The anomaly detection model focuses on the modeling the regular patterns of normal user activities. In anomaly detection, if a new user activity deviates from his/her normal behavior pattern, those can be regarded as a possible intrusion. Typical conventional anomaly detection researches [1, 2, 3] have used statistical approaches. The statistical methods have the strong point that the size of a profile for real-time intrusion detection can be minimized. However, the contents of a profile are rough to represent its precise characteristics since only statistical operators like average and standard deviation are employed. As a result, anomalous user activities can be incorrectly detected. Furthermore, the statistical methods cannot handle infrequent but periodically occurring activities.

In this paper, a new anomaly detection method is proposed by clustering the considerable amount of transactional audit data. The proposed method is explained by a density-based clustering algorithm DBSCAN [4]. While the number of data objects

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 737–744, 2007.

in a specific range of a domain is an important criterion for creating a cluster in conventional clustering algorithms, the number of distinct transactions i.e., the transaction support of data objects in a specific range is an important criterion in the proposed method. Therefore, not only the regular patterns of user normal activities but also infrequent activities can be effectively modeled by the proposed method. Meanwhile, when new transactions occur, their normal activities should be effectively reflected to the set of currently identified clusters. For this purpose, clustering should be performed on not only the set of new transactions but also all the previous transactions. Alternatively, an incremental clustering algorithm should be employed. These two methods need to maintain and process the set of all previous transactions. To avoid this, clustering can be only performed on the newly occurring transactions to produce collected data, and the clusters for total data are updated by the comparison between the previous clusters and the current clusters generated by using the newly collected data.

This paper is organized as follows. In Section 2, a method of clustering the activities of transactions generated by a user is explained. In Section 3, how to update a profile incrementally is described. In Section 4, a method of detecting anomalous behavior is described. In Section 5, an anomaly detection method based on the proposed method presents and the performance of the proposed method are comparatively evaluated and discussed. Finally, this paper is concluded in Section 6.

2 Clustering User Activities

In this paper, the activities of a user are modeled by clustering similar activities based on various features. Each feature can be such as CPU usage, file access frequency and system call frequency. After clustering is performed on each feature, the abnormality of a new user activity can be identified by the clusters of past user activities. DBSCAN [4] is one of the most popular clustering algorithms and it finds the groups of similar data objects in a plain collection of data objects. The similarity among data objects is defined by a predefined clustering range λ. In other words, two data objects are similar if the difference between their values is within λ. Consequently, they can be contained in a same cluster. In addition, in order to identify the normal behavior of a user, a transaction support is considered additionally. The transaction support of a data group is defined by the ratio of the number of distinct transactions contained in the group over the total number of transactions. If the transaction support of the data group is greater than or equal to a certain number of distinct transactions i.e. minimum support s_{min}, the group can be a cluster. Due to this mechanism, similar activities performed in a considerable number of distinct transactions are modeled as normal behavior [5].

Let TS denote a set of transactions occurring previously i.e., TS = $\{T_1, T_2, \ldots, T_n\}$. Let a_{ij} denote the j^{th} activity in the i^{th} transaction T_i ($1 \le i \le n$) and let $v_k(a_{ij})$ denote the k^{th} feature value for the activity a_{ij}. Let D denote a set of all activities occurring until now i.e., $D = \bigcup_{i=1}^{n} T_i$. For the k^{th} feature and an activity a, the similar group $G^k_\lambda(a, D)$ with respect to a predefined clustering range λ is defined as follows:

$$G^k_\lambda(a, D) = \{a' \mid v_k(a) - \lambda \le v_k(a') \le v_k(a) + \lambda,\ a' \in D\}$$

For the k[th] feature, the transaction support sup(G^k_λ(a, D)) of the similar group for an activity a is formally defined as follows:

Definition 1. Similar group support
For the k[th] feature, the transaction support of the similar group G^k_λ(a, D) for an activity a is represented as follows:

$$sup(G^k_\lambda(a,D)) = \frac{1}{n} \cdot \sum_{i=1}^{n} I(T_i, G^k_\lambda(a,D)) \qquad I(T_i, G^k_\lambda(a,D)) = \begin{cases} 1 \ if \ T_i \cap G^k_\lambda(a,D) \neq \varphi \\ 0 \ otherwise \end{cases} \quad \square$$

Given the predefined values of a minimum support and a clustering range, clustering process is performed as follows. If the support of a neighbor set is higher than or equal to the specified minimum support, the neighbor set becomes a new cluster; otherwise, it is considered as noise. In this algorithm, to denote the on-going clustering state of each data object, a state marker is associated with each data object. The state of an object is denoted by three states: *unclassified, noise* and *a cluster identifier*. Initially, all data objects in *D* are unclassified but the state of an object is fixed to either noise or a cluster identifier that it belongs to. The detailed steps of the clustering algorithm are described as follows.

[Step 1] Sort the data set *D* of the k[th] feature by an ascending order.
[Step 2] Choose the smallest unclassified data object a_{ij} in the sorted data set *D*. If there is no such a data object, terminate.
[Step 3] Retrieve $G^k_\lambda(a_{ij}, D)$ from the data set *D*, and calculate the support of $G^k_\lambda(a_{ij}, D)$ and compare it with s_{min}. If the support is less than s_{min}, make the state of the data object a_{ij} be noise and go to Step 2. Otherwise, mark the states of all objects in $G^k_\lambda(a_{ij}, D)$ with a new cluster-id, and push them to the stack except a_{ij}.
[Step 4] Read a data object *current* from the top of the stack, and then retrieve $G^k_\lambda(current, D)$ from this object. If the support of $G^k_\lambda(current, D)$ is greater than s_{min}, mark the states of unclassified or noise objects in $G^k_\lambda(current, D)$ *with* the current cluster-id.
[Step 5] If the stack becomes empty, go to Step 2. Otherwise, go to Step 4.

3 Updating a Profile

As new sessions of a user take place, new transactions corresponding to the sessions need to be reflected to the set of current clusters. In other words, the activities of the newly occurring transactions may contain some new activities that are different from the old activities. Therefore, the current profile that has summarized the old activities should be updated accordingly. One simple way of reflecting is that clustering is performed on the total data containing the previous data and the current collected data. However, it includes two undesirable drawbacks. In other words, the previous clustering information is useless and the clustering cost becomes too expensive as the collected data increases. For solving these problems, clustering is only performed on the newly collected data, and the profile of total data is updated by the comparison between the previous profile and the current profile generated by using the newly collected data. Let D_O and D_N denote the set of previous transactions and a set of

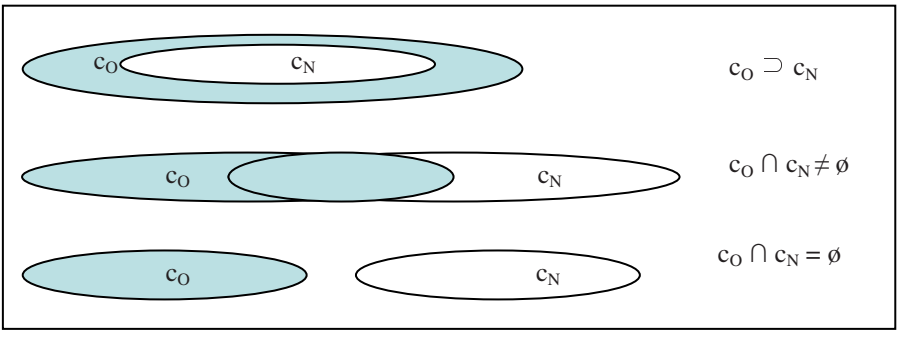


Fig. 1. Three cases of $c_O \in C(D_O)$ and $c_N \in C(D_N)$

newly occurring transactions, respectively. The entire data set D_T is $D_T = D_O \cup D_N$. Let $C(D)$ denote a set of clusters identified in a data D. There are three different cases of two clusters $c_O \in C(D_O)$ and $c_N \in C(D_N)$ are illustrated in the Figure 3.

Lemma 1. The intersection between two clusters $c_O \in C(D_O)$ and $c_N \in C(D_N)$ is contained in the updated profile.

(Proof) Let $G_\lambda(x, D)$ and $sup(G_\lambda(x, D))$ be the similar data group of x and its support, respectively. For any $x \in c_O \cap c_N$, $sup(G_\lambda(x, D_O)) \geq |D_O| \cdot s_{min}$ and $sup(G_\lambda(x, D_N)) \geq |D_N| \cdot s_{min}$. By $sup(G_\lambda(x, D_T)) = sup(G_\lambda(x, D_O)) + sup(G_\lambda(x, D_N))$, $sup(G_\lambda(x, D_O)) + sup(G_\lambda(x, D_N)) \geq |D_O| \cdot s_{min} + |D_N| \cdot s_{min} = |D_T| \cdot s_{min}$. Hence, $c_O \cap c_N$ is contained in the updated profile. ■

Lemma 2. The data not contained in both of the previous profile $C(D_O)$ and cluster set $C(D_N)$ is not contained in the updated profile $C(D_T)$.

(Proof) Let D_O-$C(D_O)$ and D_N-$C(D_N)$ be the area which cannot generate any cluster in historical data and in recent data, respectively. For any $x \in (D_O - C(D_O)) \cup (D_N - C(D_N))$, $sup(G_\lambda(x, D_O)) < |D_O| \cdot s_{min}$ and, $sup(G_\lambda(x, D_N)) < |D_N| \cdot s_{min}$. As $sup(G_\lambda(x, D_T)) < |D_T| \cdot s_{min}$, any cluster is not generated in $(D_O - C(D_O)) \cup (D_N - C(D_N))$. ■

In Figure 1, $c_O \cap c_N$ is contained in the updated profile by Lemma 1 and, $(c_O \cup c_N)^c$ is not contained in that by Lemma 2. Therefore, the data contained in $c_O \cap c_N$ and $(c_O \cup c_N)^c$ is not necessary to be accessed in the clustering process. However, the data contained in $c_O^c \cap c_N$ and $c_O \cap d_N^c$ needs to be accessed because any cluster can be existed. In this case, $c_O^c \cap c_N$ is mined in D_O by a cluster containing $c_O \cap c_N$. On the other hand, $c_O \cap d_N^c$ is mined in D_O by a cluster $c_O \cap c_N$. The cluster updating algorithm is as follows.

[Step1] The clustering is performed using the newly collected data D_N.
[Step2] The intersection between the previous profile $C(D_O)$ and the cluster set $C(D_N)$ from D_N is inserted to $C(D_T)$.

[Step3] Select c_i contained in $C(D_T)$ and expand it forward or backward with aforementioned expanding method. During the expansion of a cluster c_i, if $c_i \cap c_j \neq \varphi$ ($c_j \in C(D_T)$), c_i and c_j are merged.

[Step4] If there is no cluster to be expanded, procedure terminates.

The performance of profile updating is a very critical issue. If the historical data D_O and the recent data D_N are similar, the searching time for the original data will be very small. On the other hand, if D_O and D_N are different, the searching time will be very large.

4 Detection of User Anomaly

A user profile maintains a set of clusters for each feature generated by the proposed clustering algorithm. A cluster in a profile is represented by a tuple (average, support, length). The average of a cluster is the average value of all the data objects that are within the range of the cluster and the support of a cluster represents its actual support i.e., the ratio of distinct transactions whose data objects lie in the range of the cluster over the total number of transactions. Finally, the length of a cluster is represented by the minimum and maximum values of the cluster.

The transaction support of a cluster is important to identify the abnormality of the activities of a user. When an abnormal activity is detected between two adjacent clusters of the same feature, the abnormality of this activity is measured by the relative distance between the clusters as follows.

Definition 2. Relative Distance
Give a cluster C of a specific feature, the relative distance R(C, v) to the corresponding feature value v of a user activity is defined as follows.

$$R(C, v) = \begin{cases} \dfrac{\sup(C) \cdot |\, avg(C) - v \,|}{length(C)} & \text{if } length(C) \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad \square$$

In order to detect an anomaly for the online activities of a user, when a user logs in, his or her profile is pre-fetched to memory and the subsequent online activities of the user are monitored by evaluating the abnormality between the online activities and their corresponding clusters of related feature in the profile. The degree of abnormality increases as the difference between the current user activities and the profile becomes larger. For an online activity of a user, the gross abnormal degree of all the features related to the activity is defined by Definition 3.

Definition 3. Degree of Abnormality
Let p denote the number of features related to an activity a and C_k denote the closest cluster of the k^{th} feature defined by Definition 3 for the corresponding feature value $v_k(a)$ of the online activity. The degree of abnormality $\rho(a)$ for an activity a is defined as

$$\rho(a) = \frac{1}{p} \cdot \sum_{k=1}^{p} R(C_k, v_k(a)) \qquad \square$$

In order to decide the rate of abnormal behavior in the new object o, a set of different abnormality levels can be defined relatively to the normal behavior of the historical activities. In this paper, two different abnormality levels (*green, red*) are considered in order to classify whether the activities of a new object are anomalous or not. The green level is safe while the red is warning. Let n denote the total number of transactions in a data set TS. The range of each level is set based on an average abnormality Φ(TS) and its standard deviation *sd* for the past transactions of a user as follows.

$$\Phi(TS) = \frac{1}{n} \cdot \sum_{i=1}^{n} \frac{1}{|T_i|} \cdot \sum_{j=1}^{|T_i|} \rho(a_{ij}) \quad sd = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} \left(\frac{1}{|T_i|} \cdot \sum_{j=1}^{|T_i|} \rho(a_{ij}) \right)^2 - \Phi(TS)^2}$$

As a result, the abnormality of a specific online activity is set to one of the two levels as follows:

green level: $0 \le \rho(a) \le \Phi(TS)$ ·red level: $\Phi(TS) + sd < \rho(a)$

5 Experimental Results

In order to evaluate the performance of the proposed algorithm in a real world environment, we use DARPA log data sets collected in 1998 [6]. The feature values of the log data sets are extracted by BSM (Basic Security Module) [7] of Solaris 2.6. Among these signals, 84 signals are used as basic features in the experiments. In a log data set, an object is defined by the number of system calls occurring in a unix command on a host computer. We use two types of data sets for real world experiment: a programmer and a system administrator. In this experiment, the programmer is regarded as a target user for anomaly detection. To simulate the environment of each data stream, a data set is replicated multiple times and its transactions are looked up one by one in sequence.

In Figure 2, the execution time of the proposed method is compared with that of the original DBSCAN method when the number of transactions is varied. In this experiment, the minimum support s_{min} is set to 50% and the clustering range is set to 4. Also, 500,000 transactions are used for generate an initial profile. When the number of new transactions is varied from 1,000 through 500,000, the execution time of the proposed method is very lower than that of the DBSCAN method.

Figure 3 shows the experimental results of the anomaly detection. The anomalies using the data generated with the normal user activity profile are shown in Figure 3(a), and the anomalies using intrusion data are shown in Figure 3(b). In Figure 3(a), although the user activities were normal with 90% of the *minimum support*, the degree of user anomalies is very high. In other words, as the *minimum support* increases, the number of the generated clusters decreases and the degree of user anomalies for various activities increases. However, the degree of anomaly of an intruder should be high. Nevertheless, if the clustering range is greater than 32, the degree of the anomaly decreases drastically as shown in Figure 3(b). The reason is

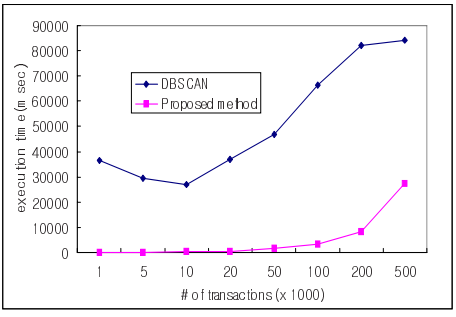
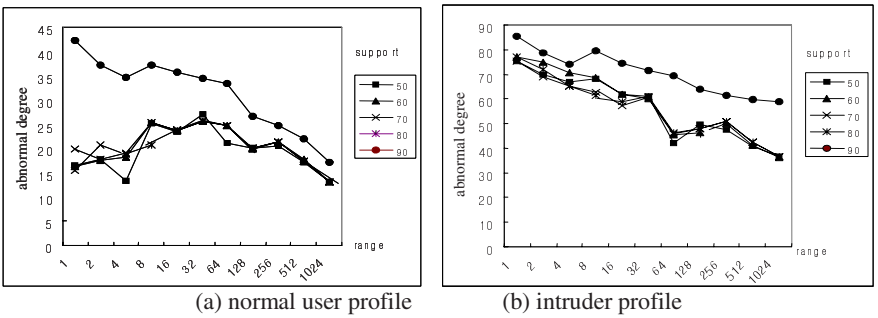


Fig. 2. Comparison between DBSCAN and the proposed method



(a) normal user profile (b) intruder profile

Fig. 3. Degree of the anomaly with the variation of clustering criteria

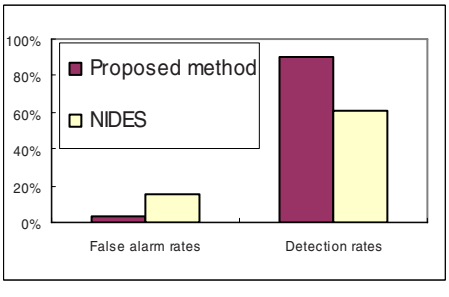


Fig. 4. Detection results

that as a clustering range increases, the activities of an intruder can be close to the range. Therefore, if *clustering range* is set under 32 and *minimum support* is set under 90%, the degree of the anomalies can be optimized.

In Figure 4, the false alarm and detection rates in the proposed method are compared with those of NIDES. As shown in Figure 4, the false alarm rate of NIDES is higher than that of the proposed method. Furthermore, the detection rate of NIDES is much lower than that of the proposed method. As a result, the proposed method can detect an anomaly more effectively than NIDES.

6 Conclusions

For the host-based intrusion detection, most of the previous approaches have been focused on the statistical techniques. But, with the statistical techniques, the anomalies can be detected incorrectly because it depends on the average values for the analysis of user activities. And it also has a drawback in analyzing infrequent user activities effectively. To handle this problem, we propose a new anomaly detection method is proposed by clustering the considerable amount of transactional audit data. Especially, abnormal activities of a user can be analyzed in various aspects. That is to say, these activities can be classified by many different types of features and for each feature, normal user patterns can be generated using the proposed clustering algorithm. In addition, when new transactions are added to the original data, the previous profile can be updated efficiently based on dynamic cluster updating. With the evaluation results, using the normal user patterns generated from the proposed scheme, we show that the user anomalies can be detected more easily and effectively than the previous statistical method.

Acknowledgement

This work was financially supported by the Ministry Of Education Human Resources Development(MOE), the Ministry of Commerce, Industry(MOCIE) and Energy and the Ministry of Labor(MOLAB) through the fostering project of the Lab of Excellency.

References

1. Harold S.Javitz and Alfonso Valdes, "The NIDES Statistical Component Description and Justification," Annual report, SRI International, 333 Ravenwood Avenue, Menlo Park, CA 94025, March 1994.
2. Phillip A. Porras and Peter G. Neumann, "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th NISSC, October 1997.
3. H.S. Javitz, A. Valdes, "The SRI IDES Statistical Anomaly Detector," In Proc. of the 1991 IEEE Symposium on Research in Security and Privacy, May 1991.
4. Martin Ester, Hans-Peter Kriegel, Sander, Michael Wimmer, Xiaowei Xu, "Incremental Clustering for Mining in a Data Warehousing Environment", Proceedings of the 24th VLDB Conference, New York, USA, 1998.
5. Sang-Huyn Oh Won-Suk Lee, “A Clustering-Based Anomaly Intrusion Detector for a Host Computer”, IEICE Trans. on Information and Systems Vol.E87-D No.8 pp.2086-2094
6. Sun Microsystems. SunShield Basic Security Module Guide.
7. http://www.ll.mit.edu/IST/ideval/index.html

Efficiently Mining Closed Constrained Frequent Ordered Subtrees by Using Border Information

Tomonobu Ozaki and Takenao Ohkawa

Graduate School of Science and Technology, Kobe University
1-1 Rokkodai, Nada, Kobe, 657-8501, Japan
{tozaki@cs., ohkawa@}kobe-u.ac.jp
http://www25.cs.kobe-u.ac.jp/

Abstract. In this paper, in order to alleviate the problem that frequent subtree miners often discover huge number of patterns, we propose two algorithms for discovering closed ordered subtrees under anti-monotone constraints about the structure of patterns. The proposed algorithms discover closed constrained subtrees by utilizing the pruning based on the occurrence matching and border patterns effectively. Experimental results show the effectiveness of the proposed algorithms.

1 Introduction

Frequent pattern miners often discover unmanageable number of patterns. To overcome this problem, several algorithms for mining closed patterns, *e.g.*[6,3], as well as constrained patterns, *e.g.*[4,5], have been proposed. Although we can expect to obtain more sophisticated and powerful miners by combining these two approaches, no such algorithm for structured data mining is proposed as far as the authors know. In this paper, in order to provide such kind of tools for mining in tree-structured databases, we propose two algorithms for discovering closed induced ordered subtrees under anti-monotone constraints about the shape of patterns such as maximum size and maximum height. The proposed algorithms discover closed constrained subtrees not by post-processing but by the search with the pruning based on the occurrence-match and patterns on the border.

The *support* of an ordered subtree t in a database of ordered subtrees D is defined as $sup_D(t) = |\{s \in D \,|\, t \preceq s\}|/|D|$ where $t \preceq s$ denotes that t is an induced subtree of s. A constraint C is called *anti-monotone* if $\forall t' \preceq t\, C(t) \rightarrow C(t')$ holds where $C(t)$ means that t satisfies C. A subtree t is called *closed constrained frequent ordered subtree*, or *closed constrained subtree* in short, if $sup_D(t) \geq \sigma$, $C(t)$ and $\nexists s \succ t\,(C(s) \wedge sup_D(t) = sup_D(s))$ hold. The problem we treat in this paper is formally defined as follows: given D, C and σ, find all closed constrained subtrees.

This paper is organized as follows. In section2, as the basis of discussion, we show a naive algorithm for mining closed constrained subtrees. In section3, two sophisticated algorithms having the pruning capability by using border patterns are proposed. Experimental results with synthesized and real world datasets are

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 745–752, 2007.

Algorithm NaiveCCLOOT(D, σ, C)
1: **for** $l \in \mathcal{L}$ in order of $\leq_{\mathcal{L}}$
2: $t := (0, l)$
3: **if** $C(t) \wedge sup_D(t) \geq \sigma$ **then**
4: Check(t, D, σ, C)

Procedure Check(t, D, σ, C)
1: **if** $TM_{D,C}(t) = \emptyset$ **then** output(t)
2: Refine(t, D, σ, C)

Procedure Refine(t, D, σ, C)
1: **for** $d \in \{d(\mathrm{rml}(t)) + 1, \cdots, 1\}$
2: **for** $l \in \mathcal{L}$ in order of $\leq_{\mathcal{L}}$
3: $t' := t(d, l)$
4: **if** $C(t') \wedge sup_D(t') \geq \sigma$ **then**
5: Check(t', D, σ, C)

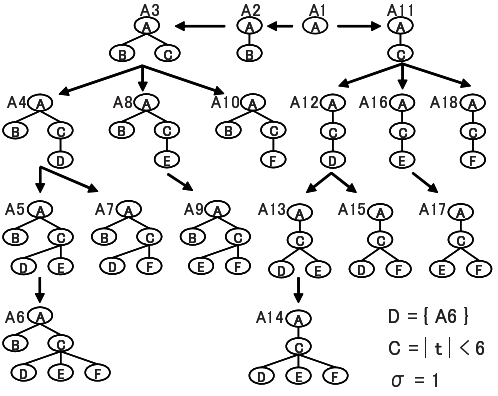


Fig. 1. A naive algorithm for mining closed constrained subtrees (left) and an example of search process (right): A subtree A_i is enumerated before $A_j (i < j)$

shown in section4. Finally, we conclude the paper and describe future work in section5.

2 A Naive Algorithm for Mining Closed Constrained Subtrees

Let $\mathcal{L}$ be a finite set of labels, on which a total order $\leq_{\mathcal{L}}$ is given. The size of an ordered subtree t on $\mathcal{L}$, denoted as $|t|$, is defined as the number of nodes in t. The label of a node v is denoted as $l(v)$. The depth of v, denoted as $d(v)$, is defined as the number of edges from the root to v. A *depth-label sequence* of t is defined as $S(t) = (d(v_1), l(v_1)) \cdots (d(v_{|t|}), l(v_{|t|})) \in (\mathrm{N} \times \mathcal{L})^{|t|}$ where $v_1, \cdots, v_{|t|}$ is the list of nodes obtained by pre-order traversal of t[2]. Since there exists one-to-one relation between a tree and its depth-label sequence, we use t and $S(t)$ interchangeably. Given two depth-label sequences $S(t_1)=(d(v_1^1), l(v_1^1)) \cdots (d(v_{|t_1|}^1), l(v_{|t_1|}^1))$ and $S(t_2)=(d(v_1^2), l(v_1^2)) \cdots (d(v_{|t_2|}^2), l(v_{|t_2|}^2))$, we denote $t_1 <_{lex} t_2$ if one of the following condition holds. Furthermore, if $t_1 = t_2$ or the third condition holds, we say that t_1 is a prefix of t_2 and denote it as $\mathrm{prefix}(t_1, t_2)$.

1. $\exists k \; s.t. \; \forall i_{(1 \leq i < k)} (d(v_i^1) = d(v_i^2) \wedge l(v_i^1) = l(v_i^2)) \wedge d(v_k^1) > d(v_k^2)$
2. $\exists k \; s.t. \; \forall i_{(1 \leq i < k)} (d(v_i^1) = d(v_i^2) \wedge l(v_i^1) = l(v_i^2)) \wedge d(v_k^1) = d(v_k^2) \wedge l(v_k^1) <_{\mathcal{L}} l(v_k^2)$
3. $\forall i_{(1 \leq i \leq |t_1|)} (d(v_i^1) = d(v_i^2) \wedge l(v_i^1) = l(v_i^2)) \; \wedge \; |t_1| < |t_2|$

All closed constrained subtrees can be mined by combining the *rightmost expansion*[1] and the closedness check by *transaction-match*[3]. *Rightmost expansion* is an extension of an efficient set enumeration technique. Each time a subtree t is enumerated, it is expanded into the set of new subtrees $\mathrm{child}(t) = \{S(t)\,(d, l) \mid 1 \leq d \leq d(\mathrm{rml}(t)) + 1, l \in \mathcal{L}\}$ where $\mathrm{rml}(t)$ denotes the rightmost leaf in t. By applying this expansion repeatedly, all subtrees can be enumerated

without duplication[1]. Note that a subtree t' s.t. prefix(t, t') will be enumerated via t. The set of *transaction-match* of t relative to a database D and an anti-monotone constraint C is defined as follows.

$$TM_{D,C}(t) = \{t' \succeq t \,|\, |t'| = |t| + 1, \forall s \in D(t \preceq s \leftrightarrow t' \preceq s), C(t')\}$$

By definition, if $C(t) \wedge TM_{D,C}(t) = \emptyset$ holds, then t is a closed constrained subtree. Note that, $TM_{D,C}(t)$ can be computed by t and its occurrences, the closedness of a pattern can be judged by itself. We show a naive algorithm named NaiveCCLOOT based on the above discussion in Fig.1(left). Note that, in this algorithm, candidate subtrees are to be enumerated in the order of $<_{lex}$.

3 Mining Closed Constrained Subtrees by Using Border Patterns

We introduce occurrence-match[3] under anti-monotone constraints for incorporating some pruning capability into NaiveCCLOOT. Two subtrees t and t' s.t. $t \preceq t'$ are said to be *occurrence-matched* if, for each occurrence of t, there exists at least one corresponding occurrence of t'[3]. The set of *left occurrence-match* of a tree t under an anti-monotone constraint C is defined as follows.

$$OM^L_{D,C}(t) = \{t' \in TM_{D,C}(t) \,|\, t' \notin child(t), t \text{ is occurrence matched with } t'\}$$

For t and $t' \in OM^L_{D,C}(t)$, $\forall x \,\text{prefix}(t, x)\, \exists x' \,\text{prefix}(t', x')\, sup_D(x) = sup_D(x')$ holds. However, unlike mining closed subtrees without constraints[3], the pruning of t s.t. $OM^L_{D,C}(t) \neq \emptyset$ causes incompleteness because $C(x')$ does not always hold even if $C(x)$ holds. For example, if A_{11} in Fig.1 is pruned based on $A_3 \in OM^L_{D,C}(A_{11})$, then a closed constrained subtree A_{14} will be never enumerated.

In order to realize the complete search with the effective pruning based on the left occurrence-match, we employ the following basic strategy. While memorizing some subtrees called *border patterns* during the search process, we apply the pruning if all subtrees which are needed to maintain the completeness can be restored by using border patterns. In the following two subsections, we show the concrete algorithms for mining closed constrained subtrees by using *positive* and *negative borders*.

3.1 Pruning by Positive Borders

A subtree t is called *positive border* relative to a database D, an anti-monotone constraint C and minimum support threshold σ, denoted as $\mathrm{B}^+_{D,\sigma,C}(t)$, iff $C(t) \wedge sup_D(t) \geq \sigma \wedge \exists t' \in child(t) \neg C(t')$ holds. Note that, if $\mathrm{B}^+_{D,\sigma,C}(t)$, then some child of t' *s.t.* $t \in OM^L_{D,C}(t')$ might be a closed constrained subtree but be pruned by the left occurrence-match. For example, A_5, A_7, A_9 and A_{14} in Fig.1 are examples of positive border and A_{14} will be pruned because $\mathrm{B}^+_{D,\sigma,C}(A_5)$, $A_5 \in OM^L_{D,C}(A_{13})$ and $A_{14} \in \text{child}(A_{13})$.

Algorithm posCCLOOT(D, σ, C)

1: Bd:= $\emptyset$
2: **for** $l \in \mathcal{L}$ in order of $\leq_{\mathcal{L}}$
3: $t := (0, l)$
4: **if** $C(t) \wedge sup_D(t) \geq \sigma$ **then**
5: Check(t, D, σ, C, Bd)

Procedure Check(t, D, σ, C, Bd)

1: **if** $\mathrm{B}^{+}_{D,\sigma,C}(t)$ **then** Bd:=$Bd \cup \{t\}$
2: **if** $OM^{L}_{D,C}(t) \neq \emptyset \wedge$
3: $\forall y \in OM^{L}_{D,C}(t)$
4: $\exists y' \in [y]^{t,+}_{D,C}$ s.t. $y' <_{lex} t$ **then**
5: T:= $\mathrm{prefix}_{min}(\bigcup_{y \in OM^{*,+}_{D,C}(t)}\{$
6: $x \setminus (y/t) \mid x \in Bd, \mathrm{prefix}(y, x)\})$
7: **for** $t' \in T$ in order of $<_{lex}$
8: **if** $\mathrm{B}^{+}_{D,\sigma,C}(t')$ **then** Bd:=$Bd \cup \{t'\}$
9: Refine(t', D, σ, C, Bd)
10: **return**
11: **if** $TM_{D,C}(t) = \emptyset$ **then** ouput(t)
12: Refine(t, D, σ, C, Bd)

Procedure Refine(t, D, σ, C, Bd)

1: **for** $d \in \{d(\mathrm{rml}(t)) + 1, \cdots, 1\}$
2: **for** $l \in \mathcal{L}$ in order of $\leq_{\mathcal{L}}$
3: $t' := t(d, l)$
4: **if** $C(t') \wedge sup_D(t') \geq \sigma$ **then**
5: Check(t', D, σ, C, Bd)

Algorithm negCCLOOT(D, σ, C)

1: Bd:= $\emptyset$
2: **for** $l \in \mathcal{L}$ in order of $\leq_{\mathcal{L}}$
3: $t := (0, l)$
4: **if** $sup_D(t) \geq \sigma$ **then**
5: Check(t, D, σ, C, Bd)

Procedure Check(t, D, σ, C, Bd)

1: **if** $\mathrm{B}^{-}_{D,\sigma,C}(t)$ **then** Bd:=$Bd \cup \{t\}$
2: **return**
3: **if** $OM^{L}_{D,C}(t) \neq \emptyset \wedge$
4: $\forall y \in OM^{L}_{D,C}(t)$
5: $\exists y' \in [y]^{t,-}_{D,C}$ s.t. $y' <_{lex} t$ **then**
6: T:= $\mathrm{prefix}_{min}(\bigcup_{y \in OM^{*,-}_{D,C}(t)}\{$
7: $x \setminus (y/t) \mid x \in Bd, \mathrm{prefix}(y, x)\})$
8: **for** $t' \in T$ in order of $<_{lex}$
9: Check(t', D, σ, C, Bd)
10: **return**
11: **if** $TM_{D,C}(t) = \emptyset$ **then** ouput(t)
12: Refine(t, D, σ, C, Bd)

Procedure Refine(t, D, σ, C, Bd)

1: **for** $d \in \{d(\mathrm{rml}(t)) + 1, \cdots, 1\}$
2: **for** $l \in \mathcal{L}$ in order of $\leq_{\mathcal{L}}$
3: $t' := t(d, l)$
4: **if** $sup_D(t') \geq \sigma$ **then**
5: Check(t', D, σ, C, Bd)

Fig. 2. The algorithms for mining closed constrained subtrees by using positive borders (left) and by using negative borders(right)

Given two subtrees x and $y \in OM^{L}_{D,C}(x)$, the set of subtrees whose child might be pruned even if it is a closed constrained subtree is defined as follows.

$$\mathrm{B}^{+}_{D,\sigma,C}(x, y) = \{y' \setminus (y/x) \,|\, \mathrm{prefix}(y, y'), \mathrm{B}^{+}_{D,\sigma,C}(y')\}$$

where y/x denotes a node v which is in y but not in x (*e.g.* $A_3/A_{11} = B$), and $t \setminus v$ denotes a tree obtained by removing v from t. For example, $\mathrm{B}^{+}_{D,\sigma,C}(A_{11}, A_3) = \{A_{13}, A_{15}, A_{17}\}$. By applying the rightmost expansion repeatedly to each element in $\mathrm{B}^{+}_{D,\sigma,C}(x, y)$, we can enumerate the subtrees which are potentially closed constrained but are pruned by the left occurrence-match. For example, A_{14} will be enumerated by the following procedure : $A_{11} \rightarrow A_3(\in OM^{L}_{D,C}(A_{11})) \rightarrow A_5(\mathrm{prefix}(A_3, A_5), B^{+}_{D,\sigma,C}(A_5)) \rightarrow A_{13}(= A_5 \setminus (A_3/A_{11})) \rightarrow A_{14}(\in \mathrm{child}(A_{13}))$.

Given a tree x, in order to guarantee the completeness, all elements in the set $\bigcup_{y \in OM^{L}_{D,C}(x)} \mathrm{B}^{+}_{D,\sigma,C}(x, y)$ have to be considered. Since the positive borders themselves are enumerated during the search process, it is necessary to compute the above set by only the borders which have been already enumerated. In NaiveCCLOOT , if $y <_{lex} x$ holds, then all element y' *s.t.* $\mathrm{prefix}(y, y')$ must be

enumerated before x. Therefore, given a tree x, the condition under which the set $\bigcup_{y \in OM^L_{D,C}(x)} \mathrm{B}^+_{D,\sigma,C}(x,y)$ can be computed is as follows.

$$\begin{array}{ll} \forall y \in OM^L_{D,C}(x) & \exists y' \in [y]^{x,+}_{D,C} \ s.t.\ y' <_{lex} x \\ \quad where & [y]^{x,+}_{D,C} = \{y' \in OM^L_{D,C}(x) \mid \mathrm{B}^+_{D,\sigma,C}(x,y) = \mathrm{B}^+_{D,\sigma,C}(x,y')\} \end{array}$$

The set $[y]^{x,+}_{C,D}$ denotes the equivalence class on the left occurrence-match of x.

From the above discussion, we show the algorithm named posCCLOOT for mining closed constrained subtrees by using positive borders in Fig.2. In this algorithm, $OM^{*,+}_{D,C}(x) = \{y \in OM^L_{D,C}(x) \mid \nexists y' \in [y]^{x,+}_{D,C}\ y' <_{lex} y\}$ denotes the set of representatives of the equivalence classes on the left occurrence-match. A function $\mathrm{prefix}_{min}(X) = \{x \in X \mid \nexists x' \in X, x \neq x', \mathrm{prefix}(x', x)\}$ is used for the avoidance of the duplicated enumeration.

3.2 Pruning by Negative Borders

The algorithm for mining closed constrained subtrees based on the negative border is quite similar to posCCLOOT.

A subtree t is called *negative border* relative to D, C and σ, denoted as $\mathrm{B}^-_{D,\sigma,C}(t)$, iff $C(t \setminus \mathrm{rml(t)}) \wedge sup_D(t) \geq \sigma \wedge \neg C(t)$ holds. If $\mathrm{B}^-_{D,\sigma,C}(t)$, then some subtree t' *s.t.* $t \in OM^L_{D,C}(t')$ might be a closed constrained subtree but be pruned. Given two subtrees x and $y \in OM^L_{D,C}(x)$, we obtain the set of potentially closed constrained subtrees which might be pruned as follows.

$$\mathrm{B}^-_{D,\sigma,C}(x,y) = \{y' \setminus (y/x) \,|\, \mathrm{prefix}(y,y'),\ \mathrm{B}^-_{D,\sigma,C}(y')\}$$

For example, A_6 in Fig.1 is an example of negative border and A_{14} will be pruned because $A_6 \in OM^L_{D,C}(A_{14})$ holds. We can restore A_{14} by the following procedure: $A_{11} \rightarrow A_3(\in OM^L_{D,C}(A_{11})) \rightarrow A_6(\mathrm{prefix}(A_3, A_6), B^-_{D,\sigma,C}(A_6)) \rightarrow A_{14}(= A_6 \setminus (A_3/A_{11}))$.

As similar to the positive borders, the set $\bigcup_{y \in OM^L_{D,C}(x)} \mathrm{B}^-_{D,\sigma,C}(x,y)$, which is required to guarantee the completeness, can be computed if the following holds.

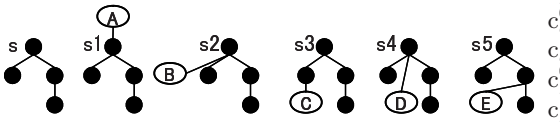
$$\begin{array}{ll} \forall y \in OM^L_{D,C}(x) & \exists y' \in [y]^{x,-}_{D,C} \ s.t.\ y' <_{lex} x \\ \quad where & [y]^{x,-}_{D,C} = \{y' \in OM^L_{D,C}(x) \mid \mathrm{B}^-_{D,\sigma,C}(x,y) = \mathrm{B}^-_{D,\sigma,C}(x,y')\} \end{array}$$

We show the algorithm named negCCLOOT for mining closed constrained subtrees by using negative borders in Fig.2. The set of representatives of the equivalence classes is defined as follows.

$$OM^{*,-}_{D,C}(x) = \{y \in OM^L_{D,C}(x) \mid \nexists y' \in [y]^{x,-}_{D,C}\ y' <_{lex} y\}$$

While $C(t)$ is checked in the procedure Refine in posCCLOOT(line 4), it is done implicitly by checking $\mathrm{B}^-_{D,\sigma,C}(t)$ in Check in negCCLOOT(line 1).

Note that, it is difficult to decide which miner should be used in advance. For instance, if we use negCCLOOT and the negative border A_6 is stored, the redundant enumeration of A_{13} can be avoided. However, the enumeration of A_6 itself might be redundant since A_6 does not satisfy the constraint.



$OM^L_{D,C}(s) = \{s_1, s_2, s_3, s_4, s_5\}$
$c^1_h = \{s_1\}$ $c^2_h = \{s_2, s_3, s_4, s_5\}$
$c^0_b = \{s_1, s_3\}$
$c^1_b = \{s_2, s_4\}$ $c^2_b = \{s_5\}$
$c^{0-1}_{k,h,b} = \{s_1\}$ $c^{0-2}_{k,h,b} = \{s_3\}$
$c^1_{k,h,b} = \{s_2, s_4\}$ $c^2_{k,h,b} = \{s_5\}$

Fig. 3. Examples of equivalence classes of the left occurrence-match

3.3 Classes of Constrained Occurrence Matching

While posCCLOOT and negCCLOOT are the general algorithms for discovering closed subtrees under anti-monotone constraints, it is necessary to implement the concrete equivalence classes of left occurrence-match for each constraint. The equivalence classes under some constraints will be explained below with examples shown in Fig.3.

Maximum size constraint: Since $\forall y \in OM^L_{D,C}(x)\ |y| = |x| + 1$ holds, all elements in $OM^L_{D,C}(x)$ belong to the same class.

Maximum height constraint: Each element $y \in OM^L_{D,C}(x)$ belongs to one of the following two classes according to the height of y denoted as $h(y)$.

$$c^1_h = \{y \in OM^L_{D,C}(x) \mid h(y) = h(x)+1\},\ \ c^2_h = \{y \in OM^L_{D,C}(x) \mid h(y) = h(x)\}$$

Maximum branching factor constraint: $OM^L_{D,C}(x)$ can be divided into the following $|\mathrm{rmb}(x)|$ classes where $\mathrm{rmb}(x)$ denotes the set of nodes on the rightmost branch of x and $b(v)$ denotes the set of all siblings of v.

$$c^0_b = \{y \in OM^L_{D,C}(x) \mid b(y/x) \cap rmb(x) = \emptyset\}$$
$$c^d_b = \{y \in OM^L_{D,C}(x) \mid y \notin c^0_b, d(y/x) = d\}\ (1 \le d \le d(\mathrm{rml}(x)))$$

If the extra node y/x in $y \in OM^L_{D,C}(x)$ does not connect to the rightmost branch, y belongs to c^0_b. Otherwise, the class of y is determined by the depth of the extra node.

Complex Constraint: The equivalence classes of left occurrence-match under the combination of constraints can be prepared by combining the classes for each constraint. We show the classes of $OM^L_{D,C}(x)$ for the combination of maximum size, height and branching factor below.

$$c^{0-1}_{k,h,b} = \{y \in c^0_b \mid y \in c^1_h\} \qquad c^{0-2}_{k,h,b} = \{y \in c^0_b \mid y \in c^2_h\}$$
$$c^d_{k,h,b} = c^d_b\ \ (1 \le d \le d(\mathrm{rml}(x)))$$

While each $c^i_{k,h,b} (1 \le i \le d(\mathrm{rml}(x)))$ is identical to c^i_b because all elements in $c^i_b (1 \le i \le d(\mathrm{rml}(x)))$ belong to c^2_h, c^0_b is further divided into two subsets by the effect of the height constraint. Note that, the maximum size constraint gives no effect because all elements belong to the same class in this constraint.

Table 1. Experimental results for D_{100}: Running Time (in second)

		max. size			max. height			max. b.f.		max. (size×height×b.f.)		
σ		7	10	15	3	5	7	2	4	7×3×2	10×5×4	15×7×4
3	P	65.3	110.1	144.7	62.0	100.3	167.1	234.2	144.2	50.7	98.6	166.1
	N	77.0	124.3	140.9	57.9	90.4	131.1	141.6	141.4	55.8	90.0	130.8
	–	90.7	179.6	315.8	64.2	134.4	241.2	250.7	324.5	49.8	131.7	243.0
1	P	123.5	248.2	382.5	118.2	232.6	468.5	713.0	363.4	92.5	219.4	460.2
	N	157.8	293.4	375.1	108.6	198.6	323.5	351.5	359.3	101.1	198.4	324.1
	–	179.3	428.3	940.1	126.8	325.0	711.8	746.0	1029.4	93.0	303.8	700.8
0.25	P	165.9	394.7	790.3	165.9	413.3	981.2	1824.7	712.2	116.4	351.8	921.4
	N	221.6	494.6	793.0	147.1	335.2	626.7	677.4	697.5	133.8	323.9	648.0
	–	239.4	683.2	2004.7	181.3	612.1	1633.2	1712.2	2574.2	118.9	490.8	1500.1

b.f. = branching factor

Table 2. Experimental results for CSLOGS

	maximum (size× height × branching factor)					
	support=0.5		support=0.3		support=0.25	
	15×5×4	15×7×6	15×5×4	15×7×6	15×5×4	15×7×6
	Running time (in second) and # of evaluated subtrees (in thousand)					
P	12.4 (38.0)	12.4 (38.0)	22.6 (117.5)	22.5 (116.8)	29.5 (205.8)	29.9 (210.3)
N	12.4 (37.8)	12.3 (37.8)	21.7 (102.6)	21.6 (102.0)	27.4 (158.8)	27.6 (162.1)
–	12.4 (38.3)	12.4 (38.3)	23.1 (131.9)	24.0 (153.7)	30.2 (231.7)	34.2 (327.2)

4 Experiments

To assess the effectiveness of the proposed algorithms, we implement three algorithms, NaiveCCLOOT, posCCLOOT and negCCLOOT, in Java and conduct experiments with the following two datasets on a PC (Pentium 4 CPU 2.80GHz) with 512Mbytes of main memory running Windows XP.

1. Synthetic dataset D_{100} generated by *Tree Generator* [7]. It consisnts of 50,000 subtrees and the average (maximum) size, height and branching factor is 10.65(143), 2.06(10) and 1.46(7), respectively.
2. Real world dataset CSLOGS which contains access trees to the website [7].

Experimental results are shown in Table 1 and Table 2. In these tables, 'P', 'N', and '–' denotes posCCLOOT, negCCLOOT and NaiveCCLOOT, respectively.

For D_{100}, with the decrease of minimum support from 3% to 0.25% gradually, we measure the running time of three miners under several single constraints as well as under the combinations of constraints. While NaiveCCLOOT runs fastest in few cases because of some overhead for searching border patterns in the proposed algorithms, posCCLOOT and negCCLOOT outperform the naive algorithm in most cases. negCCLOOT runs farster than posCCLOOT in about 70% cases. As whole, the ratio of the gain is higher when the given support is lower and the constraint is looser.

On CSLOGS, given two combinations of constraints, the running time and the number of evaluated subtrees were measured. In Table 2, each number in the parenthesis means the number of evaluated subtrees. The effects of the pruning in the proposed algorithms are confirmed because the number of evaluated subtrees decreases greatly. Furthermore, while it might be difficult to evaluate the effectiveness of the proposed algorithms because the running time is very short, the improvement of the execution time is shown in some cases. Especially, the lower minimum support and the looser constraint give a larger improvement.

From these experimental results, we can conclude that the proposed algorithms are especially effective when the search space of the problem becomes large because of the low minimum support and/or loose constraints.

5 Conclusion

In this paper, as an integration of condensed representation mining and constraint-based mining, we propose two algorithms for mining closed constrained frequent ordered subtrees. The proposed algorithms discover closed constrained subtrees by the search with the effective pruning based on the occurrence matching and border patterns.

For future work, the theoretical analysis of the proposed algorithms and further experiments with large-scale data are necessary. We also plan to apply the proposed algorithms to mining more complex structured data such as free trees and graphs.

References

1. T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Sakamoto and S. Arikawa. Efficient Substructure Discovery from Large Semi-structured Data, *Proc.* of the 2nd Annual SIAM Symposium on Data Mining, 2002.
2. T. Asai, H. Arimura, T. Uno, and S. Nakano, Discovering Frequent Substructures in Large Unordered Trees, *Proc.* the 6th International Conference on Discovery Science (DS'03), LNAI 2843, pp.47-61, 2003.
3. Y. Chi, Y. Xia, Y. Yang and R. R. Muntz. Mining Closed and Maximal Frequent Subtrees from Databases of Labeled Rooted Trees, IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.2, pp.190-202, 2005.
4. J. Pei, J. Han and W. Wang. Mining Sequential Patterns with Constraints in Large Databases, *Proc.* of the eleventh international conference on Information and knowledge management, pp.18-25, 2002.
5. C. Wang, Y. Zhu, T. Wu, W. Wang and B. Shi. Constraint-Based Graph Mining in Large Database, *Proc.* of the Seventh Asia Pacific Web Conference, pp.133-144, 2005.
6. X. Yan and J. Han. CloseGraph: Mining Closed Frequent Graph Patterns, *Proc.* of 2003 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), 2003.
7. M. J. Zaki. Efficiently Mining Frequent Trees in a Forest, *Proc.* of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.71-80, 2002.

Approximate Trace of Grid-Based Clusters over High Dimensional Data Streams

Nam Hun Park and Won Suk Lee

Department of Computer Science, Yonsei University
134 Shinchondong Seodaemungu
Seoul, 120-749, Korea
{zyonix,leewo}@database.yonsei.ac.kr

Abstract. Clustering in a large data set of high dimensionality has always been a serious challenge in the field of data mining. A good clustering method should provide flexible scalability to the number of dimensions as well as the size of a data set. We have proposed a grid-based clustering method called a *hybrid-partition method* for an on-line data stream. However, as the dimensionality of a data stream is increased, the time and space complexity of this method is increased rapidly. In this paper, a sibling list is proposed to find the clusters of a multi-dimensional data space based on the one-dimensional clusters of each dimension. Although the accuracy of identified multi-dimensional clusters may be less accurate, this one-dimensional approach can provide better scalability to the number of dimensions. This is because the one-dimensional approach requires much less memory usage than the multi-dimensional approach does. Therefore, the confined space of main memory can be more effectively utilized by the one-dimensional approach.

Keywords: Data Stream, Clustering, Grid-based Clustering, Data Mining.

1 Introduction

A data stream is defined as a massive unbounded sequence of data elements continuously generated at a rapid rate. Consequently, on-line data stream processing should satisfy the following requirements[1,2,3]. First, each data element should be examined at most once to analyze a data stream. Second, memory usage for data stream analysis should be confined finitely although new data elements are continuously generated in a data stream. Third, newly generated data elements should be processed as fast as possible to produce the up-to-date analysis result of a data stream, so that it can be instantly utilized upon request. To satisfy these requirements, data stream processing sacrifices the correctness of its analysis result by allowing some errors.

To find clusters in an on-line data stream, the k-median algorithm[4] which is the partitioning clustering method uses an O(1)-approximate k-medoid method for each sub-set of a data stream. In order to overcome the iterative evaluation of the conventional k-medoid algorithm[5], its objective is to maintain only the consistently

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 753–760, 2007.

good set of k approximate data elements ,i.e., medoids each of which represents the center of a cluster for the data elements observed so far in a data stream. Another partitioning clustering method, CluStream[6], is proposed to find the clusters of data elements generated in an evolving data stream. It executes the conventional K-means method to find initial q pseudo clusters called *micro clusters*. As a new data element arrives, the cluster features of the q micro clusters are continuously updated. The cluster feature vectors of all clusters at each specified timestamp are stored as a snapshot. The CluStream produces k final clusters called *macro clusters* by executing the K-means algorithm once more on the micro clusters of these snapshots. Although these partition-based clustering algorithm have relatively good scalability to the number of dimensions, they suffer from handling noise data elements. In other words, as in most partitioning clustering algorithms such as k-means and k-medoid, noise elements can substantially influence the generation of a cluster, so that it may be difficult to produce a correct result in some cases.

We have proposed a grid-based clustering method called a *hybrid-partition method*[7] for an on-line data stream. Since the size of each grid-cell can be different, it is impossible to employ any index structure to directly access a specific grid-cell for a specified data element. Therefore, the ranges of all the partitioned grid-cells should be sequentially examined one by one. As the number of grid-cells is increased, this operation takes longer and becomes a bottleneck for the algorithm. Furthermore, the hybrid-partition method does not have good scalability to the number of dimensions.

To cope with this shortcoming, a one-dimensional grid-based clustering algorithm is proposed in this paper. While the one-dimensional clusters of each dimension are traced independently, the clusters of multi-dimensional data space can be identified based on the one-dimensional clusters of each dimension. This one-dimensional approach can provide better scalability to the number of dimensions although the accuracy of multi-dimensional clusters may be degraded. However, this drawback is not critical since data stream processing produces approximate result anyway. Furthermore, this one-dimensional approach requires much less memory space than the multi-dimensional approach does. Therefore, the confined space of main memory can be more efficiently utilized by the one-dimensional approach. Due to these reasons, the one-dimensional approach can provide better performance in terms of time and space complexity in the grid-based clustering method.

Unlike the hybrid-partition method, the proposed method divides a dense grid-cell whose current support becomes greater than or equal to a predefined *partitioning threshold* S_{par} into h equal-size smaller grid-cells at each dimension. Conversely, a set of consecutive sparse grid-cells can be merged into a single grid-cell. Subsequently, this recursive partitioning process is terminated when the interval of every dimension of a grid-cell becomes the smallest size λ. Such a grid-cell is defined as a *unit grid-cell*. By carefully setting the value of λ to be aligned with the partitioning factor h, it is possible to make the size of every unit grid-cell be the same, so that the problem of our previous work can be avoided. A sibling list is a structure to manage the set of all grid-cells created in each dimension and it acts as an index to locate a specific grid-cell.

The remaining of this paper is organized as follows. Section 2 reviews the hybrid-partition method. Section 3 presents the structure of a sibling list and proposes a one-dimensional grid-based clustering method. In Section 4, several experiment results are

comparatively analyzed to evaluate the performance of the proposed method. Finally, Section 5 presents conclusions.

2 Preliminaries

A data stream for d-dimensional data space $\boldsymbol{N}=N_1\times\ldots\times N_d$ is defined as an infinite set of continuously generated data elements as follows:

i) A data element generated at the t^{th} turn is denoted by by $\boldsymbol{e}^t=<e_1^t,e_2^t,\ldots,e_d^t>$, $e_i^t\in N_i$, $1\le i\le$ d
ii) The current data stream $\boldsymbol{D}^{\text{t}}$ denotes all the data elements which have been generated so far ,i.e. $\boldsymbol{D}^t=\{\boldsymbol{e}^1,\boldsymbol{e}^2,\ldots,\boldsymbol{e}^t\}$.
iii) The total number of data elements generated in the current data stream $\boldsymbol{D}^{\text{t}}$ is denoted by $|\boldsymbol{D}^{\text{t}}|$. □

To find clusters over a data stream, we have proposed a statistical grid-based clustering method[7]. The range of each dimension N_i is initially partitioned by p number of mutually exclusive equal-size intervals $I_{\text{i}}^{\text{j}}=[s_i^j, f_i^j]$ $1\le j\le p$ where s_i^j and f_i^j denote the start and end values in the j^{th} interval of the i^{th} dimension. Consequently, p^d number of initial cells are formed in $\boldsymbol{N}$ and each initial cell g is defined by a set of d intervals $\{I_1,I_2,\ldots,I_{\text{d}}\}$ $I_{\text{i}}\subseteq N_i$ $1\le i\le$d. The range $R(g)$ of an initial cell g is a rectangular space $\boldsymbol{rs}=I_1\times\ldots\times I_{\text{d}}$. However, the initial rectangular space of an initial cell becomes a set of rectangular spaces $\boldsymbol{RS}=\{\boldsymbol{rs}_1,\boldsymbol{rs}_2,\ldots,\boldsymbol{rs}_{\text{q}}\}$ as a series of cell partitioning and pruning operations are performed subsequently. When these rectangular spaces are projected to the i^{th} dimension, the intervals of the i^{th} dimension of a cell g can be found and they are denoted by $IS_{\text{i}}(g)=\{I_{\text{i}}^1,I_{\text{i}}^2,\ldots,I_{\text{i}}^{\text{q}}\}$. The sum of these intervals is defined as the interval size of the i^{th} dimension of the cell g. The range of the cell g is the united spaces of all the rectangular spaces $\boldsymbol{rs}_1,\ldots,\boldsymbol{rs}_{\text{q}}$, $R(g)=\bigcup_{i=1}^{q} rs_i$. Each cell keeps the current distribution statistics of those data elements in the current data stream $\boldsymbol{D}^{\text{t}}$ that are within its range as defined in Definition 1.

[Definition 1] Distribution Statistics of a grid-cell $g(\boldsymbol{RS},c,\boldsymbol{\mu},\boldsymbol{\sigma})$
For the current data stream $\boldsymbol{D}^{\text{t}}$, a term $g(\boldsymbol{RS}, c^t, \boldsymbol{\mu}^t, \boldsymbol{\sigma}^t)$ is used to denote the distribution statistics of a cell g which is defined by a set of its rectangular spaces $\boldsymbol{RS}$. Let $\boldsymbol{D}_{\text{g}}^{\text{t}}$ denote those elements in $\boldsymbol{D}^{\text{t}}$ that are in the range of the cell g, i.e.,$\boldsymbol{D}_{\text{g}}^{\text{t}}=\{\ \boldsymbol{e}|\ \boldsymbol{e}\in\boldsymbol{D}^{\text{t}}$ and $\boldsymbol{e}\in\text{R}(g)\ \}$. The distribution statistics of the cell g are defined as follows:

i) c^t : the number of data elements in $\boldsymbol{D}_{\text{g}}^{\text{t}}$
ii) $\boldsymbol{\mu}^t=<\mu_1^t,\ldots,\mu_d^t>$: μ_i^t denotes the average of the i^{th} dimensional values of the data elements in $\boldsymbol{D}_{\text{g}}^{\text{t}}$. $\mu_i^t=\sum_{j=1}^{c^t} e_i^j / c^t$, $1\le i\le$d
iii) $\boldsymbol{\sigma}^{\text{t}}=<\sigma_1^t,\ldots,\sigma_d^t>$: σ_i^t denotes the standard deviation of the i^{th} dimensional values of the data elements in $\boldsymbol{D}_{\text{g}}^{\text{t}}$. $\sigma_i^t=\sqrt{\sum_{j=i}^{c^t}(e_i^j-\mu_i^t)^2/c^t}$, $1\le i\le$d □

As a new data element is generated continuously, each cell monitors the distribution statistics of data elements within its range. When the support of an initial cell becomes high enough, one of the dimensions of the data space is chosen as a *dividing dimension* based on the distribution statistics of data elements in the cell. Subsequently, the range of the dense cell is dynamically divided into two mutually exclusive smaller cells, called *intermediate cells*, with respect to the selected dividing dimension. In addition, the distribution statistics of the initial cell are used to estimate those of each divided cell. Similarly, when an intermediate cell itself becomes dense, it is partitioned by the same way. Differently with the initial cell, the parent intermediate cell is replaced by the divided cells when it is partitioned. Eventually, a dense region of each initial cell is recursively partitioned until it becomes the smallest cell called a *unit cell*.

To partition a dense cell, three different methods: μ-partition, σ-partition and hybrid-partition are introduced in [7]. The hybrid partition method chooses one of the two partitioning methods based on the congestion rate of the dividing dimension for a given dense cell. By selecting an appropriate partition method, the number of cell partition steps as well as the number of cells can be minimized.

3 One-Dimensional Grid-Based Clustering

The definition of a one-dimensional grid-cell is the same as in Definition 1 except it maintains scalar values rather than vectors. Given a predefined partitioning factor h, the entire data range $range(N_l)$ of the l^{th} dimension N_l ($1 \le l \le d$) is partitioned into h mutually exclusive equal-size *initial grid-cells* $G^l=\{g_1,g_2,\ldots,g_h\}$ where $\bigcup_{i=1}^{h} g_i.I \equiv range(N_l)$ and $g_i.I \cap g_j.I = \Phi$ ($i \neq j$, $1 \le i,j \le h$). In the current data stream $\boldsymbol{D}$ of a multi-dimensional data space, a grid-cell g in the dimension $\boldsymbol{N_l}$ traces the recent distribution statistics of those data elements that are in its interval $g.I$. The current support of the grid-cell g is the ratio of the number of those data elements that are inside the interval of the grid-cell over the total number of data elements in $\boldsymbol{D^t}$, i.e. $g.c/|\boldsymbol{D^t}|$. When the current support of a grid-cell becomes dense enough, i.e. greater than or equal to S_{par} ($S_{par} < S_{min}$), it is partitioned into h smaller equal-size grid-cells. Since such partitioning can be performed recursively in a dense region of the dimension N_l, the interval of each grid-cell in the dimension N_l can be different. However, among grid-cells, there exists total ordering relationship according to the interval of a grid-cell. Let $\boldsymbol{G}=\{g_1,g_2,\ldots,g_v\}$ be the set of all one-dimensional grid-cells in the dimension $\boldsymbol{N_l}$ and $\prec(g_i, g_j)$ be an ordering function i.e., $g_i(I,c,\mu,\sigma) \prec g_j(I,c,\mu,\sigma)$ *iff* $g_i.I < g_j.I$. In order to manage the dynamically varied configuration of grid-cells in the entire data space of the dimension efficiently, the grid-cells are structured by *a sibling list* defined in Definition 2.

[Definition 2] A sibling list $\boldsymbol{S}$
Given a dimension N_l of a multi-dimensional data stream, a sibling list $\boldsymbol{S}$ of order m is defined as follows;

1. A sibling list $\boldsymbol{S}=<E_1,E_2,E_3,\ldots,E_p>$ is a single linked list of sibling entries $E_1,E_2,\ldots,E_p$.
2. Each sibling entry E_i maintains a data structure $E(min,max,G[1,\ldots,m],next\_ptr)$

i) The m slots in $G[1,\ldots,m]$ can hold at most m one-dimensional grid-cells.
ii) When v ($<m$) slots are not empty, the range of the entry E_i is defined by $range(E_i)=\bigcup_{j=1}^{v} G[j].I$ and is denoted by $[min,max)$ where $min=G[1].I.s$ and $max=G[v].I.f$. In the range(E_i), $G[j].I \prec G[k].I$ $1\leq j<k\leq v$ should be true among all the one-dimensional grid-cells in G.
iii) *next_ptr* : a pointer to the next sibling entry of $\boldsymbol{S}$.

3. The range of $\boldsymbol{S}$ is defined by the union of the ranges of all the sibling entries in $\boldsymbol{S}$ and it is the same as the entire data space of the dimension N_l.

$$range(\boldsymbol{S}) \equiv \bigcup_{i=1}^{p} range(E_i) \equiv range(N_l)$$

4. Except for the first sibling entry, every sibling entry in $\boldsymbol{S}$ has at least $\lceil (m-h+2)/2 \rceil$ grid-cells at all times for a given partitioning factor h. □

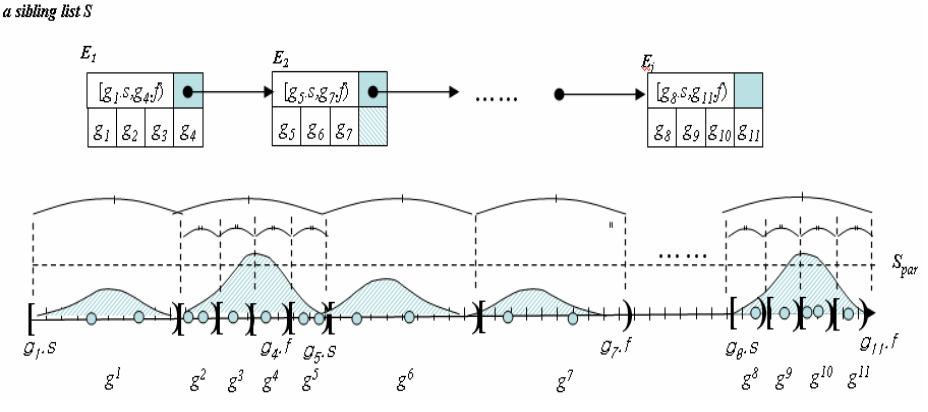


Fig. 1. A sibling list S

As shown in Figure 1, a sibling list is structured by a single linked list of sibling entries each of which can hold the distribution statistics of a fixed number of one-dimensional grid-cells. Each entry of a sibling list maintains its range [min, max) to locate a specific grid-cell efficiently. For a data element $\boldsymbol{e}$, a grid-cell g whose interval includes the data element e is searched in the sibling list. To accomplish this, the sibling entry whose range includes the element e is located first and its grid-cells are looked up in sequence. When the distribution statistics of the grid-cell g was updated lastly at the v^{th} data element in a data stream, the distribution statistics of the grid-cell g is updated for the t^{th} data element as follows:

$$g.\mu^t = (g.\mu^v \times g.c^v + e^t)/g.c^t \quad , \quad g.\sigma^t = \sqrt{g.c^v \times (g.\sigma^v)^2 / g.c^t + \{(g.\mu^v)^2 + (e^t)^2\}/g.c^t - (g.\mu^t)^2}$$

When a dense grid-cell g in a sibling list $\boldsymbol{S}$ is partitioned into h equal-size smaller grid-cells $g_1,\ldots,g_h$, the distribution statistics of each partitioned grid-cell can be more precisely monitored. The grid-cell g is replaced by the partitioned grid-cells $g_1,\ldots,g_h$, so that total ordering relationship of the grid-cells in $\boldsymbol{S}$ should be preserved. As in the hybrid-partition method, the distribution statistics of each partitioned grid-cell g_j

($1 \le j \le h$) are estimated based on those of the grid-cell g. More specifically, the distribution statistics of the j^{th} partitioned grid-cell g_j are initialized by the normal distribution function of g, $\varphi(x) = \frac{1}{\sqrt{2\pi}\, g_j.\sigma^t} e^{-\frac{(x-g_j.\mu^t)^2}{2(g_j.\sigma^t)^2}}$ as follows:

$$g_j.c^t = g.c^t \times \int_{g_j.s}^{g_j.f} \varphi(x)dx\,,\quad g_j.\mu^t = \int_{g_j.s}^{g_j.f} x\varphi(x)dx\,,\quad g_j.\sigma^t = \sqrt{\int_{g_j.s}^{g_j.f} (x\varphi(x))^2 dx - (g_j.\mu)^2)}$$

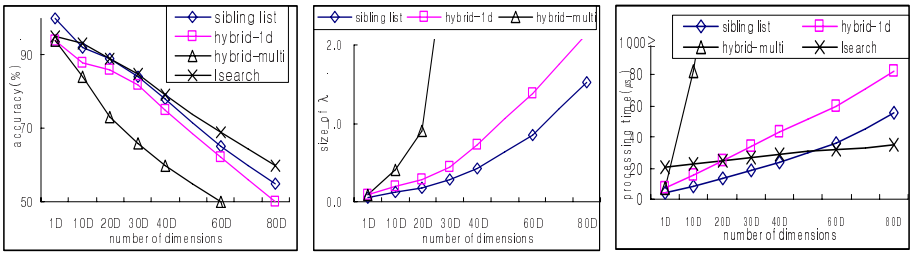
This partitioning procedure can be recursively invoked until a unit grid-cell is found. Given the range(N) of a data space of a dimension N, the total number of recursive partitioning operations needed to produce a unit one-dimensional grid-cell is $log_h(range(N)/\lambda)$. A one-dimensional cluster in the dimension N of a data stream $\boldsymbol{D}$ is a group of adjacent dense unit grid-cells whose current supports are greater than or equal to S_{min}.

While the one-dimensional clusters of each dimension are continuously traced, the set of multi-dimensional clusters at a specific time in a multi-dimensional data stream is found in off-line by the snapshot of the one-dimensional clusters of each dimension. From each sibling list $\boldsymbol{S_i}$ of each dimension $N_i(1 \le i \le d)$, its one-dimensional clusters are identified as $cluster(\boldsymbol{S_i}) = \{c_i^1, c_i^2, \ldots, c_i^p\}$. The range of a cluster $range(c_i^j)$ is denoted by two scalar values on the i^{th} dimension. Given the d-dimensions, the set of d-dimensional clusters can be found by enumerating all the combinations of the one-dimensional clusters of every dimension.

4 Experiments

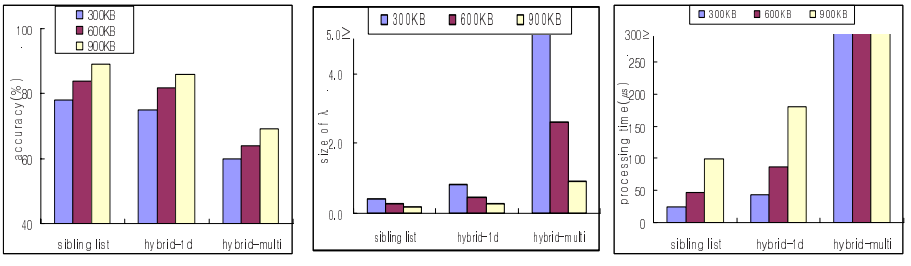
In order to analyze the performance of the proposed method, several data sets of varying dimensionality are generated by the data generator used in ENCLUS [8]. Each data set contains one million data elements. The domain size of each dimension is set to 100. Most of data elements are concentrated on randomly chosen 20 data regions whose sizes in each dimension are also randomly varied. The two support thresholds S_{mer} and S_{par} are assigned relatively to a predefined minimum support S_{min}. The conditions of most experiments are S_{min}=0.02, S_{mer}=0.1 × S_{min}, S_{par}=0.8 × S_{min}, m=10 and h=4 unless they are specified differently. Whenever 100K new data elements are processed, every sparse grid-cell is tried to be merged by traversing all the nodes of a sibling list. In all experiments, data elements are looked up one by one in sequence to simulate the environment of an on-line data stream.

In Figure 2, the proposed method is compared with the hybrid-partition method (*hybrid-1d, hybrid-multi*) and K-median algorithm[4](*lsearch)*. The term *hybrid-1d* means that the one-dimensional version of the hybrid-partition partition method is used to find only one-dimensional clusters while the term *hybrid-multi* means that the hybrid-partition method itself. The available memory space of clustering is confined to 300KB. The accuracy of the proposed algorithm is measured by the ratio of the number of correctly clustered elements by the proposed algorithm over the total number of data elements clustered by STING when λ=0.05. Let *C(i,sting)* denote a set of data elements grouped in the i^{th} cluster by STING(λ=0.05). Similarly, let *C(i,X)*



(a) Accuracy (b) The size of λ (c) Processing time

Fig. 2. Performance comparison by varying dimensionality



(a) Accuracy (b) The size of λ (c) Processing time

Fig. 3. Performance comparison by varying memory space

denote a set of data elements grouped in the same i^{th} cluster by one of the three clustering methods. The accuracy is determined as follows:

$$\text{accuracy}(\%) = \sum_i \left| C(i, sting) \cap C(i, X) \right| / \sum_i \left| C(i, sting) \right|$$

As the number of dimensions becomes larger, the accuracy of the proposed method is decreased as shown in the figure. When the dimensionality of a data stream is increased, the memory usage is rapidly increased because the number of grid-cells is also increased. Therefore, given confined memory space, to find clusters successfully, the size of λ should be enlarged, so that the number of grid-cells can be reduced. Therefore, the resolution of a grid-cell is degraded as the number of dimensions is increased. Since the memory requirements of the three grid-based methods are different, the size of λ for each method is also different. This is because the available confined memory space is fixed. The accuracy of the *hybrid-1d* method is less accurate than that of the proposed method because the range of a grid-cell is dynamically determined. The accuracy of the *hybrid-multi* method is the worst. Due to this reason, as the dimensionality is increased, its memory requirement is rapidly increased, which also makes its processing time be longer. The accuracy of LSEARCH is similar to the proposed method because the partitioning clustering method only finds the center of a cluster rather than the exact boundary of a cluster.

Figure 2-(b) shows the size variation of λ. For the same size of confined memory space, the resolution of clusters by the proposed method is more effectively maintained. As shown in Figure 2-(c), the processing time of the proposed method is better than those of the others.

Figure 3 shows the performance comparison when the size of confined memory space is varied for a 40-dimensional data stream. In Figure 3-(a), the three grid-based methods are compared in terms of accuracy. As the size of confined memory space is increased, the resolution of identified clusters becomes more precise due to the smaller size of λ as shown in Figure 3-(b). However, because the *hybrid-multi* method requires much larger memory usage than the others do, its size of λ is the largest, so that its accuracy is the worst. As shown in Figure 3-(c), the two one-dimensional approaches take less processing time than the *hybrid-multi* method does.

5 Conclusion

Since the grid-based clustering does not have good scalability to the dimensionality of a data set, the curse of dimensionality is a major challenge. A sibling list proposed in this paper can effectively maintain the on-going distribution statistics of continuously generated data elements in each dimension of a data stream. This one-dimensional approach can provide better scalability to the number of dimensions although the accuracy of identified multi-dimensional clusters may be less accurate. In other words, the one-dimensional approach requires much less memory space than the multi-dimensional approach does, so that the confined space of main memory can be more effectively utilized by the one-dimensional approach. Therefore, the one-dimensional approach can provide better performance in terms of time and space complexity.

Acknowledgments. This work was supported by the Korea Science and Engineering Foundation(KOSEF) through the National Research Lab. Program funded by the Ministry of Science and Technology (No.M10600000225-06J0000-22510).

References

1. M. Garofalakis, J. Gehrke and R. Rastogi. Querying and mining data streams: you only get one look. In the tutorial notes of the 28th Int'l Conference on Very Large Databases, Hong Kong, China, Aug. 2002
2. Mohamed Medhat Gaber, Arkady B. Zaslavsky, Shonali Krishnaswamy: Mining data streams: a review. SIGMOD Record 34(2), page 18-26, 2005
3. Joong Hyuk Chang, Won Suk Lee. Finding frequent itemsets over online data streams. Information & Software Technology 48(7), page 606-618, 2006
4. Liadan O'Callaghan, Nina Mishra, Adam Meyerson, Sudipto Guha, and Rajeev Motwani. STREAM-data algorithms for high-quality clustering. In Proc. of IEEE International Conference on Data Engineering, March 2002
5. R. O. Duda and P. E. Hart. Pattern Classification and Scene Analysis. Wiley, 1972
6. Charu C. Aggarwal, Jiawei Han, Jianyong Wang, Philip S. Yu. A Framework for Clustering Evolving Data Streams. In Proc. VLDB 29th, Berlin, 2
7. Nam Hun Park and Won Suk Lee. A statistical Grid-based Clustering over data streams. ACM SIGMOD Record, Volume 33, Issue 1, Page 32-37, 2004
8. Donald E. Knuth, The Art of Computer Programming, Addison-Wesley, volumes 1,2 and 3, 3rd edition, 1998
9. Cheng, C., Fu, A., and Zhang, Y. Entropy-based subspace clustering for mining numerical data. KDD-99, 84-93, San Diego, August 1999

BRIM: An Efficient Boundary Points Detecting Algorithm

Bao-Zhi Qiu[1], Feng Yue[1], and Jun-Yi Shen[2]

[1] School of Information & Engineering, Zhengzhou University, Zhengzhou, 450052, China
[2] School of Electronic Information & Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
bzqiu@zzu.edu.cn, yuefeng2005@tom.com

Abstract. In order to detect boundary points of clusters effectively, we propose a technique making use of a point's distribution feature of its *Eps* neighborhood to detect boundary points, and develop a boundary points detecting algorithm BRIM (an efficient Boundary points detecting algorithm). Experimental results show that BRIM can detect boundary points in noisy datasets containing clusters of different shapes and sizes effectively and efficiently.

Index Terms: Data mining, boundary points, neighborhood, density.

1 Introduction

Cluster analysis has recently become a highly hot topic in data mining research; it has been widely used in numerous applications, including data analysis, pattern recognition and image processing and so on. Up to now, many clustering algorithms have been already proposed, such as k-means, CURE[2], DBSCAN[3],CLIQUE[4] and so on. Boundary points are data points that are located at the margin of densely distributed data such as a cluster, while noises are located in the sparsely populated areas. Sometimes, boundary points are more useful and important in data mining applications because they represent a subset of population that possibly straddles two or more classes. For example, this set of points may denote a subset of population that should have developed certain diseases, but somehow they do not [1]. Special attention should be certainly warranted for this set of people since they may reveal some interesting characteristics of the disease. The knowledge of these points is also useful for data mining tasks such as classification since these points can be potentially misclassified. Boundary points analysis is so important in knowledge discovery, but there are few algorithms about it. Boundary points analysis, however, hasn't received much attention as that of clustering and outliers.

2 Related Works

DBSCAN is a density-based clustering algorithm, and defines boundary points based on density. If a point p is not a core point and it's directly density-reachable

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 761–768, 2007.

from a core point o, then the point p is called a boundary point and is added to the cluster that point o lies in. Because DBSCAN uses global density parameter, the definition of boundary points is close related to $Minpts$, that is to say, we may get different boundary points if different parameter $Minpts$ is used. What's more, the density of each cluster in the dataset is not uniform. The central density of clusters is high while the density in the border of clusters is low. Therefore it is hard to detect boundary points effectively according to the definition of boundary points in DBSCAN.

A grid-based boundary points processing technique is proposed in reference [5,6], which uses restricted k nearest neighbors and concept of relative density to recognize boundary points of clusters. The case that boundary points are located in lower density cell is merely discussed in those papers, but how to detect all the boundary points of clusters is not mentioned.

Xia et al. propose a boundary points detecting algorithm BORDER[1] which employs the special property of the reverse k-nearest neighbors, and give the formal definition of boundary points. If an object p is one of the k-nearest neighbors of object o, then o is called a reverse k-nearest neighbor of p. BORDER computes the reverse k-nearest neighbor number of each object in the first place, then data points are sorted according to their reverse k-nearest neighbor number incrementally, then select the top n objects as boundary points, because boundary points have smaller reverse k-nearest neighbor number than that of points in clusters. BORDER can detect boundary points effectively in datasets without noises. However, it still has some disadvantages: (1) The top n objects selected after sorting all the data points according to their reverse k-nearest neighbor number incrementally contain both outliers and boundary points in noisy datsets, since outliers have smaller reverse k-nearest neighbor number than that of boundary points in noisy datasets. Therefore BORDER can't correctly detect boundary points in noisy datasets. (2) The time complexity of BORDER is $\mathrm{O}(kN^2)$(N is the size of dataset) which leads to BORDER has low efficiency. (3) Given the dataset, it is hard for the users to estimate the size of boundary points n.

This paper proposes a boundary points detecting algorithm BRIM(a efficient Boundary points detecting algorithm)aiming at the disadvantages of the algorithms mentioned above, which makes use of distribution feature of Eps-neighborhood of boundary points. BRIM can detect boundary points in noisy datasets containing different shapes and sizes clusters effectively, and has higher efficiency and accuracy than BORDER.

3 BRIM Algorithms

We will give some related definitions used in our algorithm BRIM:

Definition 1. *A boundary point p is an object that satisfies the following conditions[1]:*
(1) It is within a dense region R_1.
(2) $\exists$ *region* R_2 *near p, density(*R_1*)* $\gg$ *density(*R_2*) or density(*R_1*)* $\ll$ *density(*R_2*).*

Definition 2. *Given the dataset D, the Eps-neighborhood of a point p, denoted by $N_{Eps}(p)$, is defined by[2] :*

$$N_{Eps}(p) = \{q \subseteq D \mid dist(p,q) \leq Eps\} \tag{1}$$

dist(p, q) denotes the distance between two points p, q.

Definition 3. *The density attractor of a point p. If a point o satisfies following conditions: $\forall q \subseteq N_{Eps}(p), |N_{Eps}(q)| \leq |N_{Eps}(o)|$ holds,then we call point p is attracted by o and o is density attractor of point p, denoted by Attractor(p).*

In fact the density attractor of point p is the point with maximal density in its *Eps*-neighborhood $N_{Eps}(p)$. For example in Fig.1. o is the density attractor of point p. Here we need to point out: (1) If there is no other points in $N_{Eps}(p)$ except for p itself, obviously p is a noise point, then we define the density attractor of p is p itself and let the boundary degree of p equals to minimal value. (2) If there are several density attractor in $N_{Eps}(p)$, then the point that is searched first in $N_{Eps}(p)$ is selected as the density attractor of point p.

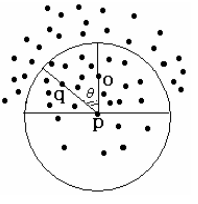


Fig. 1. The feature of boundary points

Definition 4. *Supposed o is the density attractor of a point p, the positive Eps-neighborhood of a point p, denoted by $PN_{Eps}(p)$* $PN_{Eps}(p) = \{q \subseteq D | q \subseteq N_{Eps}(p) \wedge$ *separation angle of vectors* $\overrightarrow{pq}, \overrightarrow{po} \subseteq [0^0, 90^0]\}$

Obviously p belongs to $PN_{Eps}(p)$.The Eps-neighborhood of a point $PN_{Eps}(p)$ reflects the distribution feature of the points located in $N_{Eps}(p)$ in the direction of density attractor o, as Fig.1. shows that the $PN_{Eps}(p)$ of p is the upper semicircle including o in two dimensional space.

Definition 5. *Supposed o is the density attractor of a point p, the negative Eps-neighborhood of a point p, denoted by $NN_{Eps}(p)$* $NN_{Eps}(p) = \{q \subseteq D | q \subseteq N_{Eps}(p) \bigwedge$ *separation angle of vectors* $\overrightarrow{pq}, \overrightarrow{po} \subseteq (90^0, 180^0]\} \cup \{p\}$

The $NN_{Eps}(p)$ reflects the distribution feature of the points located in $N_{Eps}(p)$ in the reverse direction of density attractor o, as Fig.1. shows that the $NN_{Eps}(p)$ of p is the lower semicircle in two dimensional space. Note that the number of points included in $NN_{Eps}(p)$ will be the denominator of the following formula computing the boundary degree of a point. However, the $NN_{Eps}(p)$ is probably

to be empty (no points in $NN_{Eps}(p)$), which brings great inconvenience to computation, so we define $p \subseteq NN_{Eps}(p)$,which avoids the case that denominator equals to zero.

Through our study of clusters and boundary points, we find that the *Eps*-neighborhood of boundary points has following distributional features: points distributed in $PN_{Eps}(p)$ are much more than points distributed in $NN_{Eps}(p)$, and the number of points distributed in the $PN_Eps(p)$ and $NN_{Eps}(p)$ is discrepant. For points in the deep of clusters, the points included in its $PN_{Eps}(p)$ and $NN_{Eps}(p)$ are not discrepant. If we define the boundary degree of a point p as the ratio of the number of points distributed in its $PN_{Eps}(p)$ and $NN_{Eps}(p)$ based on our study, then it's easy to distinguish the boundary points from points in clusters. Noises in the region of noises distributed densely also will form some "small clusters" in noisy dataset. If we define the boundary degree of a point p as the ratio of the number of points distributed in its $PN_{Eps}(p)$ and $NN_{Eps}(p)$ simply, then the boundary degree of the boundary points of the "small clusters" would be high, it's hard to distinguish the boundary points of clusters from the boundary points of the "small clusters", since both of them has great boundary degree. To solve the problem, we define the boundary degree of a point p as the ratio multiplied by the absolute value of the number of points included in $PN_{Eps}(p), NN_{Eps}(p)$.Generally speaking, the density of boundary points is greater than that of noises, so for boundary points the absolute value of the number of points included in $PN_{Eps}(p), NN_{Eps}(p)$ is also usually greater than that of noises,which greatly reduce the impact of noises on the boundary degree. The boundary degree of a point is defined as follow:

Definition 6. *$|PN_{Eps}(p)|, |NN_{Eps}(p)|$ are denoted as the number of points distributed in $PN_{Eps}(p), NN_{Eps}(p)$ respectively, then the boundary degree of a point p,denoted by $BD(p)$, is defined as:*

$$\frac{|PN_{Eps}(p)|}{|NN_{Eps}(p)|} * ||PN_{Eps}(p)| - |NN_{Eps}(p)||$$

For example in Fig.1, o is the density attractor of p, and the number of points distributed in $PN_{Eps}(p)$,$NN_{Eps}(p)$ is 21, 7 respectively, that is to say , $PN_{Eps}(p)$ =21, $NN_{Eps}(p)$ =7, so the boundary degree of point p according to the definition 6, $BD(p)$=(21/7)(21 − 7)=42

Lemma 1. *Supposed o is the density attractor of p and θ is the separation angle of vectors $(\overrightarrow{pq})$, $(\overrightarrow{po})$, If $\cos\theta \subseteq [0,1]$, then $\theta \subseteq [0^0, 90^0]$, point $q \subseteq PN_{Eps}(p)$; Otherwise if $\cos\theta \subseteq [-1,0)$, then $\theta \subseteq (90^0, 180^0]$, point $q \subseteq NN_{Eps}(p)$*

In vector space, $\cos\theta = \frac{\overrightarrow{pq} \cdot \overrightarrow{po}}{\|\overrightarrow{pq}\| \cdot \|\overrightarrow{po}\|}$, $\overrightarrow{pq} \cdot \overrightarrow{po}$ denotes the inner product of vectors and , and $\|\overrightarrow{pq}\|$ and $\|\overrightarrow{po}\|$ denotes the module of vectors and respectively.

It is so easy to prove lemma 1 that we don't depict the process in this paper. Since the value of $\|\overrightarrow{pq}\|$ and $\|\overrightarrow{po}\|$ is always non-negative, we don't need to compute whole value of $\cos\theta$. That is to say, we don't need know the exact value of θ , we only need to compute the positive(negative) of value of $\cos\theta$ in order to judge $q \subseteq NN_{Eps}(p)$ or $q \subseteq PN_{Eps}(p)$, so we only compute the range of

the inner product of vectors $\|\overrightarrow{pq}\|$ and $\|\overrightarrow{po}\|$, if $\overrightarrow{pq} \cdot \overrightarrow{po} \geq 0$, then $q \subseteq PN_{Eps}(p)$; otherwise $q \subseteq NN_{Eps}(p)$.

The main idea of BRIM is as follow: scan the whole dataset and compute the boundary degree of each point. If the boundary degree of a point p is greater than threshold δ, then define p as boundary point. We use the square of $\|\overrightarrow{pq}\|$ to measure the distance between two points p, q in the experiments.

Algorithm BRIM
Inputs: Eps, δ
Outputs: boundary points.

Step 1: select one point p unprocessed from the dataset D.
Step 2: search the Eps neighborhood of point p, and compute $Attractor(p)$.
Step 3: search the points included in the $N_{Eps}(p)$, and compute the size of $PN_{Eps}(p)$ and $NN_{Eps}(p)$ respectively,denoted by $|PN_{Eps}(p)|$ and $|NN_{Eps}(p)|$ respectively, and compute the boundary degree ($BD(p)$)of point p according to definition 6.
Step 4: If $BD(p) > \delta$, then define point p is a boundary point.
Step 5: If there are points unprocessed in the dataset D, jump to step 1.

4 Experimental Evaluations and Analysis

In the following we will evaluate BRIM from efficiency and effectiveness. Firstly, we use two dimensional synthetic datasets to verify effectiveness of our algorithm. Secondly, we use different scale datasets to verify execution efficiency of our algorithm. All experiments are run on PC with Pentium2.93G CUP, 256M memory, windows XP professional operation system. The algorithm is programmed and compiled in Visual C++6.0.

4.1 Effectiveness

In order to verify the effectiveness of algorithm, we have done experiments on many synthetic datasets (including the datasets used in the DBSCAN,Chameleon etc).But considering the limit of the paper, we choose three typical datasets to explain the effectiveness.

(1) Dataset without noises, whose geometric shape is shown in Fig.2a. The dataset has 20378 points in total and a cluster whose geometric shape looks like a five star. Fig.2b is the result of BORDER, the parameters are: the number of nearest neighbors K=50, the size of boundary points n=900; Fig.2c shows the result found by BRIM, input parameters are: the neighborhood of radius Eps=42, δ=62; From the comparison of the two figures, we can see that both BODER and BRIM can correctly detect the boundary points in dataset without noises.

(2) Dataset containing clusters with different density, whose geometric shape is shown in Fig. 3(a). The dataset has 7832 points in total, contains two clusters that are close to each other, different regions of the clusters have different

(a) Dataset1 (b) BORDER(K=50, n=900) (c) BRIM(Eps=40, δ=62)

Fig. 2. The experimental results found by BORDER and BRIM in Dataset1

densities, and noises distribute sparsely. Fig.3b is the result of BORDER, the parameters are: the number of nearest neighbors K=50, the size of boundary points n=450; Fig.3c shows the result found by BRIM, input parameters are: Eps=60, =100; From the comparison of the two figures, we can see that the boundary points found by BODER are mixed with lots of noises while BRIM can correctly detect the boundary points and distinguish boundary points from noises in dataset containing clusters with different density.

(a) Dataset2 (b) BORDER(K=40, n=450) (c) BRIM(Eps=60, δ=100)

Fig. 3. The experimental results found by BORDER and BRIM in Dataset2

(3) Dataset containing clusters with different density, size and shape, whose geometric shape is shown in Fig.4a. The dataset has 12919 points in total, contains eight clusters of different shape, size, density and orientation, as well as random noises. A particularly challenging feature of this data set is that clusters are very close to each other and they have different densities. Figure 4b shows the result found by BORDER, the parameters are: K=50, n=3000; Figure 4c shows the result found by BRIM, input parameters are: Eps=35, =42; From the comparison of the two figure, we can see from Fig.4 that the boundary points found by BODER are mixed with lots of noises while BRIM can correctly detect the boundary points.

The reason why the boundary points found by BORDER are mixed with noises in noisy datsets is that noises have smaller reverse k nearest neighbor number than that of boundary points in noisy datasets, while BRIM can correctly detect the boundary points in noisy datasets, which verify the effectiveness of our algorithm.

(a) Dataset3 (b) BORDER(K=50, n=3000) (c) BRIM(Eps=35, δ=42)

Fig. 4. The experimental results found by BORDER and BRIM in Dataset3

4.2 Time Complexity and Efficiency

The neighborhood query and computing the $PN_{Eps}(p)$, $NN_{Eps}(p)$ of each point is the most time-consuming part of our algorithm BRIM, which can be answered in $O(logN)$ time using spatial access methods such as R and SR-tree, so the time complexity of BRIM is $O(NlogN)$ if some spatial access methods are used. In the worst case, the time complexity of BRIM is $O(N^2)$, while the time complexity of BORDER is $O(kN^2)$.

Obviously the time complexity of BRIM is lower than that of BORDER. In order to verify the execution efficiency of BRIM, we have done experiments on datasets of different scales rang from 6220 to 16220 coming from Chameleon. The parameter K used in BORDER is set to 50. From the Fig.5, we can see that obviously execution efficiency of BRIM is higher than that of BORDER.

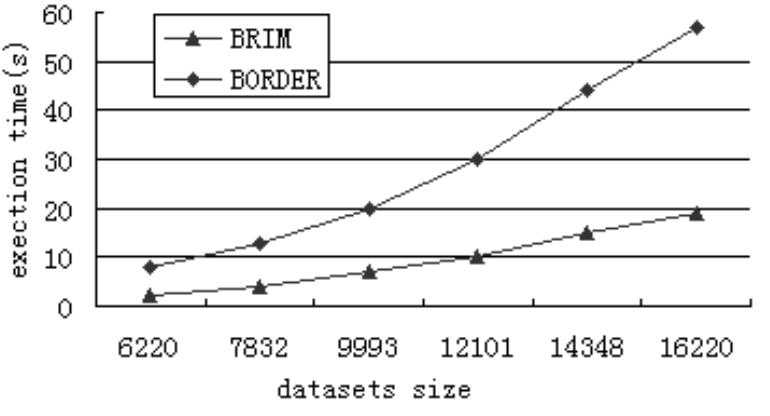


Fig. 5. Execution time of BORDER and BRIM on datasets of different scales

5 Conclusions

A boundary points detecting algorithm BRIM is proposed based on the distinct numbers of the points distributed in $PN_{Eps}(p)$ and $NN_{Eps}(p)$. BRIM can correctly detect the boundary points in dataset containing clusters with different density, size and shape with noises, and its time complexity is lower than that

of BORDER. The parameter δ has great impact on the number of resultant boundary points: the greater δ is, the smaller the number of resultant boundary points is; On the contrary, the bigger the number of resultant boundary points is. So the number of resultant boundary points is sensitive to parameter δ. It is hard for user with no prior knowledge to set proper δ. We will solve the problem and use the technique to clustering next.

References

1. Chenyi Xia, Wynne Hsu, Mong Li Lee etal. BODER:Efficient Computation of Boundary Points. IEEE transaction on knowledge and data engineering, 2006, 18(3):289-303.
2. Guha, R.Rastogi, K.Shim. CURE:an efficient clustering algorithm for large database. Proceedings of the ACM SIGMOD International Conference on Management of Data. Seattle, Washington 1998: pp.73-84. 2006(18):289-303.
3. Martin Ester, Hans-Peter Kriegel, Jörg Sander. 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). Portland, Oregon 1996: pp. 226-231. .
4. Rakesh Agrawal, Johannes Gehrke: Dimitrios Gunopulos, Prabhakar Raghavan: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Proceedings of the ACM SIGMOD International Conference on Management of Data. Seattle, Washington 1998: pp. 94-105.
5. Qiu Baozhi, Shen Junyi. a border-processing technique in grid-based clustering Pattern recognition and artificial intelligence. 2006 , 19(2): 277-280. (in Chinese)
6. Qiu Baozhi, Shen Junyi. Grid-based and Extend-based Clustering Algorithm for Multi-density. Control and decsion, 2006, 21(9): 1011-1014. (in Chinese)

Syntactic Impact on Sentence Similarity Measure in Archive-Based QA System

Guang Qiu, Jiajun Bu, Chun Chen, Peng Huang, and Keke Cai

College of Computer Science, Zhejiang University, Hangzhou, China
{qiuguang, bjj, chenc, huangp, caikeke}@zju.edu.cn

Abstract. There's now an increase in the number of Question Answering communities where large archives of question and answer pairs are collected up over time. These archives help traditional type-specified Question Answering (QA) systems to overcome type constraints and enable a service of general types. Semantic similarity measures between sentences dominate the overall performance of such Archive-based QA systems in finding similar questions in the archive to users' requests. Available approaches to sentence similarity measurement mainly utility word-to-word similarity measures directly in a bag-of-words way. In this paper, we take the syntactic evidence into account and carry out an examination on the impact of syntactic information on the sentence similarity measurement. We also compare the performance of our syntactic information incorporated approach with some baseline retrieval models. Experiments show that our approach outperforms other models both in mean average precision (MAP) and recall.

Keywords: QA system, archive, similarity measure, syntactic.

1 Introduction

There's now an emerging trend in web information service that people answer others' questions ([1]), such as Google Answers (http://answers.google.com). As time goes by, these sites possess huge number of question and answer pairs which cover various fields and hold well-formed answer texts. For example, Baidu Zhidao (http://zhidao.baidu.com), a popular Chinese web site providing this kind of service, has solved up to 11 million questions submitted by users till now (2007-1). Large archives of questions and answers are collected over time like FAQ lists. When people submit a question that has been solved already, corresponding answer can be returned immediately without waiting for manual response. Thereby, these archives can be no doubtfully ideal resources for a general Question Answering system if only similar questions could be found accurately without lag. [1] has made a first attempt to establish such a system for Korean based on translation model and declared an encouraging performance.

Performance of such systems depends heavily on the measures of sentence semantic similarity i.e. how to quantify the semantic similarity between sentences. People of different backgrounds are likely to express the same meaning with different

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 769–776, 2007.

wordings. This leads to the phenomena that two questions may have the same meaning but differ lexically. Therefore, a rigid lexical-based measure of sentence semantic similarity reduces the overall recall whereas a loose one reduces the precision.

What's more, sentence should be recognized as a sequence of words that are organized in specified structure. However, most of previous work on sentence similarity determination turns out to be bag-of-words approaches without taking the structure information of sentence into account ([2], [3], [4]). In this paper, we'll propose an approach to quantify the degree of similarity between sentences combined with syntactic knowledge, and incorporate it into a Question Answering system based on archives of questions and answers collected from Baidu Zhidao. An examination will be carried out on the impact of syntactic information on the measurement. We'll also show that our approach outperforms other known retrieval models that have been implemented in the Lemur Toolkit (http://www.lemurproject.org).

The rest of the paper is organized as follows. Section 2 illustrates our approach in details. In section 3, we describe the dataset used for our experiments and evaluation. Experimental results are given in section 4 with the evaluation. We also give a brief analysis on the results in section 4. Section 5 gives the conclusions of our work.

2 Measure Sentence Similarity

As mentioned above, most previous traditional measures of sentence (text) semantic similarity take advantage of word oriented measures just in a bag-of-words way. This kind of approach ignores lots of important information hiding in sentence structure, such as syntactic roles played by different words in the sentence ([3]). Words of different syntactic elements in a sentence make different contributions to sentence semantic similarity measurement. For example, two sentences containing same or similar subjects are more likely to be semantic similar than those only containing same or similar attributes. In other words, words that act as subject of a sentence contribute more in measuring the similarity of sentences than those as attribute. Therefore, an appropriate weighting strategy is needed to give different weights to corresponding syntactic components in the sentence to reflect the effect that the component makes to the overall measurement.

Our quantified formula for the measure is inspired by the work in [3] that used the inverse document frequency (IDF) to quantify the importance of a word. Given two sentences S_1 and S_2 of length m and n respectively, semantic similarity between them is formulated as follows:

$$sim(S_1, S_2) = \frac{(\sum_{w \in \{S_1\}} sim(w, S_2) * w(w)) / m + (\sum_{w \in \{S_2\}} sim(w, S_1) * w(w)) / n}{2} \quad (1)$$

Where $sim(w,S_2)$ is the similarity between word w and sentence S_2 measured by the maximum similarity of w and words in S_2. So it is with $sim(w,S_1)$. $w(w)$ is given according to the syntactic element of the word and guaranteed to be a value between

0.1 and 0.9. Within this formula, two critical issues have to be re-examined in details: the measure of word similarity and the weighting strategy for syntactic elements.

2.1 Measure Word Similarity

Knowledge-based methodology is adopted in our work to quantify semantic similarity between words. We take use of tongyicicilin (http://www.ir-lab.org/), a Chinese thesaurus containing 77343 terms, as the semantic network to describe word relationships. In this thesaurus, all words are first classified hierarchically into different categories according to the degree of semantic similarity: coarse-degree, medium-degree and fine-degree. Furthermore, words are categorized again into clusters and atom-clusters to gain a more fine-grained hierarchical classification. To facilitate calculation, words in the thesaurus are all encoded using numbers and letters with a length of 8. The more overlaps codes of two words contain, the more similar these two words are to each other. With this characteristic, we then define semantic similarity of two words as five levels from synonymic to unrelated by counting the overlaps of their corresponding codes.

Table 1. Word Similarity Definition (words of the same atom-cluster can be synonymic or relevant. Synonymic words are those of exactly the same meaning and can be replaced with each other while relevant words are those that are related with each other closely but are irreplaceable).

Relationship	Example
OfSameAtomCluster(Synonymic)	老人，老者，老汉，老翁 (which all mean “elder”)
OfSameAtomCluster(Relevant)	中士(sergeant)，下士(corporal)
OfSameCluster	学习者(learner)，见习员(intern)
OfSameFineDegreeCategory	学生(student)，教师(teacher)
Unrelated	人物(person)，电脑(computer)

Table 1 gives an overview of the relationships between words, explained by examples. To quantify these different levels of word similarity, corresponding values are set for each level.

2.2 Weighting Strategy

Substantial efforts have been made on syntactic parsing of natural languages, and many sophisticated parsing grammars have been proposed to describe different aspects of linguistic characteristic, such as Phrase Structure Grammar ([6]) and Dependence Grammar ([7]). It is believed that Dependence Grammar is more suitable for Chinese natural language processing ([5]). In Dependence Grammar, individual words in a sentence are considered to be linked together over dependency relations instead of being combined mechanically. The main idea of Dependence Grammar is that roles played by words of different grammar elements in a sentence are not same to each other, saying that, some words depend on others while some words govern others. In other words, relationship between words is governing or being governed.

This theory is quite similar to our weighting strategy that gives weights according to the different importance of syntactic elements.

We define totally 23 dependency elements in our Dependence Grammar parsing, and then divide these elements into 2 categories according to the contributions each makes to the similarity measurement of whole sentence. Obviously, subject, predicate and object are the skeleton of a sentence and represent the main meaning of the sentence. Therefore, if two sentences are of the same subject, predicate, or object, they tend to express the similar thing. Similarity in other grammar elements also augments whole similarity between sentences, but makes less contribution than the three ones. Contributions are then quantified based on the training corpus as two parameters (alpha and beta) in our approach. These two parameters sum to 1 and are estimated empirically.

3 Data Sets

3.1 Raw Data Set

Zhidao is one of the leading question answering service providers in China and has collected up to 11 millions question and answer pairs in Chinese till now (2007-1). This site operates like a forum in which a registered user submits his question and then any other registered user can provide his answer to the question. Thus a list of answers will always be gained in the web page, waiting for being judged. Experts of Zhidao will timely check whether there is correct answer and mark the question as solved if the correct one is found. The correct answer will also be marked out with special tags. Therefore, a straightforward web extraction technique will be capable of digging out questions and correct answers from web pages. Although answers of questions are not used during current experiments, they will be vital in building up the QA system.

Experiments conducted in our paper are based on the data set obtained from this large archive. The raw data set consists of more than 77,000 question and answer pairs over all categories and is divided into two collections: collections A and B. Collection A contains 50,000 pairs and is used to find the optimal parameter values for our approach in parameter estimation phase. Remainder pairs of the whole raw data set comprise collection B in which we carry out the evaluation of the performance of our approach with other retrieval models.

3.2 Collections with Judgment Information

Both parameter estimation and result evaluation require collections with judgment information. Therefore, raw collections A and B have to be refined for the estimation and the comparison of the performance of our proposed approach with other retrieval models. In our work, such two sets of questions with judgment information, named J_A and J_B, are constructed proportionally from collections A and B.

100 questions are selected randomly from collection A while 50 ones from collection B. These questions are regarded as queries to different retrieval models (approaches). To gather semantically relevant questions to these queries, we employ the pooling technique that is used in the TREC (http://trec.nist.gov) conference series.

We utilize the five retrieval models implemented in Lemur Toolkit to retrieve results to queries from collection A and B respectively. Only top 10 results from each retrieval model are kept as candidates. Then judgments of semantic relevancy are done among the total 50 results for one query manually, ignoring overlaps. Result (question) is considered to be semantically relevant with the query as long as they can be categorized into the same question type and are semantically identical or similar to each other.

Table 2. Question types defined in our system (already translated from Chinese)

Type	Example
HUM	Who is the current president of USA
OBJ	Which animals live only in China
LOC	Where is Zhejiang University
NUM	What is the population of India till now
TIME	When was PRC founded
DES	Why the sky is blue
Other	Is Google Answers still available

Table 2 shows the 7 question types defined in our system. In addition, to get rid of partiality generated by one judger, we employ 2 judgers to make the judgment separately and then integrate their results. Finally, we've got 335 semantically relevant questions for the 100 queries in J_A from collection A and 160 ones for the 50 queries in J_B from collection B.

4 Experiments

4.1 Parameter Estimation

The 23 dependence grammar elements are classified into two categories: skeleton elements $E_{skeleton}$ and non-skeleton elements $E_{non\text{-}skeleton}$. The former comprises of subject, predicate and object while the rest grammar elements compose the latter. Thereby, in the parameter estimation phase of our work, two parameters alpha and beta have to be estimated from J_A and J_B, which are used to quantify the importance of the elements in $E_{skeleton}$ and $E_{non\text{-}skeleton}$ respectively. As we assume that alpha and beta sum to 1, we in fact only have to estimate one parameter.

Figure 1 show the mean average precisions (MAPs) and average recalls evaluated from J_A, varying with alpha ranging from 0.1 to 0.9 (i.e. beta from 0.9 to 0.1). The results demonstrate the impact of syntactic information on the measurement of sentence semantic similarity. Specifically, our approach, when equipped with parameter alpha equal to beta (i.e. alpha=beta=0.5), can be regarded as one without considering the impact of syntactic evidence because it ignores the differentia between the importance of syntactic elements. In the graph, MAP and recall in this situation fail to reach the summits of the curves. This result indicates the necessity of considering the impact of syntactic evidence. What's more, as we can see from the

curves, situations with parameter alpha larger than beta work better in average than those with alpha smaller than beta both in MAP and average recall. This verifies our assumption that syntactic elements of $E_{skeleton}$ play a more important role in measuring sentence semantic similarity than those of $E_{non\text{-}skeleton}$. Another observation is that curves descend as alpha increases when alpha keeps larger than 0.6. It shows that as components of a sentence, elements of $E_{non\text{-}skeleton}$ also contribute more or less to the similarity measurement. Neglect of these elements degrades the overall performance. Therefore, we have to make a balance between the values of alpha and beta to give prominence to elements of $E_{skeleton}$ while still maintaining the contributions of elements of $E_{non\text{-}skeleton}$.

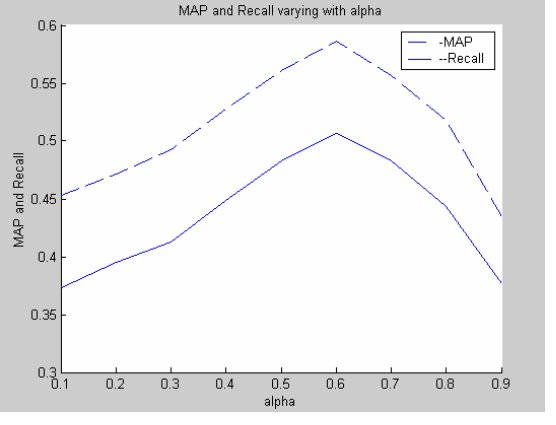


Fig. 1. MAPs and Recalls varying with alpha ranging from 0.1 to 0.9 (i.e. beta from 0.9 to 0.1). Both of these two curves reach the summits when alpha=0.6 (i.e. beta=0.4). In other words, our approach gains the best performance when alpha is set to 0.6 (i.e. beta=0.4).

4.2 Results and Evaluation

Evaluation of our approach with other models is carried out in collection J_B. Parameters alpha and beta are estimated in the phase above that are optimal in collection J_A. We use our approach to retrieve relevant questions from collection B, given the 50 queries in J_B. Results are ranked according to their semantic similarity with queries in descending order. Only top 5 results are used to evaluate the performance of the approach. Then, we compare our results with the top 5 results returned by the five baseline models implemented in the Lemur Toolkit. This toolkit is designed to facilitate researches in language modeling and information retrieval. In our experiments, the five baseline retrieval models include: the TFIDF retrieval model (TFIDF), the Okapi BM25 retrieval function (Okapi) and the KL-divergence language model based retrieval method with three different smooth algorithms (Jelinek-Mercer (KL-jm), Dirichlet prior (KL-dir) and Absolute discounting (KL-abs)).

For each of the five baseline models, their parameters are set to the optimal values in J_A. Table 3 demonstrates evaluation results of our approach (SynSim) with the five baseline models. Although both MAP and average recall are still far from being

Table 3. Evaluation Results of different models (approaches)

Models	TFIDF	Okapi	KL-abs	KL-jm	KL-dir	SynSim
MAP	0.24	0.272	0.29	0.284	0.224	0.327
Average-Recall	0.291	0.326	0.344	0.34	0.278	0.386

satisfied for a practical QA system, our approach outperforms other models. The low MAP and average recall are mostly caused by the lack of an adequate data collection which reduces the probability of encountering an existing question that is semantic similar to the query. We believe that an improvement in MAP and recall will be gained as long as a large data collection is available.

4.3 Example and Analysis

One typical example is showed in Table 4 to explain why our approach works better than other five models.

Table 4. Examples (already translated from Chinese)

Query	Which ***dynasty*** does ***Du Fu*** belong to (***杜甫***是哪个***朝代***的人)	**KL-abs**	**SynSim**
Question 1	When was ***Du Fu*** born (***杜甫***是什么时候出生的)	Rank=2	Rank=3
Question 2	Which ***dynasty*** do the four famous literatures reflect (四大名著反映的是哪个***朝代***的事)	Rank=5	Rank=10

In the example, Question 1 is considered to be semantic similar to the query while question 2 is not. The two questions both contain a word that is also in the query. However, these two words are of different grammar elements in the query. “Du Fu” acts as subject of the query and question 1 while “dynasty” acts as attribute of the query and question 2 (All these grammar elements are derived when query and questions are in Chinese). According to our weighting strategy, we regard “Du Fu” more important than “dynasty” in measuring the sentence similarity. As a result, we finally get question 1 as a result at rank 3 while ranking question 2 at 10. However, in the other 5 models, they fail to get the point and return both of these two questions in the top 5 results.

5 Conclusions and Future Work

Our work is conducted for an archive-based QA system in which measures of sentence semantic similarity dominate the overall performance when given a large archive. In this paper, we propose such a method that incorporated with syntactic information. Words in a sentence are parsed using Dependence Grammar. The syntactic elements are categorized into two classes according to their contributions to

the similarity measurement of whole sentence. A weighting strategy is introduced to quantify the contributions.

During the parameter estimation phase in our experiments, the results demonstrate the impact of syntactic information with different weights on similarity measurement and show that the consideration of syntactic information enhances the performance of a similarity measure. We also compare the performance of our approach with five baseline retrieval models implemented in the Lemur Toolkit. Experiments show that our approach outperforms other models in finding semantic similar sentences.

However, in order to incorporate our sentence similarity measure into a practical archive-based QA system, we plan to build up a larger corpus to improve the performance of our approach. We also plan to add in the information of question types to help to measure the similarity in the future.

Acknowledgments

This research was supported in part by the National Basic Research Program of China (973 Program) under Grant No. 2006CB303000.

References

1. J. Jeon, W. B. Croft, and J. Lee. Finding similar questions in large question and answer archives. In Proceedings of ACM Fourteenth Conference on Information and Knowledge Management (CIKM 2005), 84-90, 2005.
2. Li Sujian, Zhang Jian, Huang Xiong, et al. Semantic Computation in a Chinese Question-Answering System. Journal of Computer Science and Technology, 17(6): 933-939, 2002.
3. Rada Mihalcea, Courtney Corley and Carlo Strapparava. Corpus-based and Knowledge-based Measures of Text Semantic similarity. In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006), 775-780, 2006.
4. Zhang Huaping, Sun Jian, Wang Bing and Bai shuo. Computation on Sentence Semantic Distance for Novelty Detection. Journal of Computer Science and Technology, 20(3): 331-337, 2005.
5. M. Zhou, C. N. Huang. An Efficient Syntactic Tagging Toll for Corpora. In Proceedings of COLING'94, 945-955, 1994.
6. N. Chomsky. Three models for the description of language. IRI Transactions on Information Theory, 2(3): 113-124, 1956.
7. L. Tesniere. Elements de Syntaxe Structurale. Librairie C. Klincksieck, Paris, 1959.

Semi-structure Mining Method for Text Mining with a Chunk-Based Dependency Structure

Issei Sato[1] and Hiroshi Nakagawa[2]

[1] Graduate School of Information Science and Technology, The University of Tokyo
sato@r.dl.itc.u-tokyo.ac.jp
[2] Information Technology Center
nakagawa@dl.itc.u-tokyo.ac.jp

Abstract. In text mining, when we need more precise information than word frequencies such as the relationships among words, it is necessary to extract frequent patterns of words with a dependency structure in a sentence. This paper proposes a semi-structure mining method for extracting frequent patterns of words with a dependency structure from a text corpus. First, it describes the data structure representing the dependency structure. This is a tree structure in which each node has multiple items. Then, a mining algorithm for this data structure is described. Our method can extract frequent patterns that cannot be extracted by conventional methods.

1 Introduction

In text mining, when we need more precise information than a word frequencies such as the relationships among words, it is necessary to extract frequent patterns of words with a dependency structure in a sentence. In Japanese, the dependency structure is a chunk-based dependency structure, which is a dependency structure based on a chunk of words as a unit. Some examples of word chunks are verb phrases in Japanese and noun phrases and prepositional phrases in English. For example, Figure 1(a) shows the chunk-based dependency structure of "安倍総理が中国に訪れた (Japan's premier Abe went to China)". Generally speaking, a chunk-based dependency structure is represented as a tree structure in which a chunk of words is regarded as one label. This representation enables conventional semi-structure mining algorithms such as FREQT [5,6] to be used. However, the representation causes the following problem. If we want to extract frequent substructures from two trees (a) and (b) in figure 1 based on this representation schema , substructure (c) is extracted. It is a serious problem that substructure (c) has so few nodes that there is insufficient information about the relationship among words. In Figure 1, relationships such as"安倍(Abe)", "訪れた(went)"and "中国(China)"are not extracted. Consequently, patterns that have too few nodes are extracted.

To solve this problem, we propose a new data structure that represents a chunk-based dependency structure. It is a tree structure in which each node has multiple items rather than one label. The multiple items correspond to the words

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 777–784, 2007.

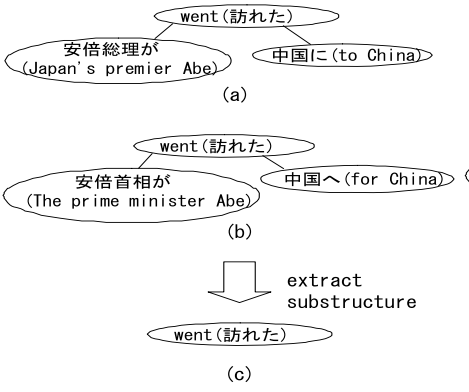


Fig. 1. The Example of Extracting Substrucure by Conventional Methods

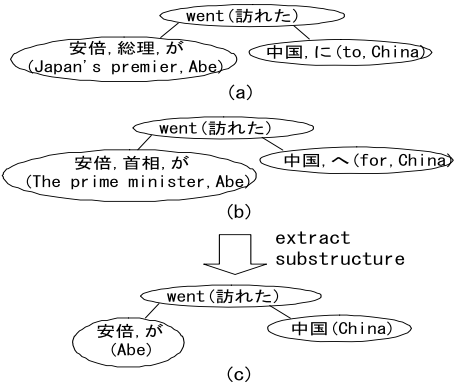


Fig. 2. Example of Extracting Substructure by Our Method

in a chunk. In addition, we propose a mining algorithm for this data structure. Our mining algorithm is an improved algorithm of the sequential pattern mining *PrefixSpan*[1,2] . An example of the data structure is shown in Figure 2. Each node has multiple items and each item is a word. In Figure 2, substructure (c) is extracted from two trees (a) and (b). The key point is that in Figure 2, it is possible to extract nodes with more items than those in Figure 1. Substructure (c) in Figure 2 has three nodes, while substructure (c) in Figure 1 has one node. Therefore, our method can extract patterns that have much more information, such as the extracted patterns shown in Figure 2(c).

The remainder of this paper is organized as follows. Sections 2 and 3 introduce, as related work, sequential pattern mining and its algorithm *PrefixSpan* and labeled orderd tree mining and its algorithm *FREQT*, respectively. Section 4 explains our method and Section 5 evaluates it. Section 6 summarizes our work.

2 Sequential Pattern Mining[1,2]

In this section ,we explain briefly Sequential Pattern Mining.

Let $I = \{i_1, i_2, \cdots, i_n\}$ be a set of items. This set of items is also called **an element**. An element is denoted by $(i_1, i_2, \cdots, i_m)$. An inclusive relation between element e_1 and element e_2 is denoted as $e_1 \subseteq e_2$ if all items of e_1 are included in e_2. Since an element is a set of items, it is necessary to sort items in an element in lexical order in advance: it is possible to consider the original order of items if we do not sort them in the element. **A sequence** is an ordered list of elements. For example, the sequence$< (a,b)(a,d) >$ is not equivalent to $< (a,d)(a,b) >$. A sequence s is denoted by $s =< e_1, e_2, \cdots, e_l >$ where e_k is an element. The number of items in a sequence is called the length of the sequence. A sequence with length L is called an L-sequence. A sequence $\alpha =< a_1, a_2, \cdots, a_n >$ is called a subsequence of another sequence $\beta =< b_1, b_2, \cdots, b_m >$ if there exist integers

$1 \leq j_1 \leq j_2 \leq \cdots \leq j_n \leq m$ such that $a_1 \subseteq b_{j_1}$,$a_2 \subseteq b_{j_2}$,$\cdots$,$a_n \subseteq b_{j_n}$ This relationship between α and β is denoted as $\alpha \sqsubseteq \beta$. **A sequence database** S is a set of tuples (sid, s), where sid is a sequence id and s is a sequence defined as follows: $S = \{(sid_1, s_1), (sid_2, s_2), \cdots, (sid_n, s_n)\}$. **The support** of a sequence α in a sequence database S is the number of tuples containing α in the database defined as follows: $support_S(\alpha) = \| \{(sid, s) \mid (sid, s) \in S \wedge \alpha \sqsubseteq s\} \|$. **A frequent sequence** α is definded as a sequence whose support is greater than the minimunm support ζ, which is a threshold, i.e., $support_S(\alpha) \geq \zeta$.

Sequential Pattern Mining is defined as the problem of extracting all frequent sequences from a sequence database.

Constraint-based Sequential Pattern Mining is Sequential Pattern Mining that uses other constraints as well as minimum support[3] [4]. Our proposed mining algorithm is also categorized into this type of constraint based sequential pattern mining.

PrefixSpan is proposed in 2000 as a fast sequential pattern mining algorithm[2]. PrefixSpan extracts frequent sequences with a depth-first search by recursively executing projection operations, which is called Prefix projection. We explain briefly Prefix projection and Prefixspan algorithm as follows . Given a sequence $s =< e_1, e_2, \cdots, e_n >$ and item a, it is assumed that there exists a positive integer m $(\leq n)$ such as $e_1 \not\ni a, e_2 \not\ni a, \cdots, e_{m-1} \not\ni a, e_m \ni a$. Moreover, suppose that $e_m = (a_1, a_2, \cdots, a_j, \cdots, a_t)$, $e_m^- = (a_1, a_2, \cdots, a_j), a_k \neq a(k < j), a_j = a$Cand $e_m^+ = (a_{j+1}, \cdots, a_t)$. A sequence$< e_1, e_2, \cdots, e_m^- >$ is defined as **the prefix** of s based on item a. A sequence$< e_m^+, e_{m+1}, \cdots, e_n >$ is defined as **the postfix** of s based on item a. If there is no m, prefix and postfix are not defined. **Prefix projection** of a sequence database S with item a is defined as the operation that constructs a projected database from the postfixes of sequences based on item a. It adds prefix '_' to items that are in the same element as a projecting item. An $< a >$-projected database S is defined as a database that is projected with item a and is denoted by $S_{<a>}$. For example, when a sequence database $S = \{(sid_1, < (c,d)(b)(a,d)(b,c,a) >), (sid_2, < (d,a)(d,a)(b,c) >)\}$ is projected with item a , the projected database $S_{<a>}$ is $\{(sid_1, < (\_d)(b,c,a) >), (sid_2, < (d,a)(b,c) >)\}$

PrefixSpan algorithm is as follows:

1. Find length 1-frequent sequences
Scan a sequence database S to find frequent items(= length 1-frequent sequences) whose supports are greater than the minimum support. Then, all frequent sequences are partitioned into subsets that have the length 1-frequent sequences as each prefix.
2. Find subsets of length $k-$frequent sequence
For $k(\geq 2)$, do the following procedure by incrementing k until frequent items can not be extracted.

Extract each subset of the $k-$frequent sequence by finding frequent items from a projected database projected corresponding to each $(k-1)-$frequent sequence.

3 Labeled Ordered Tree Mining

In this section ,we explain briefly Labeled Ordered Tree Mining. **A labeled orderd three** is a tree in which each node has one label and which keeps the order among siblings. If a labeled ordered tree α is a subset of another labeled ordered tree β, it is defined that β 'includes' α, denoting $\alpha \sqsubseteq \beta$. **A labeled ordered tree database** T is a set of tuples (tid, t), where tid is an ordered tree id and t is a labeled ordered tree, defined as s: $T = \{(tid_1, t_1), (tid_2, t_2), \cdots, (tid_n, t_n)\}$. **The support** of a labeled ordered tree α in a labeled ordered database T is the number of tuples in the database containing α defined as follows: $support_T(\alpha) = \| \{(tid, t) \mid (tid, t) \in T \wedge \alpha \sqsubseteq t\} \|$. **A frequent labeled ordered tree** α is defined as a labeled ordered tree whose support is greater than minimum support ζ, which is the threshold, i.e., $support_T(\alpha) \geq \zeta$. **Labeled Ordered Tree Mining** is defined as the problem of extracting all frequent labeled ordered trees from a labeled ordered tree database corresponding to the minmum support. The number of nodes of a labeled ordered tree t is called the size of a labeled ordered tree t and is denoted as $\mid t \mid$.

FREQT is proposed in 2002 as a fast labeled ordered tree mining algorithm[5,6]. FREQT extracts frequent ordered trees by the technique of growing a tree by attaching new nodes only on the rightmost branch of the tree, which is called rightmost expansion. Rightmost expansion was also proposed by Zaki et al.[7].

4 Proposed Method

In this section, we propose a new data structure to represent the chunk-based dependency structure and a mining algorithm for the new data structure. This is a tree structure whose node has a element,that is, a set of items. For example, a chunk such as "with a depth-first search" is represented as a node with an element including four items, i.e., ('with' , 'a' , 'depth-first','search'). In this regard, however, items in the element keep their order.

Since our data structure is not a labeled ordered tree, an existing mining algorithm such as FREQT cannot be applied directly. But, if we transform the data in the following way, we can make the sequential pattern mining algorithm applicable to our data structure.

1. Enumerate elements by traversing the data structure from the root node with a depth-first search in the anticlockwise direction.
2. Assign an index to each node in the enumerated order. As the result, the data structure becomes one that can be dealt with sequential pattern mining.
3. Each element has index information about a structure, i.e., (parent, first child, next sibling). If there is none, the index is set to '-1'. The first child of a node is the leftmost node among its child nodes. The next sibling of a node is a neighboring node among its sibling nodes, that is, the leftmost node among its sibling nodes.

The transformed data structure is called a semi-structured sequence.

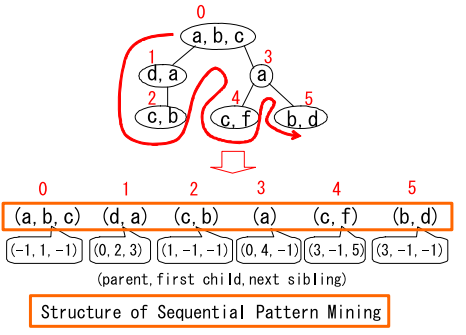


Fig. 3. Example of the Transformation of the Proposed Data Structure

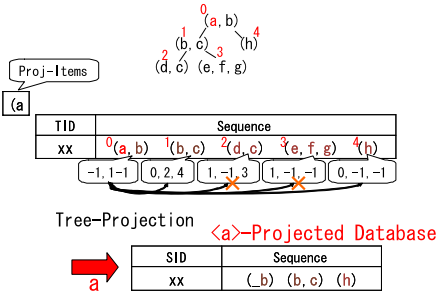


Fig. 4. Example of the Tree Projection with Item a(index=0)

An example of the above transfor mation is shown in Figure 3. The data structure surrounded by the quadrangle in Figure 3 is the data structre of sequential pattern mining. Therefore, the sequential pattern mining algorithm PrefixSpan can be applied to our data structure. However, arbitrary use of PrefixSpan leads to the extraction of disconnected patterns such as $< (a, b, c)(c, b) >$ in Figure 3. Therefore, it is necessary to put a certain constraint on PrefixSpan. Our mining algorithm is an expanded PrefixSpan algorithm with a projection that is constrained in order to extract only connected patterns from the data structure transformed by the method described above. This constrained projection is called a Tree Projection.

4.1 Tree Projection

The tree projection of a semi-structured sequence S with item i is a projection constrained as follows. **Constraint:** The only items included in projected database $S_{<i>}$ are ones that have a path with the items in Proj-Items.

Proj-Items is a stack into which projecting items are pushed. After projection with item i, the item i is pushed into the Proj-Items. When the projection with item i has finished, item i is deleted from the projecting stack. The patterns of items in the projecting stack are frequent patterns. Figure 4 shows the running example of tree projection with item a (index=0).

Tree projection calls the tree projection in the following order:(1).element-projection,(2).child-projection and (3).Level k sibling-projection. After the extraction of frequent items on each projection has finished, the next projecting item is selected and tree projection is called recursively in each projection. We explain each projection as follows.

(1) element-projection with item i is a projection selecting items whose element is equal to the element of the projecting item i. In PrefixSpan, these items are denoted by adding '_' as a prefix.

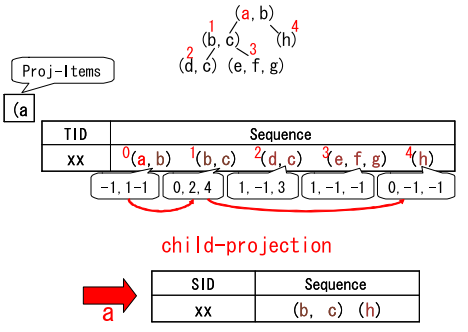


Fig. 5. Example of child-projection

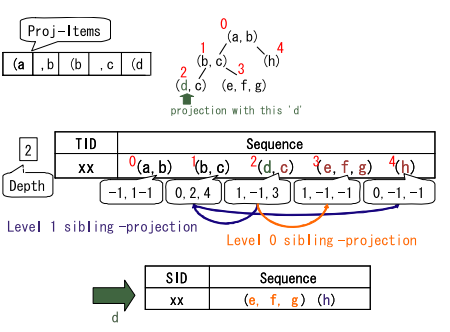


Fig. 6. Example of sibling-projection

(2) child-projection with item i is a projection selecting items whose element is the child element of projecting item i. Specifically, it operates as follows:

1. Find the index of the first child element of the projecting item i
2. Find the index of the next sibling element of the first child element.
3. Find the index of the next sibling element iteratively

As described above, child-projection can extract items that have the parent-child relationship of item i. These items are stored into a projected database. Figure 5 shows a running example of child projection.

(3) Level k sibling projection with item i is a projection selecting items whose element is a sibling element of the k-th ancestor of the projecting item i. **The k-th ancestor** A of the element I indicates that there exists a path between an element I and element A and the difference from depth(A) to depth(I) is k-1. The parent element of the element I is 1st ancestor of the element I. The 0th ancestor of the element I is I itself. For example, in Figure 3, the 1st ancestor of element (c, b) is element (d, a) and the 2nd ancestor is element (a, b, c).

Specifically, it operates as follows:

1. Find the index of the k-th ancestor of the projecting item i.
2. Find the index of the next sibling element of the k-th ancestor.
3. Find the index of the next sibling element iteratively.

Suppose that d is the depth of the projecting item i from the projection start item. Iterate Level k sibling-projection from k=0 to k=d. As described above, Level k sibling-projection can extract the items that are the sibling elements of the k-th ancestor of item i. Put these items into a projected database. Figure 6 shows a running example of sibling-projection.

5 Evaluation of Proposed Method

In this section, we present two evaluations: one evaluating the execution time and the other evaluating the statistics of the number of extracted nodes. In

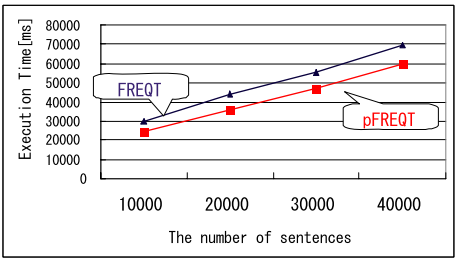


Fig. 7. Comparision of Execution Times

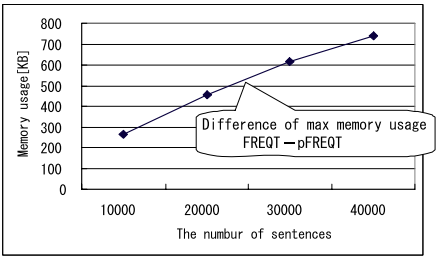


Fig. 8. Difference in Memory Usage between FREQT and pFREQT

particular,we show that our mining method with our new data structure extracts frequent patterns with more nodes than conventional methods.

The dataset was *Aviation Safety Report*[1] gathered by Japan Airlines International Co., Ltd. Dependency parsing by Cabocha[2] is run to one sentence as a unit. In a labeled ordered tree, a segment in Japanese is regarded as a label[3]. In our data structure, a segment is an element and a morpheme in a segment is an item[4].

No conventional mining method can handle our data structure, so we could not evaluate the execution time of our method in comparison with a conventional method. However, our data structure includes a labeled ordered tree as a subset, that is, a labeled ordered tree is our structure where an element has only one item. In this case, tree projection corresponds to the rightmost expansion of FREQT. Therefore, we evaluated the execution time of our method based on a labeled ordered tree. Our method is called pFREQT(projection-based FREQT).

Figure 7 compares the execution time by pFREQT and by FREQT[5] based on a labeled ordered tree. The minimum support was two. pFREQT extracted frequent patterns faster than FREQT. It also projected an item partially corresponding to the relationship of the projectiong nodes. Therefore, pFREQT used less memory than FREQT when searching for patterns. Figure 8 shows the difference in memory usage for FREQT and pFREQT. Apparently, the more sentences there were to handle, the greater the additional memory consumed by FREQT compared with pFREQT. Figure 9 shows the statistics of the number of extracted nodes. We extracted frequent patterns with more than two nodes. We made five datasets, which had a total of two thousand sentences. We evaluated the meanscore, median and maximal values of the number of extracted nodes. The horizontal axis shows each dataset. The vertical axis shows the number of extracted nodes. Figure 9 shows that for each set of statistics, more nodes are extracted by our method than by conventional methods. The more nodes that

[1] It cannot identify individuals because individual information was eliminated.
[2] http://chasen.org/~taku/software/cabocha/
[3] Refer to Figure 1.
[4] Refer to Figure 2.
[5] http://chasen.org/~taku/software/freqt/

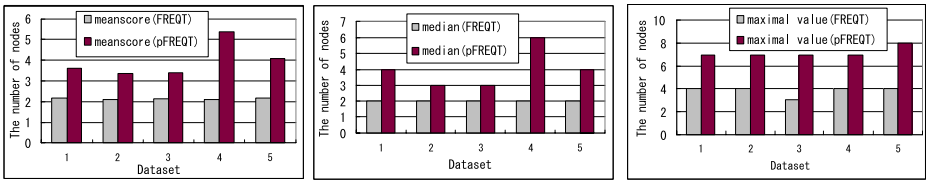


Fig. 9. Statistics of the Number of Extracted Nodes

can be extracted, the more relationships among words with a dependency structure can be extracted.

6 Conclusion

This paper described a semi-structure mining method for extracting frequent patterns of words with a chunk-based dependency structure. It also described a new data structure representing a chunk-based dependency structure and a mining algorithm for it. Our method can extract frequent patterns that the conventional methods cannot extract.

Acknowledgments. We appreciate the cooperation of Takashi Saito and Akira Terada of Japan Airlines International Co., Ltd. This research was funded in part by MEXT Grant-in-Aid for Scientific Research on Priority Areas "i-explosion" in Japan.

References

1. R. Agrawal and R. Srikant: Mining Sequential Patterns, in Proc. of ICDE1995, IEEE Press, pp.3-14(1995)
2. J. Pei, J. Han, B. Mortazavi-Asl, H. Pnto,Q. Chen, U. Dayal, and M. Hsu : PrefixSpan:Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, in Proc. of ICDE2001, IEEE Press, pp.215-224(2001)
3. J. Pei, J. Han, and W. Wang: Mining sequential patterns with constraints in large databases, in Proc. of CIKM2002, pp.18-25(2002)
4. Yu Hirate and Hayato Yamana : Sequential Pattern Mining with Time Interval, in Proc. of PAKDD2006(2006)
5. Kenji Abe, Shinji Kawasoe, Tatsuya Asai, Hiroki Arimura, Setsuo Arikawa: Optimized Substructure Discovery for Semi-structured Data, in Proc. of PKDD 2002(2002)
6. Tatsuya Asai, Kenji Abe, Shinji Kawasoe, Hiroki Arimura, Hiroshi Sakamoto, Setsuo Arikawa: Efficient Substructure Discovery from Large Semi-structured Data , in Proc. of SDM 2002(2002)
7. M. J. Zaki: Efficiently mining frequent trees in a forest, in Proc. of SIGKDD2002(2002)

Principal Curves with Feature Continuity

Ming-ming Sun and Jing-yu Yang

Department of Computer Science, Nanjing University of Science and Technology, Nanjing, 210094, P.R. China
sunmm@njust.edu.cn, yangjy@mail.njust.edu.cn

Abstract. Principal curves were proposed as the nonlinear generalization of PCA. However, for the tasks of feature extraction for signal representation at which PCA is adept, existing definitions of principal curves have some weakness in their theoretical bases thus fail to get reasonable results in many situations. In this paper, a new definition of principal curves - Principal Curve with Feature Continuity (PCFC) is proposed. PCFC focuses on both reconstruction error minimization and feature continuity. It builds a continuous mapping from samples to the extracted features so the features preserve the inner structures of the sample set, which benefits the researchers to learn the properties of the sample set. The existence and the differential properties of PCFC are studied and the results are presented in this paper.

1 Introduction

Feature extraction for signal representation is an invariable topic in pattern recognition, which aims to concisely describe the dominant feature of a sample set. Numerous methods have been developed to extract reliable features for various data sets. PCA (principal component analysis) may be the most well-known and widely used method. The features extracted by PCA have many good properties and were successfully used in many problems. However, the world is nonlinear. When the linear model of PCA fails to describe the complex inner structures of nonlinear patterns, many nonlinear generalizations of PCA are developed. Among those methods, principal curve methods are most attractive. Principal curve methods, such as HS principal curves (HSPC) [1], T principal curve (TPC)[2], K principal curves (KPC)[3] ,principal curves with bounded turn (PCBT)[4], D principal curves(DPC)[5] and so on, are proposed by generalizing respective properties of the first principal component line. These principal curves have been successfully applied for the tasks of data description, that is, looking for descriptive models that can describe the data best [6] [7].

However, for the tasks of extracting reliable features for signal presentation, existing principal curves are weak on their theoretic bases. In fact, for such tasks, the most important requirement is that the extracted features preserve the inner structures of the data distribution so that the properties of the data sets can be reliably reflected by the extracted features. Generally, it is considered that the inner structures of data sets are embodied in neighbor relationships between samples. So, a basic requirement of feature extractor is that the features extracted from neighboring samples should still be neighboring. It is easy to know that PCA does keep the neighbor relationship for neighboring

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 785–792, 2007.

samples. However, when we take the project index functions of principal curves as the feature extractors, existing principal curves do not guarantee the neighbor relationship preserving for neighboring samples. Furthermore, reconstruction error minimization is another requirement of feature extraction for signal representation, but some kinds of principal curves do not satisfy the requirement.

Recently, we propose a new definition of principal curves: Principal Curve with Feature Continuity(PCFC). PCFC is proposed for extracting one-dimensional features for signal representation. It is focused on both the reconstruction error minimization and the neighbor relationship preserving for neighboring samples. From the viewpoint of feature extraction for signal representation, PCFC is an optimal one-dimensional feature extractor in the sample space.

The rest of the paper is organized as follows: In Section 2, we introduce the framework of feature extraction for signal representation using curves and evaluate the existing kinds of principal curves inside this framework. Then the definition of PCFC is introduced and its properties are analyzed in Section 3. Finally, Section 4 concludes with a description of directions for future research.

2 One-Dimensional Feature Extraction for Signal Representation Using Curves

In this section, we introduce some basic concepts of curves and the framework of feature extraction using curves.

2.1 Preliminaries and Notation

Definition 1. *A parameterized curve* $\mathbf{f}$ *is a continuous mapping* $\mathbf{f} : \Lambda \to \mathbb{R}^d$*, where* Λ *is a close subset of* $\mathbb{R}$*.*

We denote by $I_{\mathbf{f}}$ the domain of $\mathbf{f}$ and by $G_{\mathbf{f}}$ its range.

Definition 2. *The length of a parameterized curve* $\mathbf{f} : \Lambda \to \mathbb{R}^d$ *over an interval* $[\alpha, \beta] \subset I_{\mathbf{f}}$*, denoted by* $l(\mathbf{f}, \alpha, \beta)$*, is defined by*

$$l(\mathbf{f}, \alpha, \beta) = \sup \sum_{i=1}^{N} \|\mathbf{f}(t_i) - \mathbf{f}(t_{i-1})\|, \tag{1}$$

where the supremum is taken over all finite partitions of $[\alpha, \beta]$ *with arbitrary subdivision points* $\alpha = t_0 \leq t_1 < \cdots \leq t_N = \beta$ *for* $N \geq 1$*. The total length of the parameterized curve* $\mathbf{f}$ *is defined as:*

$$l(\mathbf{f}) = \sup_{\alpha,\beta \in I_{\mathbf{f}}} l(\mathbf{f}, \alpha, \beta) \tag{2}$$

A parameterized curve $\mathbf{f}$ *is called rectifiable if* $l(\mathbf{f}) < \infty$*.*

Definition 3. *Consider a piecewise-linear curve* $\mathbf{f}$ *with vertices* $v_0 \ldots v_n$. *Let* $a_i = v_i - v_{i-1}$ *and let* ϕ_i *be the angle between* a_i *and* a_{i+1}. *The total turn of this piecewise-linear curve is defined by*

$$\kappa(\mathbf{f}) = \sum_{i=1}^{n-1} \phi_i. \tag{3}$$

For a general curve $\mathbf{f}$, *the turn accumulated over an interval* $[\alpha, \beta]$ *of its domain is defined as the supremum over all piecewise-linear inscriptions in* $[\alpha, \beta]$, *i.e.,*

$$\kappa(\mathbf{f}, \alpha, \beta) = \sup_n \sup_g \kappa(\mathbf{g}) \tag{4}$$

where $\mathbf{g}$ *is a piecewise-linear curve with vertices* $\mathbf{f}(t_0) \ldots \mathbf{f}(t_n)$ *such that* $t_i \in I_\mathbf{f}$ *and* $\alpha = t_0 < t_1 < \cdots < t_{n-1} < t_n = \beta$. *The total turn of the*

$$\kappa(\mathbf{f}) = \sup_{\alpha,\beta \in I_\mathbf{f}} \kappa(\mathbf{f}, \alpha, \beta) \tag{5}$$

A parameterized curve $\mathbf{f}$ over $I_\mathbf{f}$ is called to be parameterized by its arc length if for any interval $[\alpha, \beta] \subset I_\mathbf{f}$ and $t \in [\alpha, \beta]$, $l(\mathbf{f}, \alpha, t) = t - \alpha$.

The projection index of a point x to the parameterized curve $\mathbf{f}$ is defined as:

$$t_\mathbf{f}(x) = \sup_{t \in I_\mathbf{f}} \{t : \|x - \mathbf{f}(t)\| = \inf_{\tau \in I_\mathbf{f}} \|x - \mathbf{f}(\tau)\|\}. \tag{6}$$

We denote the distortion of x due to its projection onto a parameterized curve $\mathbf{f}$ as:

$$\Delta(x, \mathbf{f}) = \|x - \mathbf{f}(t_\mathbf{f}(x))\|^2. \tag{7}$$

If the set $\{t : \|x - \mathbf{f}(t)\| = \Delta(x, \mathbf{f})\}$ contains more than one element, then x is called an ambiguity point to $\mathbf{f}$.

For a random variable X, we denote the expected distortion due to its projection onto $\mathbf{f}$ as:

$$\Delta(\mathbf{f}) = E[\Delta(x, \mathbf{f})] = E[\|x - \mathbf{f}(t_\mathbf{f}(x))\|^2]. \tag{8}$$

Two different parameterized curves $\mathbf{f} : I_\mathbf{f} \to R^d$ and $\mathbf{g} : I_\mathbf{g} \to R^d$ may define a same path passing the same set of points with the same order . We can regard these two parameterized curves define one and the same *curve* L. In fact, an equivalence relation can be established between the parameterized curves, and an equivalence class defines a *curve*(for more details, please consult [8]). In this situation, we call $\mathbf{f}$ and $\mathbf{g}$ the parameterizations of L. It is easy to prove that all parameterizations of a curve have the same length, the same total turn and the same distortion from a point and a random variable. From the theory of irregular curves [8], we know that curves with finite turn can be parameterized by its arc length. To eliminate the ambiguity of parameterizations, *in the remaining parts of the paper, we only consider the curves with finite turn; and unless explicitly mentioned, we always prescribe that the parameterized curves are parameterized by their arc length and satisfy* $t_\mathbf{f}(O) = 0$.

2.2 Feature Extractor and Reconstruction Function

Given a random variable X and a parameterized curve $\mathbf{f}$, we define

$$F_{\mathbf{f}}(x) = t_{\mathbf{f}}(x). \tag{9}$$

as the one-dimensional feature extractor. In fact, according to the prescription in above section, $F_{\mathbf{f}}(x)$ is the directed arc length between the projection points of O and x on the curve. So $F_{\mathbf{f}}$ is independent of the parameterization.

Now the one-dimensional feature extraction framework using curves can be established by selecting $F_{\mathbf{f}}$ as the feature extractor and $\mathbf{f}$ as the reconstruction function. That is, given a sample x of X, $F_{\mathbf{f}}(x)$ is the feature of x; and given a feature value t, $\mathbf{f}(t)$ is the reconstruction point of the feature t. Note that the framework is coincident with PCA, in which $\mathbf{f}$ is the first principal component line of the random variable.

2.3 Evaluation Criterions of Feature Extractors

When evaluating a feature extractor for signal representation, there are commonly two criterions:

- Feature Continuity: the feature extractor should be a continuous function of samples. This criterion evaluates the ability of feature extractor to preserve the inner structure of data set. If this criterion is not satisfied, the features of similar samples would be dissimilar, which is disadvantageous for researchers to learn the properties of the sample set via the extracted features.
- Reconstruction Error Minimization: the distortion between the samples and reconstruction points should be minimized. This criterion is the natural requirement for signal representation.

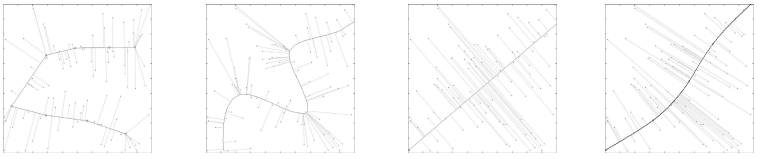
In the one-dimensional feature extraction framework using curves, these two criterions becomes: 1) $F_{\mathbf{f}}$ should be continuous; 2) $\Delta(\mathbf{f})$ should be minimized.

According to these two criterions, existing principal curve methods are evaluated as follows from the view point of feature extraction for signal representation.

Feature Continuity. We can easily show instances that existing kinds of principal curves do not preserve the continuity of $F_{\mathbf{f}}$. For a random variable uniformly distributed on a circle $\{(r, \phi) : r = R\}$, the HSPC, DPC, TPC, KPC with upper-bound of length larger than $2\pi R$ and PCBT with upper-bound of total turn larger than 2π are all the circle. Given any parameterization $\mathbf{f}$ of the circle, the $F_{\mathbf{f}}$ cannot be continuous. From the instance, we can see that the theories of existing principal curves do not guarantee the continuity of the feature extractor $F_{\mathbf{f}}$.

Because of lacking the theoretic assurance of the continuity of $F_{\mathbf{f}}$, in practice, the learning algorithms of existing principal curves failed to learn a reasonable feature extractor. Figure 1 exhibits the learning results of K principal curve and HS principal curve on a data set of 100 samples uniformly distributed on a unit square, with comparison to those of first principal component line and a desired curve.

We can see from the results that the learning results of K principal curve and HS principal curve achieves smaller reconstruction errors. However, neighboring samples



(a) (b) (c) (d)

Fig. 1. Learning results of (a) K principal curve, (b) HS principal curve (c) Principal component line. The desired curve is shown in (d). In the K principal curve learning algorithm [3], the penalty coefficient is set to 0.13 recommended in [3]. Gaussian kernel with parameter 0.3 is employed in HS principal curve learning algorithm.

are projected to the points far from each other on the curves, which means that the feature continuity has been violated. The first principal component line shown in Figure 1 (c) does preserve the continuity of $F_{\mathbf{f}}$, but the reconstruction error of it is high. The curve illustrated in Figure 1 (d) preserves the feature continuity; and compared with the first principal component line, it achieves a lower value of $\Delta(\mathbf{f})$, which means more effective in feature extraction. It is a desired curve for feature extraction.

Reconstruction Error Minimization. KPC and PCBT minimize $\Delta(\mathbf{f})$ in respective curve classes, while other principal curves such as HSPC, TPC and DPC do not necessarily do so.

To sum up, existing definitions of principal curves are not competent for the task of feature extraction for signal representation. The main reason is that those definitions do not preserve the continuity of $F_{\mathbf{f}}$; furthermore, some of them do not minimize $\Delta(\mathbf{f})$ either.

3 Principal Curves with Feature Continuity

3.1 Concept and Definition

According to the two criterions proposed in above section, the desired curve is the one that minimizes $\Delta(\mathbf{f})$ with continuous $F_{\mathbf{f}}$. We call it Principal Curve with Feature Continuity(PCFC). To support the validity of the concept of PCFC, we impose more restrict regularity conditions on the class of curves to be studied.

Let us denote the open ball centered at the origin with radius r as B_r. Given a parameterized curve $\mathbf{f}$, if $G_{\mathbf{f}} \cap B_r \neq \phi$, then we know that $\mathbf{f}(0) \in B_r$. Let $t_1 = \inf\{t \leq 0 | \forall s \in (t, 0], \mathbf{f}(s) \in B_r\}$ and $t_2 = \sup\{t \geq 0 | \forall s \in [0, t), \mathbf{f}(s) \in B_r\}$. We define $\mathbf{f}|_{B_r}$ as the curve $\mathbf{g}$ with $I_{\mathbf{g}} = [t_1, t_2]$ and $\mathbf{g}(t) = \mathbf{f}(t), t \in [t_1, t_2]$. We call $\mathbf{f}|_{B_r}$ the restriction of $\mathbf{f}$ to the ball B_r.

To avoid the situation that infimum of $\Delta(\mathbf{f})$ may not be achieved by any curve, (an example is detailed in [4]), we consider the following class of curves [4]:

$$C_{T,\tau} = \{\mathbf{f} |\ \kappa(\mathbf{f}) \leq T, \kappa(\mathbf{f}) - \kappa(\mathbf{f}|B_R) \leq \tau(R)\}, \tag{10}$$

where $\tau(R)$ is continuous and decreasing to zero in R.

In order to get reliable cognition about the properties of the data set through the extracted features, the scale of differences between the extracted features of neighboring samples must be under control. So we consider the curves inside the following class:

$$\Omega_{\delta,K,S(X)} = \{\mathbf{f} \mid \|F_{\mathbf{f}}(x_1) - F_{\mathbf{f}}(x_2)\| \leq K\|x_1 - x_2\|, \\ \forall x_1, x_2 \in S(X),\ \|x_1 - x_2\| < \delta\}. \tag{11}$$

where $S(X)$ denote the support of the random variable X.

Finally, for a random variable X, we consider the class of curves as follows:

$$\Gamma_{T,\tau,\delta,K,S(X)} = \Omega_{\delta,K,S(X)} \cap C_{T,\tau}. \tag{12}$$

Now, we give the definition of principal curve with feature continuity:

Definition 4. *Given a random variable X, a parameterized curve $\mathbf{f}^*$ is called the principal curve with feature continuity for X with parameter (T, τ, δ, K) if it minimizes $\Delta(\mathbf{f})$ among all the curves in $\Gamma_{T,\tau,\delta,K,S(X)}$.*

Given the definition, following theorem ensures that the PCFC always exists for data distributions with finite second moments and open support:

Theorem 1. *If $E(\|X\|^2) < \infty$ and $S(X)$ is open, then $\forall T > 0, \delta > 0, K \geq 0$ and τ continuously decrease to zero, then there exists a principal curve with feature continuity for X with parameter (T, τ, δ, K).*

Because of the limitation of the space of the paper, the detailed proof is not presented here. The outline of the proof is shown below. The key to prove the theorem is the following lemma:

Lemma 1. *For any open set Y and compact set $A \subset \mathbb{R}^d$, the set of curves $\{\mathbf{f} | G_{\mathbf{f}} \subset A, \kappa(\mathbf{f}) \leq T\} \cap \Omega_{\delta,K,Y}$ is compact.*

Given this lemma, the other part of the proof roughly follows the way of the proof of the existence of PCBT [4]. Let $\Delta^* = \inf_{\mathbf{f} \in \Gamma_{T,\tau,\delta,D,S(X)}} \Delta(\mathbf{f})$. We know that there exists a sequence of curves $\mathbf{f}_n \in \Gamma_{T,\tau,\delta,D,S(X)}$ such that $\Delta(\mathbf{f}_n) \to \Delta^*$. In a similar way with the proof of the existence of PCBT, we can prove that there exist in $\{\mathbf{f}_n\}$ a subsequences $\{\mathbf{f}_{n_k}\}$ whose restrictions to any ball with radius large enough converge. Then a 'limiting curve' $\mathbf{f}^*$ can be defined as the curve that satisfies when $\forall r$ large enough,

$$\mathbf{f}_{n_k}|_{B_r} \to \mathbf{f}^*|_{B_r}. \tag{13}$$

Then, following the way in [4], we can proof that $\mathbf{f}^* \in \Gamma_{T,\tau,\delta,D,S(X)}$ an $\Delta(\mathbf{f}^*) = \Delta^*$. Thus, $\mathbf{f}^*$ is the desired principal curve with feature continuity.

When $S(X)$ is not open, we consider to expand the support of the distribution: unite $S(X)$ with a set of small open balls centered at its boundary points. We denote the resulting set as $U(X)$. Then we find in $\Gamma_{T,\tau,\delta,K,U(X)}$ a curve $\mathbf{f}^*$ which minimizes $\Delta(\mathbf{f})$. Similarly, we can prove that $\mathbf{f}^*$ exists. This disposal is reasonable under the consideration of generalization and influences of noises in real world problems.

3.2 Properties of Principal Curves with Feature Continuity

The definition of principal curves with feature continuity directly results in following proposition:

Proposition 1. *If* $\mathbf{f}$ *is a PCFC for random variable* X *with an open support* $S(X)$*, then there is no ambiguity point of* $\mathbf{f}$ *in* $S(X)$*.*

Now we study the differential properties of principal curves with feature continuity. Since the PCFC has finite turn, according to the theory of irregular curves [8], following proposition is valid.

Proposition 2. *If* $\mathbf{f}$ *is the principal curves with feature continuity of* X *(parameterized by its arc length), then:*

- $\forall t \in I_{\mathbf{f}}$*, the left derivative vector* $\mathbf{f}'_l(t)$ *and right derivative vector* $\mathbf{f}'_r(t)$ *exist and their length equal to 1;*
- *the non-differentiable points of* $\mathbf{f}(t)$ *are not more than countable.*

Proposition 3. *If* $\mathbf{f}$ *is the principal curve with feature continuity of* X *(parameterized by its arc length). If for* t_0*, there* $\exists x_0, \delta_0 > 0$ *satisfying following conditions:*

- $t_0 = t_{\mathbf{f}}(x_0)$*;*
- $\mathbb{O}(x_0, \delta_0) \subset S(X)$*;*
- $\forall 0 < \delta < \delta_0, t_f(\mathbb{O}(x_0, \delta))$ *is open.*

then $\mathbf{f}$ *is first order differentiable at* t_0*. Especially, when* $\mathbf{f}(t_0)$ *is an inner point of* $S(X)$*,* $\mathbf{f}(t)$ *is first order differentiable at* t_0*.*

The facts revealed by proposition 3 are illustrated in Figure 2.

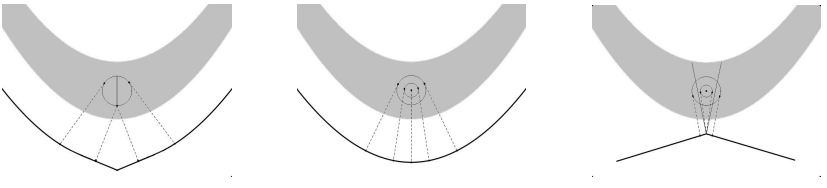


Fig. 2. The differential properties of PCFC. (a) A case that should not happen for PCFC, (b) A case satisfying the condition of proposition 3, (c) A case of non smooth PCFC. In the figure, gray region denotes a part of the support of a distribution and the circles inside them denote the neighborhoods.

Figure 2 shows a part of a distribution and various possibilities of PCFC. In (a), the projection of the neighborhood is split into separated parts, thus violates the feature continuity of PCFC. Thus, the non-smoothness described in (a) is not allowed for PCFC. (b) shows the situation that the conditions of proposition 3 describe: however small the neighborhood is, the set of projection indexes of points in the neighborhood is open. According to proposition 3, the PCFC must be differentiable at t_0. A PCFC that is somewhere not differentiable is shown in (c). In this case, when the neighborhood is small enough, the points in the neighborhood all project into one and the same point, thus the condition of proposition 3 is not satisfied.

4 Conclusion and Future Work

In this paper, we proposed a new definition of principal curves - Principal Curve with Feature Continuity to extract reliable one-dimensional features for signal representation. PCFC focuses on feature continuity and reconstruction error minimization. From the viewpoint of signal representation, it is an optimal one-dimensional feature extractor in the sample space. The existence of the PCFC is guaranteed for a large set of data distributions. PCFC also has good differential properties.

Developing an effective and efficient algorithm to learn the PCFC is an important task in our future work. Furthermore, multi-dimensional features may benefits researchers to understand the detailed properties of the data set. To extract multi-dimensional features for signal representation, we would like to study the definition and properties of principal manifold with feature continuity, which will be a part of our future work.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant No. 60473039, No. 60503026, No. 60472060 and No. 60632050.

References

1. Hastie, T., Stuetzle, W.: Principal curves. Journal of the American Statistical Association **84**(406) (1989) 502–516
2. Tibshirani, R.: Principal curves revisited. Statistics and Computation (1992) 183–190
3. Kégl, B., Krzyzak, A., Linder, T., Zeger, K.: Learning and design of principal curves. IEEE Transaction on Pattern Analysis and Machine Intelligence **22**(3) (2000) 281–297
4. Sandilya, S., Kulkarni, S.R.: Principal curves with bounded turn. IEEE Trans. on Information Theory **48**(10) (2002) 2789–2793
5. Delicado, P.: Another look at principal curves and surfaces. Journal of Multivariate Analysis **77**(2) (2001) 84–116
6. Kégl, B., Krzyzak, A.: Piecewise linear skeletonization using principal curves. IEEE Transaction on Pattern Analysis and Machine Intelligence **24**(1) (2002) 59–74
7. Byers, S., Raftery, A.E.: Nearest neighbor clutter removal for estimating features in spatial point processes. Technical report, Department of Statistics University of Washington (1996)
8. Alexandrov, A., YU.G.Reshetnyak: General Theory of Irregular Curves. Kluwer Academic Publishers, Netherland (1989)

Kernel-Based Linear Neighborhood Propagation for Semantic Video Annotation

Jinhui Tang[1,⋆], Xian-Sheng Hua[2], Yan Song[1], Guo-Jun Qi[1], and Xiuqing Wu[1]

[1] Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, 230027 China
jhtang@mail.ustc.edu.cn
[2] Microsoft Research Asia, Beijing, 100080 China
xshua@microsoft.com

Abstract. The insufficiency of labeled training samples for representing the distribution of the entire data set (include labeled and unlabeled) is a major obstacle in automatic semantic annotation of large-scale video database. Semi-supervised learning algorithms, which attempt to learn from both labeled and unlabeled data, are promising to solve this problem. In this paper, we present a novel semi-supervised approach named *Kernel based Local Neighborhood Propagation* (Kernel LNP) for video annotation. This approach combines the *consistency assumption* and the *Local Linear Embedding* (LLE) method in a nonlinear kernel-mapped space, which improves a recently proposed method *Local Neighborhood Propagation* (LNP) by tackling the limitation of its local linear assumption on the distribution of semantics. Experiments conducted on the TRECVID data set demonstrate that this approach can obtain a more accurate result than LNP for video semantic annotation.

Keywords: Video Annotation, Kernel Method, Label Propagation.

1 Introduction

Automatic annotation (or we may call it high-level feature extraction) of video and video segments is essential for enabling semantic-level video search. As manually annotating large video archive is labor-intensive and time-consuming, many automatic approaches are proposed for this issue. Generally, these methods build statistical models from manually pre-labeled samples, and then assign the labels for the unlabeled ones using these models. This process has a major obstacle: the labeled data is limited so that the distribution of the labeled data typically does not well represent the distribution of the entire dataset (include labeled and unlabeled), which usually leads to inaccurate annotation results.

Semi-supervised learning, which attempts to learn from both labeled and unlabeled data, is a promising approach to deal with the above obstacle. Many

⋆ This work was performed when the first and fourth authors were research interns at Microsoft Research Asia.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 793–800, 2007.

works on this topic are reported in literature of machine learning community [1]. And some of them have been applied to video or image annotation [2,3,4].

The key point of semi-supervised learning is the *consistency assumption* [5]: nearby samples in the feature space or samples on the same structure (also referred to as a cluster or a manifold) are likely to have the same label. This assumption considers both the local smoothness and the structure smoothness of the semantic distribution.

There are close relations between the *consistency assumption* and nonlinear dimensionality reduction schemes [6,7,10] since intrinsically they follow the same idea of reducing the global coordinate system of the dataset to a lower-dimensional one while preserving the local distribution structure. Recently, a method called *Local Neighborhood Propagation* (LNP) [8,9] is proposed to combine these two strategies. LNP borrows the basic assumption of *Local Linear Embedding* (LLE) [6,10] that each sample can be reconstructed by its neighboring samples linearly, and further assumes that the label of the sample can be reconstructed by the labels of its neighboring samples using the same coefficients. This method potentially assumes that the mapping from the feature to label is linear in a local area (to be detailed in Section 2). However, if the semantic super-plane has a high curvature in this area, LNP will fail. In other words, if the labels of the samples in the local area distribute complexly in the feature space, this linear assumption is not appropriate. Therefore this method is not suitable to tackle the video semantic annotation problem, since typically the semantic distribution of video segments, which are collected from different sources and span a large time interval, is very complex in the feature space.

In this paper, we propose a novel method for automatic video annotation named *Kernel based Local Neighborhood Propagation* (Kernel LNP), which also combines the *consistency assumption* and LLE but is applied in a nonlinear kernel-mapped space. This method is able to handle more complex situation since it holds both the advantages of LNP and kernel methods, through mapping the input feature space to a nonlinear kernel-mapped feature space. The experiments conducted on the TRECVID [13] data set demonstrate that Kernel LNP is more appropriate than LNP for complex applications and can obtain a more accurate result for video high-level feature extraction.

The rest of this paper is organized as follows. In Section 2, we briefly introduce LNP and analyze its limitation; and the proposed Kernel LNP for the video semantic annotation problem is detailed in Section 3. Experiments are introduced in Section 4, followed by the conclusion remarks and future work in Section 5.

2 LNP and Its Limitation

In this section, we briefly introduce LNP and analyze its limitation. LNP is based on the assumption that the label of each sample can be reconstructed linearly by its neighbors' labels, and the coefficients of reconstructing the label is the same as the ones for reconstructing the feature vector of the sample. This can be formulated as

$$x_i = \sum_{x_j \in N(x_i)} \alpha_j x_j \Rightarrow f_i = \sum_{x_j \in N(x_i)} \alpha_j f_j, \tag{1}$$

where $N(x_i)$ is the neighbors of sample x_i, and f_i is the label of x_i.

Define the mapping from the feature to label as $f : x_i \mapsto f_i$, we can obtain:

$$f_i = f(x_i) = f(\sum_{x_j \in N(x_i)} \alpha_j x_j) \tag{2}$$

and

$$f_i = \sum_{x_j \in N(x_i)} \alpha_j f_j = \sum_{x_j \in N(x_i)} \alpha_j f(x_j) \tag{3}$$

Combine (2) and (3), we have

$$f(\sum_{x_j \in N(x_i)} \alpha_j x_j) = \sum_{x_j \in N(x_i)} \alpha_j f(x_j) \tag{4}$$

Equation (4) indicates that the mapping from the feature to label is linear in a local area. Furthermore, from the viewpoint of LLE, LNP assumes that the semantic is a 1-D manifold embedded in the feature space. Local linear assumption or 1-D manifold assumption is not able to handle the data with complex semantic distribution. If the semantic super-plane has a high curvature in the local area, this assumption is not appropriate. So this method cannot tackle the video semantic annotation problem since the semantics of video segments often have very complex distributions. We call this drawback *limitation of local linear assumption on the distribution of semantics*.

3 Kernel-Based Local Neighborhood Propagation

To tackle the aforementioned limitation of LNP, in Kernel LNP, we map the features to a kernel-mapped space and then try to obtain the reconstruction coefficients in this nonlinear space. Kernel LNP also assumes that the label of each sample can be reconstructed linearly by its neighbors' labels, but the reconstruction coefficients are obtained in the nonlinear mapped space.

Let $X = \{x_1, x_2, ..., x_l, x_{l+1}, ..., x_n\}$ be a set of n samples (i.e., video shots for our application) in R^d (feature space with d-dimensional features). $X = L \bigcup U$, where the sample set $L = \{x_1, x_2, ..., x_l\}$ contains the first l samples labeled as $y_i \in \{1, 0\}$ $(1 \leqslant i \leqslant l)$ and $U = \{x_{l+1}, x_{l+2}, ..., x_n\}$ are unlabeled ones. According to the task of high-level feature extraction in TRECVID [13], the objective here is to rank the remaining unlabeled samples, so we will assign real values (between 0 and 1) to the samples as labels instead of "0" or "1".

Considering a kernel mapping $\phi(\cdot)$ operating from input space X to a mapped space Φ:

$$\begin{aligned} \phi : X &\rightarrow \Phi, \\ x &\mapsto \phi(x). \end{aligned}$$

The data set can be mapped to $\{\phi(x_1), \phi(x_2), ..., \phi(x_l), \phi(x_{l+1}), ..., \phi(x_n)\}$. Then, the kernel matrix K of dot products can be represented as

$$K = (k_{ij})_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n}, \tag{5}$$

where $k_{ij} = \phi^t(x_i) \cdot \phi(x_j)$, and for the kernel function, the *Radial Basis Function* (RBF) is adopted in our experiments.

We find the k nearest neighbors $N(\phi_i)$ of every $\phi(x_i) \in \Phi$ using the following distance (for concision, we use ϕ_i to substitute $\phi(x_i)$):

$$\begin{aligned} dist(\phi_i, \phi_j) &= \|\phi(x_i) - \phi(x_j)\| \\ &= \sqrt{\phi_i^T \phi_i - 2\phi_i^T \phi_j + \phi_j^T \phi_j} \\ &= \sqrt{k_{ii} - 2k_{ij} + k_{jj}}. \end{aligned} \tag{6}$$

Please note this formula shows that the distance can be obtained directly from the kernel matrix instead of the mapping function $\phi(\cdot)$.

According to the assumption of LLE, x_i can be linearly reconstructed from its neighbors. Using the kernel mapping, the coefficients obtained in the mapped space can reconstruct the labels better than the coefficients obtained in the original feature space. To compute the optimal reconstruction coefficients, the reconstruction error of ϕ_i is defined as:

$$\varepsilon_i = \|\phi_i - \sum_{\phi_j \in N(\phi_i)} w_{ij} \phi_j\|, \tag{7}$$

where w_{ij} is the reconstruction coefficient for ϕ_i from $\phi_j \in N(\phi_i) = \{\phi_{i1}, ..., \phi_{ik}\}$, $\mathbf{w}_i = (w_{i1}, w_{i2}, ..., w_{ik})^T$ is the vector of reconstruction coefficients. We could obtain the optimal coefficients for ϕ_i by solving the optimization problem:

$$\begin{aligned} \mathbf{w}_i^* &= argmin_{\mathbf{w}_i} \varepsilon_i \\ &= argmin_{\mathbf{w}_i} \|\phi_i - \sum_{\phi_j \in N(\phi_i)} w_{ij} \phi_j\| \\ s.t.\ w_{ij} &= 0 \quad for \quad \forall \phi_j \notin N(\phi_i), \quad and \quad \sum_j w_{ij} = 1 \end{aligned} \tag{8}$$

Introduce a "local" Gram matrix of ϕ_i in the kernel-mapped space here:

$$\begin{aligned} G_i &= (\phi_i \mathbf{1}^T - \Phi_i)^T (\phi_i \mathbf{1}^T - \Phi_i) \\ &= ((\phi_i - \phi_p)(\phi_i - \phi_q))_{\phi_p \in N(\phi_i), \phi_q \in N(\phi_i)} \\ &= (k_{ii} - k_{ip} - k_{iq} + k_{pq})_{\phi_p \in N(\phi_i), \phi_q \in N(\phi_i)} \\ &= (g_{ipq})_{\phi_p \in N(\phi_i), \phi_q \in N(\phi_i)} \end{aligned} \tag{9}$$

where Φ_i is a matrix formed by the mapped feature vectors for the k nearest neighbors of i-th sample, "$\mathbf{1}$" is a k-dimensional column vector with each entry equals to 1,and $g_{ipq} = k_{ii} - k_{ip} - k_{iq} + k_{pq}$ is the entry in matrix G_i $(k \times k)$.

Please note here the subscripts p and q do not mean g_{ipq} is the element in p-th row and q-th column ($1 \leqslant p, q \leqslant n$) in the matrix, but it is obtained according to the positions of ϕ_p and ϕ_q in Φ_i. For example, if ϕ_p is the r-th column and ϕ_q is the s-th column in $\Phi_i (1 \leqslant r, s \leqslant k)$, g_{ipq} is of the element in r-th row and s-th column of G_i. In some unusual cases, this Gram matrix may be singular or nearly singular. So it must be added a small multiple of the identity matrix for regularization [10].

Similar to the distance measure (see equation (6)), we can see that the Gram matrix in (9) also can be calculated directly from the kernel matrix instead of the mapping function. Therefore, in the entire computing procedure, the mapping function actually is not explicitly required.

This obtained Gram matrix is symmetric and semi-positive definite. We can use the Lagrange multiplier to enforce the constraint $\sum_j w_{ij} = 1$ for the optimization problem in (8). According to the inverse Gram matrix, the optimal reconstruction coefficients vector $\mathbf{w}_i^*$ for i-th sample can be obtained as:

$$\mathbf{w}_i^* = \frac{G_i^{-1}\mathbf{1}}{\mathbf{1}G_i^{-1}\mathbf{1}} \tag{10}$$

It is intuitive that the obtained reconstruction coefficients reflect the intrinsic local semantic structure of the samples in the mapped space. These coefficients will be applied to reconstruct the unlabeled samples' labels (which are real values instead of "0" or "1"), that is, to estimate the prediction function f. In order to obtain the optimal f, we define the following cost function

$$C(f) = \sum_{i=1}^{n} \|f_i - \sum_{\phi_j \in N(\phi_i)} w_{ij} f_j\|^2 + \infty \sum_{i=1}^{l} (f_i - y_i)^2, \tag{11}$$

where f_i is the label of sample x_i.

Minimizing this cost will optimally reconstruct the labels of all unlabeled samples from the counterparts of their neighbors. And from the view of label propagation, minimizing this cost results in iterative label information propagations from labeled samples to other samples according to the linear neighborhood structure in the nonlinear mapped space. Formally, this optimization objective is represented as

$$\begin{aligned} f^* \;&=\; argmin_f \sum_{i=1}^{n} \|f_i - \sum_{\phi_j \in N(\phi_i)} w_{ij} f_j\|^2 \\ s.t. \quad & f_i = y_i \quad (1 \leqslant i \leqslant l) \end{aligned} \tag{12}$$

It has the same form as the optimization problem in [8], where three methods were proposed to solve it: Lagrangian method, eigen-decomposition method [11] and the method similar to anisotropic diffusion [12]. Since the video data set typically is very large (e.g., TRECVID 2005 dataset has about 126,000 subshots), it is difficult to storage the similarity matrix and compute its inversion

or eigenvalues. To avoid handling large matrix, we adopt the third method. Therefore, we just need to record a small amount of neighbors of each sample, as well as the distances between the sample and its neighbors. This is actually an information propagation process from the label of each sample to its neighbors. The main procedure of the above algorithm is summarized as followed:

a. Using the RBF as the kernel, compute the kernel matrix $K = (k_{ij})_{1 \leqslant i,j \leqslant n}$ with respect to X (K is a sparse matrix as there are many entries equal to 0);
b. Find the k nearest neighbors of each sample ϕ_i in Φ using the distance measure in (6);
c. Compute the Gram matrix G_i according to (9), then $\mathbf{w}_i^*$ can be computed according to (10);
d. Predict the unlabeled samples' real-value labels by solving the optimization problem in (12).
e. Rank the unlabeled samples according the labels obtained in step d.

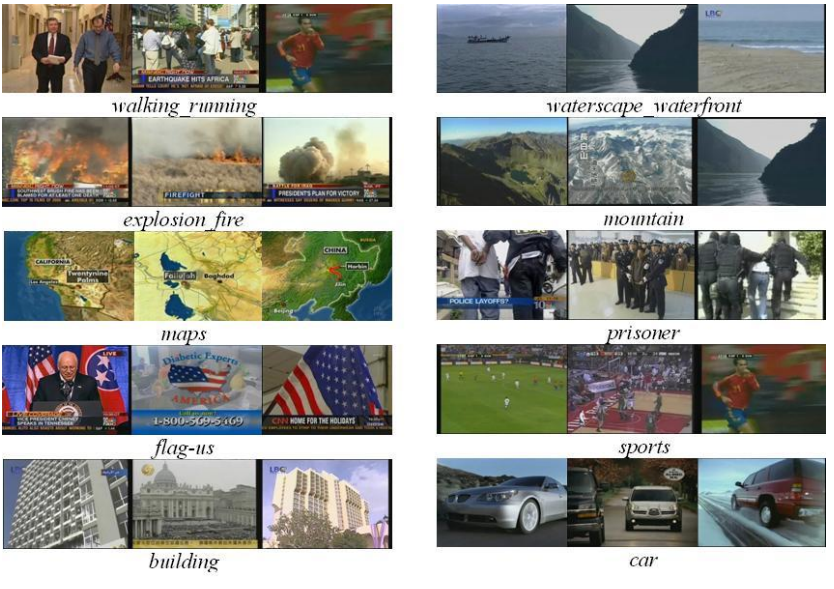


Fig. 1. The key-frame examples of the ten concepts

4 Experiments

In the following experiments, we use the video data set of the TRECVID05 corpus, which is consisted of about 170 hours of TV news videos from 13 different programs in English, Arabic and Chinese [13]. After automatic shot boundary detection, the development (DEV) set contains 43907 shots, and the evaluation (EVAL) set contains 45766 shots. Some shots are further segmented into sub-shots, and there are 61901 and 64256 sub-shots for DEV and EVAL set respectively.

The *high-level feature extraction* task is to detect the presence or absence of 10 predetermined benchmark concepts in each shot of the EVAL set. The 10 semantic concepts are *walking_running, explosion_fire, maps, flag-US, building, waterscape_waterfront, mountain, prisoner, sports* and *car* with concept IDs 1038∼1047. Some key-frame examples for these concepts are shown in Fig.1. For each concept, systems are required to return ranked-lists of up to 2000 shots, and system performance is measured via non-interpolated *mean average precision* (MAP), a standard metric for document retrieval.

The low level features we used here are 225-D block-wise color moments, which are extracted over 5×5 fixed grid partitions, each block is described using a 9-D feature.

Using the Kernel LNP method, the 64256 sub-shots are labeled as $f(subshot_i)$, and the sub-shots in the same shot are merged using the "max" rule:

$$f(shot_m) = max_{subshot_i \in shot_m}(f(subshot_i)) \tag{13}$$

Then the shots can be ranked according to $f(shot_m)$.

We compared Kernel LNP with LNP and SVM, and the experimental results are shown in Fig.2. The evaluations are accomplished when all the parameters are tuned to be nearly optimal by cross validations. Comparing these results, we can see that Kernel LNP significantly outperforms LNP for video high-level feature extraction, except that the result of *prisoner* is a little worse. The main reason is that *prisoner* is too difficult to be detected and the results of all approaches are nearly random. Kernel LNP outperforms SVM for detecting *maps*, *flag-US*, *waterscape_waterfront*, *mountain*, *prisoner* and *sports*, and remarkably outperforms the results named *Median* [14] (i.e., the average results of all the participants in TRECVID05). The MAP of Kernel LNP is 0.2303, which has an improvement of 4.7% and 64.2% over SVM and LNP, respectively. These

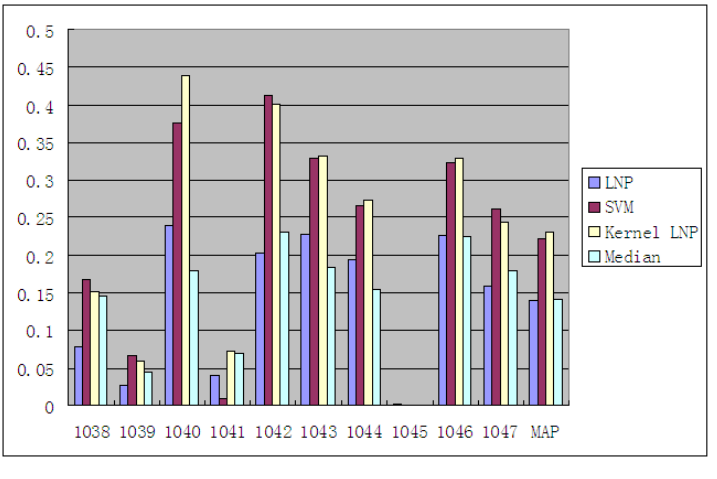


Fig. 2. The results comparisons among LNP, SVM, Kernel LNP and Median

comparisons demonstrate that Kernel LNP is more appropriate than LNP and is effective for semantic video annotation.

5 Conclusion and Future Work

We have analyzed the linear limitation of local semantics for LNP on ranking data with complex distribution, and proposed an improved method named Kernel LNP, in which a nonlinear kernel-mapped space is introduced for reconstruction coefficients optimization. The experiments conducted on the TRECVID dataset demonstrate that the proposed method is more appropriate for the data with complex distribution and is effective for the semantic video annotation task. However, this method needs a complex cross validation procedure to obtain an optimal set of parameters. Our next-step work will be focused on reducing computation cost brought by this procedure.

References

1. O. Chapelle, A. Zien, B. Scholkopf: Semi-supervised Learning, MIT Press, 2006.
2. R. Yan, M. Naphade: Semi-supervised Cross Feature Learning for Semantic Concept Detection in Videos. In: IEEE Conference on Computer Vision and Pattern Recognition, San Diego, July, 2005
3. M. Wang, X.-S. Hua, Y. Song, X. Yuan, S. Li, H.-J. Zhang: Automatic Video Annotation by Semi-supervised Learning with Kernel Density Estimation. In: ACM Multimedia, Santa Barbara, October, 2006
4. N. Grira, M. Crucianu, N. Boujemaa: Semi-Supervised Image Database Categorization using Pairwise Constrains. In: IEEE International Conference on Image Processing, Genova, September, 2005
5. D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Scholkopf: Learning with Local and Global Consistency. In: Advances in Neural Information Processing System, 2003
6. S.T.Roweis, L.K.Saul: Nonlinear Dimensionality Reduction by Locally Linear Embedding. In: Science, vol 290, 2000, 2323–2326
7. M. Belkin, P. Niyogi: Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In: Advances in Neural Information Processing System, 2000
8. F. Wang, J. Wang, C. Zhang, H. C. Shen: Semi-Supervised Classification Using Linear Neighborhood Propagation. In: IEEE Conference on Computer Vision and pattern Recognition, New York City, June, 2006
9. F. Wang, C. Zhang: Label Propagation through Linear Neighborhoods. In: 23rd International Conference on Machine Learning, Pittsburgh, June, 2006
10. L.K. Saul, S.T. Roweis: Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. In: Journal of Machine Learning Research, 2003, 119-155
11. M. Belkin, I. Matveeva, P. Niyogi: Regularization and Semisupervised Learning on Large Graphs. In: Conference on Learning Theory, 2004
12. P. Perona, J. Malik: Scale-Space and Edge Detection Using Anisotropic Diffusion. In: IEEE Transaction on Pattern Analysis and Machine Intelligence, 12(7), 1990
13. Guidelines for the TRECVID 2005 Evaluation. http://www-nlpir.nist.gov/projects/tv2005/tv2005.html
14. P. Over, T. Ianeva, W. Kraaij, A. F. Smeaton: TRECVID 2005 - An Overview. In: TREC Video Retrieval Evaluation Online Proceedings, NIST, 2005

Learning Bayesian Networks with Combination of MRMR Criterion and EMI Method

Fengzhan Tian[1], Feng Liu[2], Zhihai Wang[1], and Jian Yu[1]

[1] School of Computer & Information Technology,
Beijing Jiaotong University, Beijing 100044, China
{fzhtian,zhhwang,jianyu}@bjtu.edu.cn
[2] School of Computer science & Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China
lliufeng@hotmail.com

Abstract. Currently, learning Bayesian Networks (BNs) from data has become a much attention-getting issue in fields of machine learning and data mining. While there exists few efficient algorithms for learning BNs in presence of incomplete data. In this paper, we present a scoring function based on mutual information for evaluating BN structures. To decrease computational complexity, we introduce MRMR criterion into the scoring function, which enables the computation of the scoring function to involve in only two-dimensional mutual information. When the dataset is incomplete, we use EMI method to estimate the Mutual Information (MI) from the incomplete dataset. As for whether a node ordering is manually given or not, we develop two versions of algorithms, named as MRMR-E1 and MRMR-E2 respectively and evaluate them through experiments. The experimental results on Alarm network show good accuracy and efficiency of our algorithms.

1 Introduction

In recent years, learning BNs from data has become an attractive and active research issue in fields of machine learning and data mining. At present, there has been many successful algorithms used for learning BN structures from complete data, which include search & scoring based algorithms such as Bayesian method[1], Minimal Description Length (MDL) based method [2], dependency analysis based algorithms such as SGS[3], PC[3], TPDA[4], SLA[4] as well as the combination of the two kinds of algorithms mentioned above[5].

But when the training data is incomplete (i.e. containing missing values), BN learning becomes much more difficult. Now there has been some advances in learning BNs from incomplete data. One earlier research on BNs learning from incomplete data was made by Chickering[1], which deals with missing data by Gibbs sampling method. The most significant advance was the SEM algorithm[6]. SEM algorithm turns the learning problem with incomplete data into the easy solving learning problem with complete data using EM algorithm. Another research was made by W. Myers et al. They complete the incomplete data using generic operations, and evolve the network structures and missing data at the same time[7]. Using the idea of Friedman and Myers for reference,

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 801–808, 2007.

we put forward EM-EA algorithm[8], which completes data using EM algorithm and evolves BNs using an evolutionary algorithm (EA). And we also presented a method, named EMI, for estimating (conditional) Mutual Information from incomplete data and described an Extended TPDA (E-TPDA) algorithm which could learn BN structures from incomplete data[9].

Although E-TPDA is more efficient than SEM and EM-EA algorithms, it needs to calculate conditional mutual information, which is not reliable when the dataset is not big enough. Furthermore, calculating conditional mutual information and finding small d-separation set in E-TPDA algorithm also lead to high computation cost. In this paper, we present a scoring function based on mutual information. To decrease computational complexity, we introduce Max-Relevance and Min-Redundancy (MRMR) criterion into the scoring function. When the dataset is incomplete, we use EMI method to estimate the Mutual Information (MI) from the incomplete dataset. As for whether a node ordering is manually given or not, we develop two versions of algorithms, named as MRMR-E1 and MRMR-E2 respectively. Finally, through experiments on Alarm network, we compare MRMR-E1 with K2 and E-TPDA algorithms given a node ordering and compare MRMR-E2 with CB and E-TPDA algorithms without given a node ordering.

2 A Scoring Function with Combination of MRMR Criterion

Suppose that P is the underlying true distribution over variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ and Q is a distribution over these variables defined by some Bayesian network model, then the K-L cross entropy between P and Q, $C_{KL}(P, Q)$, is a distance measure of how close Q is to P and is defined by the following equation:

$$C_{KL}(P, Q) = \sum_{\mathbf{x}} P(\mathbf{x}) log_2 \frac{P(\mathbf{x})}{Q(\mathbf{x})} \tag{1}$$

The goal of BN learning is to search for the BN model with minimal $C_{KL}(P, Q)$ [2]. And also in [2], Lam and Bacchus proved the following theorem:

Theorem 1. *The K-L cross entropy $C_{KL}(P, Q)$ is a monotonically decreasing function of $\sum_{i=1}^{n} I(X_i, \mathbf{\Pi}_i)$. Hence, it will be minimized if and only if the sum is maximized.*

Where $\mathbf{\Pi}_i$ denotes the set of parents of X_i in a BN. Each element of the sum, $I(X_i, \mathbf{\Pi}_i)$, measures the mutual information of a local structure, X_i and its parents $\mathbf{\Pi}_i$, and is defined by the following equation:

$$I(X_i, \mathbf{\Pi}_i) = \sum_{x_i, \pi_i} P(x_i, \pi_i) log \frac{P(x_i, \pi_i)}{P(x_i) P(\pi_i)} \tag{2}$$

Where x_i and π_i respectively represent the value of X_i and the instantiation of $\mathbf{\Pi}_i$. According to the above theorem, we can use $\sum_{i=1}^{n} I(X_i, \mathbf{\Pi}_i)$ as a scoring function of BN models and search for the BN model with maximal value of $\sum_{i=1}^{n} I(X_i, \mathbf{\Pi}_i)$. Given the node ordering that means all the parents of any variable can only occur to the left of variable in the ordering, maximizing each $I(X_i, \mathbf{\Pi}_i)$ is equivalent to maximizing

$\sum_{i=1}^{n} I(X_i, \mathbf{\Pi}_i)$. Following this idea, $I(X_i, \mathbf{\Pi}_i)$ can be used as a local scoring function of a BN, and BN learning can be implemented by determining a set of parents for each variable on condition that the mutual information between the variable and its parents gets to the maximum. Unfortunately, despite the theoretical computability of the local score, the computational cost is very high because of the difficulty in getting an accurate estimation for multivariate probability distribution $P(x_i, \pi_i)$.

Peng et al called it Max Dependency criterion maximizing the mutual information between a variable and a variable set in their study of feature selection and proposed an alternative criterion for selecting features, named Max-Relevance and Min-Redundancy (MRMR) criterion[10], which can be formalized as follows:

$$\Phi(\mathbf{F}) = \frac{1}{|\mathbf{F}|} \sum_{X_i \in \mathbf{F}} I(X_i, C) - \frac{1}{|\mathbf{F}|^2} \sum_{X_i, X_j \in \mathbf{F}} I(X_i, X_j) \tag{3}$$

Where $\mathbf{F}$ represents the feature set, and C is the goal variable. Suppose already having $\mathbf{F}_{m-1}$, the feature set with $m-1$ features, the m^{th} feature to be selected from the set $\mathbf{X} - \mathbf{F}_{m-1}$ is the feature that maximizes the following equation:

$$I(X_j, C) - \frac{1}{m-1} \sum_{X_i \in \mathbf{F}_{m-1}} I(X_j, X_i), X_j \in \mathbf{X} - \mathbf{F}_{m-1} \tag{4}$$

This type of feature selection that adds one feature at one time is called the "first-order" incremental search, which can be used to find the near-optimal features defined by $\Phi(.)$. Peng et al proved the following theorem[10]:

Theorem 2. *For the first-order incremental search, MRMR is equivalent to Max Dependency.*

According to the above theorem, we can use first-order incremental style of MRMR criterion to find all the local structures of a BN, which is equivalent to the local scoring function defined by Eq. (2). When there exists missing values in dataset, we use EMI method to estimate each $I(X_j, X_i)$, the details of EMI method can be found in literature [9].

3 Learning Procedures with Combination of MRMR Criterion and EMI Method

3.1 Learning Procedure of MRMR-E1

Given a node ordering, we use the idea of K2 algorithm for reference and develop MRMR-E1 algorithm. MRMR-E1 adds incrementally those nodes among the predecessors in the ordering to the parent set of a node according to MRMR criterion and stops the adding process when no additional single parent can satisfy MRMR criterion. The details of MRMR-E1 algorithm is described as follows:

Input: a dataset D and a node ordering
Output: all local structures $\mathbf{\Pi}_i$;

For each node $X_i (1 \leq i \leq n)$, find $\mathbf{\Pi}_i$ as follows:
$\mathbf{\Pi}_i = \phi$; NotDone = True;
WHILE NotDone and $|\mathbf{\Pi}_i| < u_i$ do
$V = max\{MI[X_l, X_i] - \frac{1}{|\mathbf{\Pi}_i|-1} \sum_{X_j \in \mathbf{\Pi}_i} MI[X_l, X_j] | X_l \in \mathbf{\Pi}_i^c - \mathbf{\Pi}_i\}$;
Set Z is the node that makes V get to the maximum;
If $V > 0$, then $\mathbf{\Pi}_i = \mathbf{\Pi}_i \cup \{Z\}$,
else NotDone = false;
End {**WHILE**};

In the above procedure, $\mathbf{\Pi}_i^c$ is a set of the parent candidates of X_i; $MI[n, n]$ is a matrix storing mutual information; u_i denotes the max number of parents of X_i. They can be calculated by the following procedure:

Initial $MI[n, n]$ to 0;
Set $u_i = 0$ for all $i = 1, \ldots, n$;
Set $\mathbf{\Pi}_i^c = \phi$ for all $i = 1, \ldots, n$;
For each pair $(i, j)(1 \leq i \leq n; i < j \leq n)$, do the following:

$MI[i, j] = MI[j, i] = I'(x_i, x_j)$;
If $MI[i, j] \geq \sigma$, then
$u_j = min\{u_j + 1, u_{max}\}$;
$\mathbf{\Pi}_j^c = \mathbf{\Pi}_j^c \cup \{X_i\}$;

Where u_{max} is the max number of parents of all the nodes; σ is the threshold of the mutual information, which can be set as a small positive number in reality. $I'(x_i, x_j)$ is estimated mutual information, which can be calculated by EMI method[9].

3.2 Learning Procedure of MRMR-E2

To avoid the overdependence of the node ordering in K2 algorithm, Singh and Valtorta presented an algorithm, called CB[11], which can use CI tests to generate a node ordering. CB algorithm basically consists of two phases: Phase I uses CI tests to generate an undirected graph, and then orients the edges to get an ordering on the nodes; Phase II takes as input a total ordering consistent with the DAG generated by phase I, and applies the K2 algorithm to construct the network structure.

Using the Phase I of CB algorithm, we extend MRMR-E1 and present another version of algorithm, namely MRMR-E2. MRMR-E2 use the total ordering generated in Phase I of CB algorithm as the input of MRMR-E1 algorithm. The procedure of MRMR-E2 algorithm is as follows.

Input: a dataset D;
Output: all local structures $\mathbf{\Pi}_i$;

Let G_1 be the complete graph on the set of nodes $\mathbf{X}$;
ord = 0; old_Prob = 0; $old\_\mathbf{\Pi}_i = \phi$, for all $1 \leq i \leq n$;
REPEAT
Modify G_1 and get a total node ordering following Step 2 to Step 7 in CB algorithm [11];
Perform MRMR-E1 procedure to get all the local structures $\mathbf{\Pi}_i, 1 \leq i \leq n$ using the above ordering;

$new\_Prob = \prod_{i=1}^{n} g(X_i, \mathbf{\Pi}_i)$;
If $new\_Prob > old\_Prob$ then
$ord = ord + 1$; $old\_Prob = new\_Prob$;
$old\_\mathbf{\Pi}_i = \mathbf{\Pi}_i$, for all $1 \leq i \leq n$;
UNTIL $new\_Prob \leq old\_Prob$
$\mathbf{\Pi}_i = old\_\mathbf{\Pi}_i$, for all $1 \leq i \leq n$;
Output $\mathbf{\Pi}_i$, for all $1 \leq i \leq n$.

In the above procedure, ord denotes the order of CI relations being tested, and each **REPEAT** circulation tests each order of CI relations, first for 0^{th} order CI relations, then for 1^{st} order CI relations, and so on until the score of the learned network does not increase any more. $g(X_i, \mathbf{\Pi}_i)$ is a scoring function for local structures of a BN, which can be given by the following equation when a uniform priori is adopted[12]:

$$g(X_i, \mathbf{\Pi}_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \tag{5}$$

In the above equation, q_i is the number of particular instantiations of $\mathbf{\Pi}_i$ and r_i is the number of values that variable X_i can take; N_{ijk} represents the times of the occurrence of (x_i^k, π_i^j) in the dataset; x_i^k and π_i^j respectively represent the k^{th} value of X_i and j^{th} instantiation of $\mathbf{\Pi}_i$. $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

4 Experimental Analysis

For evaluating our algorithms, we compare MRMR-E1 with K2 and E-TPDA algorithms given a node ordering and compare MRMR-E2 with CB and E-TPDA algorithms without given a node ordering. Because K2 and CB algorithms can't learn BNs from incomplete datasets, they simply delete the cases with missing values in datasets and learn BNs from relative small and complete datasets in reality. We do this is to show the effects of the cases with missing values to the accuracy of learned BNs. We implement K2, CB, E-TPDA and our algorithms in a Java based system. All the experiments were conducted on a Pentium 3.2 GHz PC with 1G MB of RAM running under Windows 2003. The experimental outcome is the average of the results run 5 times for each level of missing data.

4.1 Experimental Results Given a Node Ordering

The given total ordering on the 37 nodes is consistent with the partial order of the nodes as specified by Alarm network. The threshold of Mutual Information in E-TPDA and MRMR-E1 algorithms is set as 0.01.

Table 1 and Table 2 show respectively the learning accuracy and the running time (Seconds) of K2, E-TPDA and MRMR-E1 given a node ordering. The "A+B" in Table 1 represents there are A extra arcs and B missing arcs in the learned networks compared with the true network. We treat a reversed arc as if there are a missing arc and an extra arc. From this table, we can see that the accuracy of K2 algorithm degrades very sharply with the increase of missing data, which states that simply deleting the data records

with missing entries will lose large amount of useful information. And also the learned networks by MRMR-E1 are more accurate than that by E-TPDA algorithm for 10,000 and 20,000 cases and equal accurate to that by E-TPDA algorithm for 40,000 cases. That MRMR-E1 outperforms E-TPDA for 10,000 and 20,000 cases is because that the calculation of conditional mutual information in E-TPDA algorithm is not reliable for relative small datasets.

Table 1. Learning accuracy of K2, E-TPDA and MRMR-E1 algorithms on Alarm network given a node ordering

Sample Size	Algorithms	Complete Data	10% Missing	20% Missing	30% Missing
10,000	K2	1+1	18+3	32+5	52+7
	E-TPDA	2+1	3+2	4+3	7+3
	MRMR-E1	1+1	3+1	3+2	5+2
20,000	K2	1+1	12+3	26+4	41+6
	E-TPDA	2+1	2+2	2+2	3+3
	MRMR-E1	1+1	2+1	2+1	2+2
30,000	K2	1+0	8+2	17+3	32+5
	E-TPDA	2+0	2+0	2+0	2+2
	MRMR-E1	1+0	2+0	2+0	2+1
40,000	K2	1+0	6+1	11+2	22+4
	E-TPDA	1+0	1+0	2+0	2+1
	MRMR-E1	1+0	1+0	2+0	2+1

From Table 2, the running time of K2 algorithm decreases very quickly with the increase of missing data because K2 runs on very small datasets after deleting the data records with missing entries. While fixing the percentage of missing data, the running time of E-TPDA and MRMR-E1algorithms is roughly linear to the size of the data samples. Nevertheless, MRMR-E1 is more efficient than E-TPDA, whose running time is about half of that of E-TPDA algorithm. This also proves that calculating conditional mutual information and finding small d-separation set in E-TPDA algorithm lead to high computation cost.

4.2 Experimental Results Without Given a Node Ordering

In our experiments, CB algorithm also uses the partial order and a bound of 15 on the maximal degree of the undirected graph just as in literature[11].

Table 3 and Table 4 show respectively the learning accuracy and running time of CB, E-TPDA and MRMR-E2 without given a node ordering. The experimental results in Table 3 and Table 4 are similar respectively to that in Table 1 and Table 2, which also show that MRMR-E2 is advantageous over E-TPDA algorithm in terms of accuracy and efficiency without given a node ordering. Compared with that in Table 1, the accuracy showed in Table 3 decreases to some extent and the running time in Table 4 increases to some extent compared with that in Table 2. This is because, without given a node ordering, the three algorithms have to compute the partial order between variables or infer the orientation of an arc which decreases their accuracy and efficiency.

Table 2. Running time (Seconds) of K2, E-TPDA and MRMR-E1 algorithms on Alarm network given a node ordering

Sample Size	Algorithms	Complete Data	10% Missing	20% Missing	30% Missing
10,000	K2	128	6.2	4.3	2.1
	E-TPDA	28	29	28	29
	MRMR-E1	15	15	14	16
20,000	K2	259	8.5	5.7	3.2
	E-TPDA	54	54	55	55
	MRMR-E1	28	28	29	29
30,000	K2	398	10.8	6.1	4.4
	E-TPDA	78	79	78	79
	MRMR-E1	39	39	40	39
40,000	K2	532	15	12	5.8
	E-TPDA	102	102	101	102
	MRMR-E1	48	47	47	48

Table 3. Learning accuracy of CB, E-TPDA and MRMR-E2 algorithms on Alarm network without given a node ordering

Sample Size	Algorithms	Complete Data	10% Missing	20% Missing	30% Missing
10,000	CB	2+1	22+4	39+5	64+7
	E-TPDA	2+2	5+2	7+3	9+3
	MRMR-E2	2+1	4+1	5+2	7+3
20,000	CB	2+1	18+4	29+5	54+6
	E-TPDA	2+1	4+2	6+2	7+3
	MRMR-E2	2+1	3+1	4+1	5+2
30,000	CB	1+1	14+3	25+4	37+5
	E-TPDA	2+1	3+1	4+1	5+2
	MRMR-E2	1+1	2+1	2+2	4+2
40,000	CB	1+1	7+2	13+3	25+4
	E-TPDA	1+1	2+1	3+1	4+2
	MRMR-E2	1+1	2+1	3+1	3+2

Table 4. Running time (Seconds) of CB, E-TPDA and MRMR-E2 algorithms on Alarm network without given a node ordering

Sample Size	Algorithms	Complete Data	10% Missing	20% Missing	30% Missing
10,000	CB	189	8.9	6.1	3.1
	E-TPDA	45	46	46	47
	MRMR-E2	22	22	23	23
20,000	CB	368	13	8.4	4.6
	E-TPDA	82	84	82	83
	MRMR-E2	42	43	43	44
30,000	CB	549	21	12	7.4
	E-TPDA	118	119	118	119
	MRMR-E2	61	62	63	62
40,000	CB	724	29	20	10.8
	E-TPDA	152	151	152	153
	MRMR-E2	80	82	81	83

5 Conclusion

In this paper, we present a scoring function evaluating BN structures based on mutual information and introduce MRMR criterion into the scoring function to decrease computational complexity. When the dataset is incomplete, we use EMI method to estimate the Mutual Information (MI) from the incomplete dataset. As for whether a node ordering is manually given or not, we develop two versions of algorithms, named as MRMR-E1 and MRMR-E2 respectively. The experimental results on Alarm network show that MRMR-E1 and MRMR-E2 algorithms are advantageous over E-TPDA in terms of accuracy and efficiency whether given a node ordering or not. The experimental results also state that making fully use of data records with missing entries will significantly improve the accuracy of learned BNs.

Acknowledgment

This work is supported by NSF of China under grant NO. 60503017 and 60673089 as well as Science Foundation of Beijing Jiaotong University under Grant No. 2005SM012.

References

1. Chickering, D.M., Heckerman, D.: Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. Machine Learning **29**(2-3) (1997) 181–212
2. Lam, W., Bacchus, F.: Learning Bayesian belief networks: An approach based on the MDL principle. Computational Intelligence **10** (1994) 269–293
3. Spirtes, P., Glymour, C., Scheines, R.: Causation, prediction, and search. MIT Press, CAMBRIDGE, MA, USA, second edition (2001)
4. Cheng, J., Greiner, R., Kelly, J.: Learning Bayesian networks from data: an information-theory based approach. Artificial Intelligence Journal **137** (2002) 43–90
5. Ioannis, T., Laura, E., Constantin, F.A.: The max-min hill-climbing Bayesian network structure learning algorithm. Machine Learning **65**(1) (2006) 31–78
6. Friedman, N.: The Bayesian structural EM algorithm. Proceddings of 14th Conference on Uncertainty in Artificial Intelligence (1998)
7. Myers, J., Laskey, K., Levitt, T.: Learning Bayesian networks from incomplete data with stochastic search algorithms. Proceedings of the 15th Conference on UAI (1999)
8. Tian, F., Lu, Y., Shi, C.: Learning Bayesian networks with hidden variables using the combination of EM and evolutionary algorithm. Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (2001) 568–574
9. Tian, F., Zhang, H., Lu, Y.: Learning Bayesian networks from incomplete data based on EMI method. Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), Melbourne, Florida, USA (2003) 323–330
10. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(8) (2005) 1226–1238
11. Singh, M., Valtorta, M.: Construction of Bayesian network structures from data: a brief survey and an efficient algorithm. International Journal of Approximate Reasoning **12** (1995) 111–131
12. Cooper, G.F., Herskovits, E.A.: Bayesian method for the induction of probabilistic networks from data. Machine Learning **9** (1992) 309–347

A Cooperative Coevolution Algorithm of RBFNN for Classification

Jin Tian, Minqiang Li, and Fuzan Chen

School of Management, Tianjin University, Tianjin 300072, P.R. China
jtian_tju@yahoo.com.cn

Abstract. This article presents a new learning algorithm, CO-RBFNN, for complex classifications, which attempts to construct the radial basis function neural network (RBFNN) models by using a cooperative coevolutionary algorithm (Co-CEA). The Co-CEA utilizes a divide-and-cooperative mechanism by which subpopulations are coevolved in separate populations of evolutionary algorithms executing in parallel. A modified *K*-means method is employed to divide the initial hidden nodes into modules that are represented as subpopulation of the Co-CEA. Collaborations among the modules are formed to obtain complete solutions. The algorithm adopts a matrix-form mixed encoding to represent the RBFNN hidden layer structure, the optimum of which is achieved by coevolving all parameters. Experimental results on eight UCI datasets illustrate that CO-RBFNN is able to produce a higher accuracy of classification with a much simpler network structure in fewer evolutionary trials when compared with other alternative standard algorithms.

Keywords: RBFNN; cooperative coevolutionary algorithms; *K*-means clustering; classification.

1 Introduction

Different variants of radial basis function neural networks (RBFNN) are used for complex classification tasks due to a number of advantages compared with other types of artificial neural networks (ANN), such as better classification capabilities, simpler network structures, and faster learning algorithms.

The main difficulty in RBFNN configuration lies in the determination of the hidden layer structure. A variety of approaches based on evolutionary algorithms (EAs) have been developed [1,2], where an individual of the population corresponds to a candidate of the solution regardless of other individuals in the population. Therefore a large population size is required and the computation time is usually prohibitive. The cooperative coevolutionary algorithm (Co-CEA), which is capable of dealing with learning tasks with multiple, interacting subpopulations, is a powerful approach to solve the problem. An individual in a subpopulation receives fitness based on how well it performs in couple with individuals from other subpopulations. The Co-CEA has been successfully applied to complex optimization problems and recently to training the cascade networks [3] and the ANN ensemble [4].

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 809–816, 2007.

The proposed algorithm in this article, named CO-RBFNN, attempts to construct the RBFNN models using a specially designed Co-CEA. A modified *K*-means method is employed to divide the initial hidden nodes into modules that are represented as subpopulation of the Co-CEA. Then the modules are used to generate populations to carry on the coevolution operation.

The rest of the paper is structured as follows: Section 2 presents the fundamental theory of the Co-CEA and the RBFNN architecture. In Section 3, the proposed algorithm is described in detail. Section 4 illustrates the new algorithm's efficiency on eight UCI datasets. Finally, Section 5 summarizes the key points of the paper.

2 Fundamentals

2.1 Cooperative Coevolutionary Architecture

The architecture of Co-CEA model consists of two or more interacting co-adapted subpopulations. Each subpopulation contains individuals that represent a particular component of the RBFNN, so that one representative from each subpopulation is selected in order to assemble a complete solution. In this article, it is appropriate to simply let the current best individual from each subpopulation be the representative. Each subpopulation is evolved independently and adapts to the learning task through the application of an EA. Although any EA can be used in principle, coevolutionary versions of GA have always been employed.

2.2 RBFNN Architecture

The RBFNN can be viewed as a three-layer feed-forward neural network with multi-inputs, multi-outputs, and linear output mapping. The RBFNN structure defines the mapping from the input vector $\boldsymbol{x} \in \boldsymbol{R}^m$ to the output vector $\boldsymbol{y} \in \boldsymbol{R}^n : f : \boldsymbol{x} \to \boldsymbol{y}$. The response by a hidden node is produced through a radial basis function:

$$\varphi_j(\boldsymbol{x}) = \exp\left\{-\frac{\left\|\boldsymbol{x}-\boldsymbol{\mu}_j\right\|^2}{\sigma_j^2}\right\}, \tag{1}$$

where $\boldsymbol{\mu}_j$ is the center, and σ_j is the radius width of the j^{th} hidden node. The i^{th} output may be expressed as a linear combination: $y_i = \sum_{j=1}^{k} w_{ji}\varphi_j(\boldsymbol{x})$, where the weights, w_{ji}, are calculated by the pseudo-inverse method [5].

3 Configuration of RBFNN Using the Co-CEA

In this study, an additional clustering layer is introduced into the standard RBFNN construction. After the initial hidden nodes are generated, an alterative *K*-means approach is utilized to further group them into modules.

3.1 Matrix-Form Mixed Encoding

We design a matrix-form mixed encoding representation, which assigns the hidden nodes and the radius widths as real-valued encoding matrices and a control vector as a binary string. The l^{th} individual in the t^{th} module $\boldsymbol{C}_t^l = [\boldsymbol{c}_t^l \quad \boldsymbol{\sigma}_t^l \quad \boldsymbol{b}_t^l]$, $l = 1,2,\ldots,L$, $t = 1,2,\ldots K$. $\boldsymbol{c}_t^l = [\boldsymbol{c}_{ti}^l]_{num_t \times m}$ is the center of the hidden nodes, $\boldsymbol{\sigma}_t^l = [\sigma_{ti}^l]_{num_t \times 1}$ is the radius width and $\boldsymbol{b}_t^l = [b_{ti}^l]_{num_t \times 1}$ is the control vector, where $b_{ti}^l = 0$ means that the i^{th} hidden node is invalid, otherwise $b_{ti}^l = 1$ denotes that it is valid, $i = 1,2,\ldots,num_t$. num_t is the initial number of the hidden nodes in this module.

3.2 Initialization

The initialization of the algorithm is done in three steps. Firstly, the DRSC method [6] is used to compute the hidden nodes. Secondly, a modified *K*-means method decomposes these nodes into many modules. As the evolution proceeds, the *K* can be shrunk automatically. Finally, a population contained *L* individuals is generated and every individual in one module contains all or some of the hidden nodes in this module. And the *K*-means method mentioned above can be described as follows:

Step 1: The *K* samples are chosen randomly from the N_c initial hidden nodes φ_j as the initial centers of the modules, $\tilde{\boldsymbol{c}}_t^1$ ($t = 1,2,\ldots,K$). Counter *i* is assigned as 1.

Step 2: The distances between centers and samples, $D(\varphi_j, \tilde{\boldsymbol{c}}_t^i)$, are calculated. If $D(\varphi_j, \tilde{\boldsymbol{c}}_{t^*}^i) = \min_{t=1,\cdots,K} \{D(\varphi_j, \tilde{\boldsymbol{c}}_t^i), j = 1,2,\ldots,N_c\}$, then $\varphi_j \in \boldsymbol{M}_{t^*}$, where $\boldsymbol{M}_{t^*}$ is the set (or module) of the hidden nodes whose nearest clustering center is $\tilde{\boldsymbol{c}}_{t^*}^i$.

Step 3: If $\exists\ \boldsymbol{M}_s = \{\tilde{\boldsymbol{c}}_s^i\}$, $s \in \{1,2,\ldots,K\}$, which means there isn't other samples in the s^{th} set except for the center $\tilde{\boldsymbol{c}}_s^i$, $\tilde{\boldsymbol{c}}_s^i$ should be deleted, and the value of *K* be changed.

Step 4: The new clustering centers are re-calculated: $\tilde{\boldsymbol{c}}_t^{i+1} = \dfrac{1}{num_t} \sum_{\varphi_u \in \boldsymbol{M}_t} \varphi_u$.

Step 5: If centers of the modules are changed, or $i < I$ (the maximum iteration), then $i = i + 1$, and go to Step 2; else the algorithm terminates.

From this point of view, the introduction of the *K*-means clustering layer is to group the hidden nodes that have the similar characteristics, into the same module. The redundant hidden nodes in one module can be removed by modify the corresponding control vectors in the last step of the initialization.

3.3 Evaluation and Selection of the Individuals

The individuals in one module are assigned the fitness based on the cooperation with individuals in other modules. The best individual in each module is chosen as the representative, all of which compose the elite pool $\boldsymbol{\Theta}^* = \{\boldsymbol{C}_1^*, \boldsymbol{C}_2^*, \ldots, \boldsymbol{C}_K^*\}$. The individual fitness is evaluated as a multi-objective optimization task in this algorithm

because the solutions based on Pareto optimality can guarantee the diversity [7] of the population in evolution. Two objectives are selected here for the fitness evaluation:

1) Classification accuracy: $A_r(\boldsymbol{\Theta}_t^l) = N_s(\boldsymbol{\Theta}_t^l)/N_e$. The first objective is commonly used. $N_s(\boldsymbol{\Theta}_t^l)$ is the number of samples classified correctly in the evaluating set by the network, whose hidden nodes are $\boldsymbol{\Theta}_t^l = \{\boldsymbol{C}_1^*, \ldots, \boldsymbol{C}_{t-1}^*, \boldsymbol{C}_t^l, \boldsymbol{C}_{t+1}^*, \ldots, \boldsymbol{C}_K^*\}$. N_e is the size of the evaluating set. However, similar accurate rates give rise to less selection pressure in the population. So a modified objective is employed: $f_1(\boldsymbol{\Theta}_t^l) = \alpha(1-\alpha)^{I(\boldsymbol{\Theta}_t^l)}$, where $I(\boldsymbol{\Theta}_t^l)$ is the sort order of $\boldsymbol{\Theta}_t^l$'s classification accuracy after all accuracies are ranked from big to small, and α is a pre-designed real number between 0 and 1.

2) Shared performance: This objective enforces the networks to classify different patterns [8]. In this way, the individuals that contribute to the network to accurately classify samples that are incorrectly classified by many others are rewarded. Each sample in the evaluating set receives a weight, namely $w_i = Nc_i/(N_m \times L)$. Nc_i is the number of individuals that classify the i^{th} sample correctly in all modules, N_m is the number of modules. The value assigned to an individual for this objective is given by $f_2(\boldsymbol{\Theta}_t^l) = \sum_{i=1}^{N_e} e_{ti}^l \times (1-w_i)$, where e_{ti}^l is 1 if the i^{th} sample is correctly classified by the network constructed by $\boldsymbol{\Theta}_t^l$, and 0, otherwise.

Totally, the fitness of an individual $\boldsymbol{C}_t^l$ is calculated using an aggregating approach:

$$fit_t^l = a \times f_1(\boldsymbol{\Theta}_t^l) + b \times f_2(\boldsymbol{\Theta}_t^l) \ , \tag{2}$$

where a and b are the coefficients of objective importance, $a, b \in [0,1]$, and $a+b=1$.

The roulette wheel selection is utilized and the elitist selection [9] is adopted so that the best solution survives definitely from one generation to the next.

3.4 Crossover

The uniform crossover [9] is used in this paper. The individuals that undergo the crossover operation are grouped into pairs, and for every pair a mask binary string is generated randomly. Scanning the string from left to right, if the current bit is 1, the genes at the position in the first parent are selected; otherwise, the genes in the second one are selected. Thus one offspring is produced. The second offspring is produced by repeating the process but with the positions of 0 and 1 being exchanged in the string.

3.5 Mutation

The mutation is an auxiliary but also a significant operation. A ratio, p_{ad}, has been introduced to decide the mutation occurrence in the control bit or the real-valued part. For every component $\boldsymbol{c}_{ti}^l$ in $\boldsymbol{C}_t^l$, a random number r_{ad} is generated. If $p_{ad} > r_{ad}$, the operation only inverts the control bit. If $p_{ad} \le r_{ad}$ and $b_{ti}^l = 1$, the mutation operates in

the real-valued part as $\boldsymbol{c}_{ti}^{l\prime} = \boldsymbol{c}_{ti}^{l} + r_{ti}^{l} \cdot \left(\boldsymbol{c}_{ti}^{*} - \tilde{\boldsymbol{c}}_{ti}^{l}\right)$. $\boldsymbol{c}_{ti}^{l\prime}, \boldsymbol{c}_{ti}^{l}, \tilde{\boldsymbol{c}}_{ti}^{l}$ denote the mutated, present and history value respectively, and $\boldsymbol{c}_{ti}^{*}$ is the corresponding value in the elite pool.

4 Experimental Studies

We have applied our model to eight UCI datasets. For each dataset, 50% of the patterns were used for learning, 25% of them for validation, and the remaining 25% for testing. 30 runs of the algorithm were performed for each dataset.

4.1 Experiment 1

The experiments were carried out to test CO-RBFNN against some traditional training algorithms, such as GA-RBFNN [10] which evolves the RBFNN with a standard GA, DRSC, *K*-means, the probabilistic neural network (PNN) and the *K*-nearest neighbor algorithm (KNN). These methods generate RBFNN without a validation set, adding the validation set to the training set, except GA-RBFNN. Therefore, the valuation accuracies are omitted in the tables below for the simplicity in comparison.

The experiment parameters were set as follows. The size of subpopulation, *L*, was 50, the number of modules, *K*, was 4, and the number of generations, *G*, was 200. The crossover rate p_c was 0.6, the non-structure mutation rate p_m was 0.2, and the structure mutation rate p_{ad} was 0.2. The average accuracies, the hidden node numbers, and the iterations of convergence are shown in Table 1.

Table 1. Comparison with other algorithms for eight UCI datasets. The *t*-test compares the average testing accuracy of CO-RBFNN with that of each of the other used algorithm.

Methods		Cancer	Glass	Heart	Iono	Pima	Soy	Vowel	Wines	Ave	N_c
CO-RBFNN	Train	0.9625	0.7449	0.8532	0.9397	0.7807	0.9465	0.8940	0.9685	*0.8863*	30.35
	Test	0.9694	0.7107	0.8300	0.9377	0.7698	0.9265	0.7501	0.9689	*0.8579*	
	Gen	10.93	60.60	18.07	21.37	81.40	61.20	200	38.13	*61.46*	
GA-RBFNN	Train	0.9624	0.7277	0.8361	0.9346	0.7747	0.9342	0.8523	0.9727	*0.8743*	30.92
	Test	0.9689	0.6893	0.8126	0.9240	0.7625	0.9065	0.7331	0.9682	*0.8456*	
	Gen	200	200	200	200	200	200	200	200	*200*	
	t-test	0.3739	2.075	1.864	1.652	0.9863	4.376	2.689	0.4712	-	
DRSC	Train	0.9653	0.8121	0.8693	0.9330	0.7897	0.9488	0.7715	0.9749	*0.8831*	43.03
	Test	0.9667	0.6157	0.8224	0.9227	0.7372	0.8723	0.6696	0.9583	*0.8206*	
	t-test	0.8872	6.391	1.592	2.085	3.931	10.09	11.60	1.369	-	
K-means	Train	0.9641	0.7362	0.8424	0.8958	0.7609	0.6769	0.5408	0.9744	*0.7989*	30
	Test	0.9634	0.6610	0.8054	0.8970	0.7415	0.6545	0.4820	0.9667	*0.7714*	
	t -test	1.248	4.018	2.158	3.551	4.188	38.36	40.40	0.3214	-	
PNN	Train	1.000	1.000	1.000	1.000	1.000	0.9988	1.000	1.000	*0.9999*	362.8
	Test	0.9494	0.6686	0.7309	0.9443	0.6950	0.7910	0.9558	0.9447	*0.8350*	
	t -test	5.736	3.397	9.080	-0.9833	9.948	22.71	-42.09	3.207	-	
KNN	Train	0.9736	0.7863	0.8653	0.8203	0.8073	0.9303	0.8898	0.9756	*0.8811*	362.8
	Test	0.9686	0.6566	0.8000	0.7871	0.7181	0.8982	0.7795	0.9697	*0.8222*	
	t -test	0.6982	3.878	3.065	18.12	6.964	6.394	-4.675	-0.1134	-	

The results illuminate that CO-RBFNN yielded classification accuracies close to the best one for most problems and produced a smaller network structure than the compared algorithms. The average hidden node sizes obtained by CO-RBFNN were only 70.53% of the original by DRSC. Moreover, the averaged iterations needed to convergence were 61.46, which were quite smaller compared with the 200 iterations needed by GA-RBFNN. Compared with many trials in the *K*-means to search a suitable solution, only one run is needed since CO-RBFNN can design the network structure dynamically. CO-RBFNN has both a satisfying accuracy for training and a comparatively high accuracy for testing while PNN and KNN need a big number of hidden nodes. In terms of the *t*-test results, CO-RBFNN outperforms significantly the other methods in Glass, Pima and Soybean with a confidence level of 95%.

Moreover, the results obtained are better when they are compared with other works using these datasets. Table 2 shows a summary of the results reported in literatures by other classification methods. Although the experimental setups are different in many papers, comparisons are still illustrative. Some of the papers use tenfold cross-validation and obtain more optimistic estimations on some problems. However, we didn't utilize tenfold cross-validation because it does not fit to the triplet samples partition. With these cautions, we can say that for Cancer, Heart, Ionosphere, Pima, Soybean and Wines datasets our algorithm achieves a better or similar performance.

Table 2. Results of previous works using the same datasets

Datasets	CO-RBFNN	[11][1]	[12][2]	[13][1]	[14][1]	[15][2]	[16][1]	[17][1]	[18][1]
Cancer	0.9694	0.9580	-	0.9470	0.9620	-	0.9650	0.9780	0.9490
Glass	0.7107	0.7050	0.6856	0.6710	0.7620	0.6837	0.7510	0.7050	0.7000
Heart	0.8300	0.8370	0.8249	-	-	-	0.8030	0.8370	0.7890
Iono	0.9377	0.8970	0.9789	-	0.9370	0.8817	-	0.9310	0.9060
Pima	0.7698	0.7720	-	0.7400	-	0.6872	0.7560	0.7430	0.7400
Soy	0.9265	0.9250	-	-	-	-	0.9300	0.8100	0.9140
Vowel	0.7501	0.8170	-	-	0.4830	-	-	0.6520	0.7810
Wines	0.9689	-	-	0.7920	-	0.9444	-	0.9290	-

[1] k-fold cross-validation; [2] hold out

4.2 Experiment 2

The main difference between CO-RBFNN and GA-RBFNN is the introduction of the Co-CEA. The modified *K*-means combines the CEA frame with RBFNN. Thus the selection of the initial value of *K* is very important although it can be changed dynamically in the evolution process. We carried out another experiment by assigning different values, i.e., a set of 2,4,6,8 and 10, to *K* over four datasets. The other parameters were assigned as the same as experiment 1. Fig. 1 gives a description about the trend of the averaged testing accuracies and the iterations of convergence.

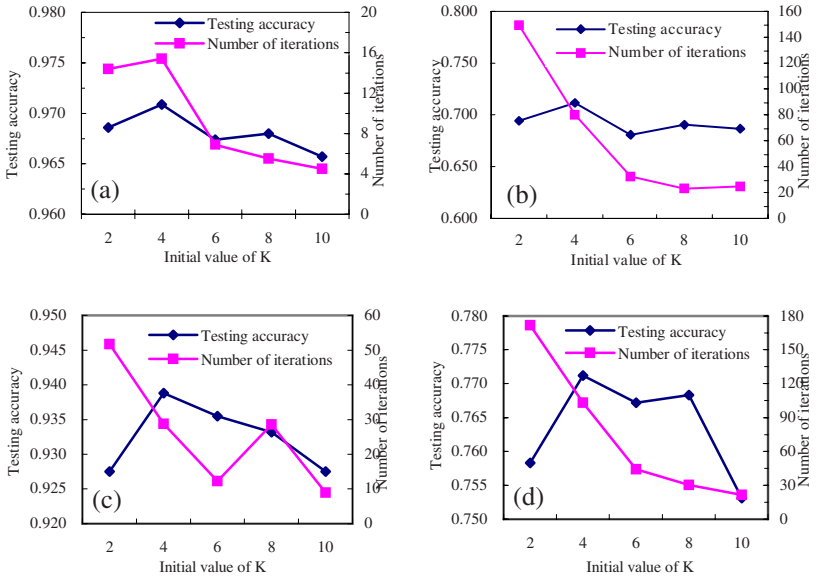


Fig. 1. Average testing accuracy and number of iterations obtained with different initial value of *K* in four datasets: (a) Cancer (b) Glass (c) Ionosphere (d) Pima

The experiment results show that the testing accuracies in all the four datasets get to the maxima when K=4. Generally, a smaller K can lead the performance of CO-RBFNN similar to GA-RBFNN so that the convergence is not easy to realize. On the other hand, although the algorithm can be speeded up with a lager K, there will appear premature convergence. Thus the testing accuracy does not increase along with the increment of K. Overall, a too small or too large initial value of K is not good for increasing both the testing accuracy and the convergence speed.

5 Conclusions

A special designed RBFNN algorithm, CO-RBFNN, has been presented, in which the introduction of the Co-CEA realizes the coevolution of the subpopulations in separate populations of GA executing in parallel. By evolving the hidden nodes and corresponding parameters simultaneously, the model is able to produce a higher accuracy of classification with a network which is much simpler in structure but stronger in generalization capability compared with other training algorithms. To sum up, CO-RBFNN is a quite competitive and powerful algorithm for classification. Two points should be concerned in our future work: one is the auto-division of modules by combining some clustering methods without human involvement and the other is the introduction of some new objectives of fitness.

Acknowledgments. The work was supported by the National Science Foundation of China (Grant No.70171002, No. 70571057) and the Program for New Century Excellent Talents in Universities of China (NCET).

References

1. Blanco, A., Delgado, M., Pegalajar, M.C.: A Real-Coded Genetic Algorithm for Training Recurrent Neural Networks. Neural Networks, Vol. 14. (2001) 93-105
2. Sarimveis, H., Alexandridis, A., et al: A New Algorithm for Developing Dynamic Radial Basis Function Neural Network Models Based on Genetic Algorithms. Computers and Chemical Engineering, Vol. 28. (2004) 209-217
3. Potter, M., De Jong, K.: Cooperative Coevolution: an Architecture for Evolving Coadapted Subcomponents. Evolutionary Computation, Vol. 8. (2000) 1-29
4. García-Pedrajas, N., Hervás-Martínez, C., Ortiz-Boyer, D.: Cooperative Coevolution of Artificial Neural Network Ensembles for Pattern Classification. IEEE transactions on evolutionary computation, Vol. 9. (2005) 271-302
5. Casasent, D., Chen, X.W.: Radial Basis Function Neural Networks for Nonlinear Fisher Discrimination and Neyman-Pearson Classification.Neural Networks,Vol.16.(2003)529-535
6. Berthold, M.R., Diamond, J.: Boosting the Performance of RBF Networks with Dynamic Decay Adjustment. In: Tesauro, G., Touretzky, D.S., Leen, T.K. (eds.): Advances in Neural Information Processing Systems, Vol. 7. MIT Press, Denver Colorado (1995) 512-528
7. Bosman, P.A.N., Thierens, D.: The Balance between Proximity and Diversity in Multiobjective Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, Vol. 7. (2003) 174-188
8. Liu, Y., Yao, X., Higuchi, T.: Evolutionary Ensembles with Negative Correlation Learning. IEEE Transactions on Evolutionary Computation, Vol. 4. (2000) 380–387
9. Li, M.Q., Kou, J.S., *et al*: The Basic Theories and Applications in GA. Science Press, Beijing (2002)
10. Tian, J., Li, M.Q., Chen, F.Z.: GA-RBFNN Learning Algorithm for Complex Classifications. Journal of Systems Engineering, Vol. 21. (2006) 163-170
11. Frank, E., Wang, Y., Inglis, S., *et al*: Using Model Trees for Classification. Machine Learning, Vol. 32. (1998) 63-76
12. Merz, C.J.: Using Correspondence Analysis to Combine Classifiers. Machine Learning, Vol.36.(1999) 33-58
13. Cantú-Paz, E., Kamath, C.: Inducing oblique Decision Trees with Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, Vol. 7. (2003) 54-68
14. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: A Statistical View of Boosting. Annals of Statistics, Vol.28. (2000) 337-407
15. Draghici, S.: The Constraint Based Decomposition (CBD) Training Architecture. Neural Networks, Vol. 14. (2001) 527-550
16. Webb, G.I.: Multiboosting: A Technique for Combining Boosting and Wagging. Machine Learning, Vol. 40. (2000) 159-196
17. Yang, J., Parekh, R., Honavar, V.: DistAI: an Inter-pattern Distance-based Constructive Learning Algorithm. Intelligent Data Analysis, Vol. 3. (1999) 55-73
18. Frank, E., Witten, I.H.: Generating Accurate Rule Sets Without Global Optimization. In: Shavlik, J.W. (eds.): Proc. of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco CA (1998) 144-151

ANGEL: A New Effective and Efficient Hybrid Clustering Technique for Large Databases

Cheng-Fa Tsai and Chia-Chen Yen

Department of Management Information Systems,
National Pingtung University of Science and Technology,
91201 Pingtung, Taiwan
{cftsai,m9556001}@mail.npust.edu.tw

Abstract. This paper presents a new clustering algorithm named ANGEL, capable of satisfying various clustering requirements in data mining applications. As a hybrid method that employs discrete-degree and density-attractor, the proposed algorithm identifies the main structure of clusters without including the edge of clusters and, then, implements the DBSCAN algorithm to detect the arbitrary edge of the main structure of clusters. Experiment results indicate that the new algorithm accurately recognizes the entire cluster, and efficiently solves the problem of indentation for cluster. Simulation results reveal that the proposed new clustering algorithm performs better than some existing well-known approaches such as the K-means, DBSCAN, CLIQUE and GDH methods. Additionally, the proposed algorithm performs very fast and produces much smaller errors than the K-means, DBSCAN, CLIQUE and GDH approaches in most the cases examined herein.

Keywords: data clustering, data mining, hybrid clustering algorithm.

1 Introduction

Clustering in data mining is essential for various business applications. Numerous data clustering schemes have been proposed in recent years, subsequently attracting strong attention [1]-[6]. Many existing clustering methods have high computational time, or may have pattern recognition problems when using large databases. Therefore, an efficient and effective clustering algorithm is important. Clustering approaches can be categorized as partitioning, hierarchical, density-based, grid-based and mixed. Partitioning methods like K-means attempt to identify K partitions containing similar objects [3]. K-means algorithm is easily and quickly implemented, but does not accurately recognizing the shapes of patterns that are non-spherical or not the same size. DBSCAN, density-based clustering method, measures the density of a region, and thus accurately recognizes arbitrary shapes and different size clusters, and filters noise [4]. However, DBSCAN needs to examine all objects, and thus has a high computational time. Grid-based clustering methods define clusters using grid-cell structures. These methods consider the grid-cell as a point to improve the problem of time cost, and

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 817–824, 2007.

can therefore cluster all objects quickly. However, they cannot smoothly detect the edges of clusters, or remove indentations in neighboring clusters. CLIQUE is a classical grid-based clustering method [5].

To fulfill data clustering requirements and solving limitations of the above clustering methods, this work presents a new algorithm named **A N**ew **G**rid-based clustering method with **E**liminating indention for **L**arge databases (**ANGEL**) by hybridizing hierarchical, density-based and grid-based clustering approaches. Simulation results show that the proposed ANGEL approach is a highly effective and efficient clustering technique.

2 Related Works

K-means, which was presented in 1967, was the first clustering algorithm [3]. It includes the following steps. (1) Select K partition centers randomly from data sets. (2) Assign each object to its closest center. (3) Recalculate K partition centers and repeat step 2 until the centers convergence. K-means always converges to a local optimum. Moreover, K-means can not filter noise, and does not cluster non-spherical patterns correctly.

CLIQUE integrates grid-based and density-based clustering techniques [5]. CLIQUE initially generates a grid map from feature space. For each dimension, the algorithm identifies the high-density units by using the priori method. Although CLIQUE has a fast clustering time, but its cluster boundaries are either horizontal or vertical, owing to the nature of the rectangular grid.

DBSCAN is a first-density-detecting method, which depends on two arguments, namely Eps and MinPts [4]. Eps denotes the radius of the search circle, and MinPts represents a number of minimal neighbors in the search circle. These arguments are adopted to examine the ε-neighbors contained in each object. DBSCAN can accurately recognize any arbitrary pattern by applying this expansion. Since each expansion must examine all objects, the time complexity of DBSCAN is also high when the database size is large.

GDH is a hybrid grid-based, density-based and hierarchical clustering technique, presented by Wang [6]. GDH refers the idea of density function and gradient decrease and concept of sliding window [6]. It can significantly reduce the limitation of edge indention of traditional grid-based algorithms. However, GDH may fail in the edge of indention if two clusters are the same time in the hypercube.

3 The Proposed ANGEL Clustering Algorithm

This section introduces the new ANGEL clustering concept, algorithm and its implemented steps in the algorithm step by step as follows:

The basic concept of ANGEL clustering can be described in terms of the following four parts.

(1) **Feature space slicing and objects assigning:** This step reduces the number of searching spaces is the main idea. Like GDH, ANGEL inputs the

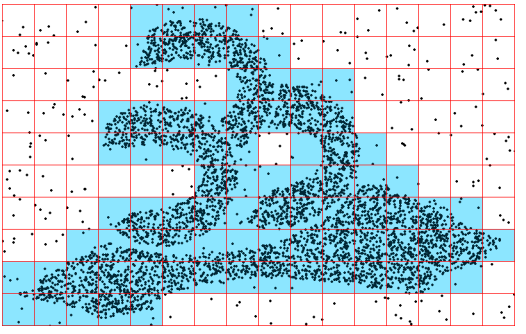


Fig. 1. In the structure of the hypercube map, the hypercubes with dark colors are called populated cube

argument of hypercube's length, and splits the feature space into a hypercube set. Each object of the dataset is assigned to an appropriate hypercube. A hypercube is called populated cube if the number of objects in the hypercube is greater than the threshold **Hd**. Fig. 1 describes this concept.

Influence function [6] is defined as a mathematical description that has the influence of an object has within its neighborhood. The density function [6] is defined as the sum of influence function of all objects in the region, and can be any arbitrary function. For simplicity, this work applies the Euclidean density function and Gaussian representation. The Gaussian density function is given by [6]:

$$f_{Gauss}^{D}(x) = \sum_{i=1}^{N} e^{-\frac{d(x,x_j)^2}{2\sigma^2}}, \tag{1}$$

where $\boldsymbol{N}$ represents the number of objects of the region; $\boldsymbol{d(x, x_i)}$ denotes the distance between $\boldsymbol{x}$ and $\boldsymbol{x_i}$, and $\boldsymbol{\sigma}$ is the standard deviation. A populated cube is called a density-attractor if it has the highest Gaussian density function among all cubes [6]. The density-attractor is the initial point of search space.

(2) **Identifying the main structure:** This investigation employs the discrete-degree as a measure of grid-density detecting preprocesses to identify the main structure of cluster excluding the cluster edge. All populated cubes are split up into nine sub-hypercubes. ANGEL computes the number of objects within each sub-hypercube according to the location of the objects. The range of discrete-degree is derived and defined as follows:

$$UL = (n/9) * (1 + PTV) \tag{2}$$

$$LL = (n/9) * (1 - PTV) \tag{3}$$

$\boldsymbol{UL}$ and $\boldsymbol{LL}$ represent the upper and lower limits of discrete-degree, respectively. $\boldsymbol{n}$ denotes the number of objects of populated cube, and $\boldsymbol{PTV}$ is the percentage of the tolerance value. If all of the density of sub-hypercubes in the hypercube is between $\boldsymbol{UL}$ and $\boldsymbol{LL}$, then the density in the hypercube is equally distributed, and the hypercube is the main structure of the cluster, and can

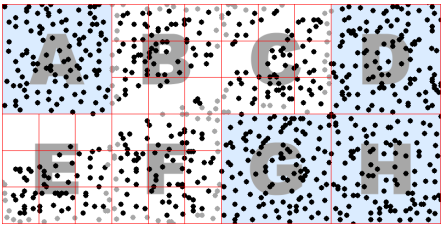


Fig. 2. ANGEL concept for data clustering

be assigned to cluster directly. Otherwise, the edge detection method has to be utilized, as displayed on hypercube B, C, E and F of Fig. 2.

(3) **Edge detection:** The aim of this step is to detect accurately the edge of a cluster. A populated cube that does not belong to the main structure of the cluster may contain objects belonging to two different clusters, as illustrated on hypercube B and C in Fig. 2. Core points and border points of the cluster and noise can be recognized by using DBSCAN to perform detection on hypercubes B, C, E and F on the diagram of Fig. 2. Border points are redefined as objects resulting from a DBSCAN run that are the closest to the hypercube border. This redefinition shortens the computational time in DBSCAN. The light color points (on the border) on hypercube B, C, E and F of Fig. 2 represent border points.

(4) **Merge stage:** After the edge detection stage, the algorithm merges the edge of the cluster with the main structure of the cluster, depending on which border is closest to the main structure. ANGEL repeats the process to recognize all clusters.

The ANGEL clustering algorithm can be described as follows:

```
ANGEL(DataSets,Cl,PTV,Hd,Eps,MinPts)
    Initialization;
    CreatGridStructure(Cl);
    PopulCubes = CacluateGridsInfo(DataSets,PTV,Hd);
    WHILE(TRUE) DO
      C = SelectDensityAttractor(PopulCubes);
      IF C = NULL
        END ALGORITHM
      END IF
      IF isDiscreteDgreeEqual(C) == TRUE
        ChangeClusterId(C,ClusterId);
        SearchNeighbors(C);
      ELSE
        Cs = DBSCAN(C,Eps,MinPts);
        MPC = ChooseMaxsizeSubcluster(Cs);
        ChangeClusterId(MPC,ClusterId);
        SearchNeighbors(C);
      END IF-ELSE
      ClusterId++;
    END WHILE
END ANGEL
```

`DataSets` is an entire database or a partial dataset. `Cl` represents the length of a hypercube; `PTV` denotes the percentage tolerance value, and `Hd` is the threshold

of the hypercube's density. `Eps` represents a search radius, and `MinPts` denotes the smallest number of objects in the region. The algorithm is presented step by step below.

Step 1. Initialization of all arguments.
Step 2. `CreatGridStructure()` function generates the structure of the hypercube map, and assigns all objects to the appropriate hypercube.
Step 3. `CacluateGridsInfo()` function computes the range of discrete-degree for each hypercube, filters it, and returns the populated-cube-set `PopulCubes`.
Step 4. Repeat the process by while loop.
Step 5. `SelectDensityAttractor()` function obtains the density-attractor, and returns to cube `C`.
Step 6. If cube `C` is null, then stop the algorithm.
Step 7. If the discrete-degree of cube `C` is equally distributed, then assign cube `C` directly to the cluster, and continue searching neighbors by the `SearchNeighbors()` function.
Step 8. Otherwise, ANGEL applies `DBSCAN` for the edge detecting and returns a sub-cluster set to `Cs`.
Step 9. Assign a sub-cluster of `Cs` resulting from a `DBSCAN` run to a cluster using the `ChangeClusterId()` function if it is the most populated.
Step 10. ANGEL then searches the neighbors of the cube `C` with the `SearchNeighbors()` function.

The neighbor searching process `SearchNeighbors(Cube)` is as follows:

```
SearchNeighbors(Cube)
    NeighborCubes = SelectNeighbors(Cube);
    WHILE NeighborCubes.length() <> Empty DO
      CurrCube = HighDensity(NeighborCubes);
      IF isDiscreteDgreeEqual(CurrCube) == TRUE
        ChangeClusterId(CurrCube,ClusterId);
        SearchNeighbors(CurrCube);
      ELSE
        NCs = DBSCAN(CurrCube,Eps,MinPts);
        FOR i FROM 1 TO NCs.length()
          IF NCs.SubCluster(i).Borders.areNear(CurrCube) == TRUE
            ChangeClusterId(NCs.SubCluster(i),ClusterId);
          END IF
        END FOR
        SearchNeighbors(CurrCube);
      END IF-ELSE
      NeighborCubes.DeleteFirstNeighbor();
    END WHILE
END SearchNeighbors
```

The neighbor searching step `SearchNeighbors(Cube)` can be illustrated as follows:

Step 1. The `SelectNeighbors()` function returns a set of neighbors `NeighborCubes` located on upside, downside, left side, right side, left up side, left down, right up side and right down side of the cube `Cube`.
Step 2. Continue the process until the neighbors of the cube `Cube` is empty.

Step 3. `HighDensity()` function returns a search subject to cube `CurrCube` with the highest density function.
Step 4. As stated above, if the discrete-degree of cube `CurrCube` is equally distributed, then it is assigned directly to the cluster by `ChangeClusterId()` function, and the neighbors searching continues by the `SearchNeighbors()` function recursively.
Step 5. Otherwise, ANGEL applies `DBSCAN` for edge detection, and returns a sub-cluster set to `NCs`.
Step 6. Each sub-cluster of `NCs` is assigned to a cluster if its border points are close to cube the `CurrCube`.
Step 7. ANGEL then searches the neighbors of the cube `CurrCube` by the `SearchNeighbors()` function recursively.

The process is repeated to merge the entire cluster.

4 Experiment and Analysis

In this work, ANGEL was implemented in a Java-based program, and run on a desktop computer with 256MB RAM, an Intel 1.5GHz CPU on Microsoft Windows 2000 professional Operational System. Seven synthetic datasets were employed in the experiment. Fig. 3 presents the original datasets. The results of the proposed ANGEL algorithm were compared with DBSCAN, K-means, CLIQUE and GDH. Four datasets, with 11,500, 115,000, 230,000 and 575,000 objects in seven synthetic datasets, and all with 15% noise, were utilized in this experiment. The computational time of DBSCAN increases significantly as the number of databases increases. Hence, Table 1 does not list all of the simulation results for DBSCAN (N/A means that the simulations were not performed). Table 1 indicates that the proposed ANGEL with the lowest time cost, and the best clustering correctness rate and noise filtering rate. Due to the limitation of length, not all experimental results are shown. Fig. 4 depicts the experimental results of ANGEL. The experimental results demonstrate that ANGEL can handle arbitrary shapes for clustering. However, K-means cannot recognize arbitrary shapes. Although CLIQUE and GDH could handle the arbitrary shapes in Dataset 4 to 7, CLIQUE could not smoothly detect the edges of clusters, or remove the indentations of neighboring clusters, due to the nature of the rectangular grid. Additionally, the gradient decrease function in GDH placed some clusters in the wrong position if the hypercubes were neighbors but the gradient decrease between the hypercubes was too high. Table 1 shows the clustering experimental results with ANGEL, K-means, DBSCAN, CLIQUE and GDH using 230,000 datasets.

In complex datasets such as DataSets 4, 5, 6 and 7, GDH and CLIQUE require set small capacity of hypercube for segmenting and detecting the edges of the clusters that are close to each other. Hence, the time cost of GDH and CLIQUE raises with increasing numbers of hypercubes to be searched and processed. ANGEL usually yields more accurate results and performs fast than K-means, DBSCAN, CLIQUE and GDH, as shown in Table 1.

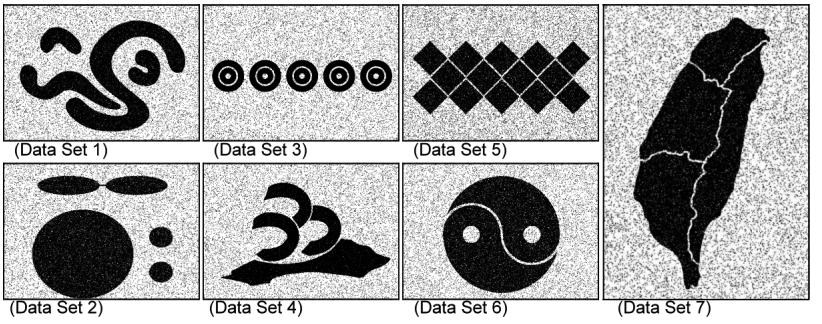


Fig. 3. The original datasets for Experiment

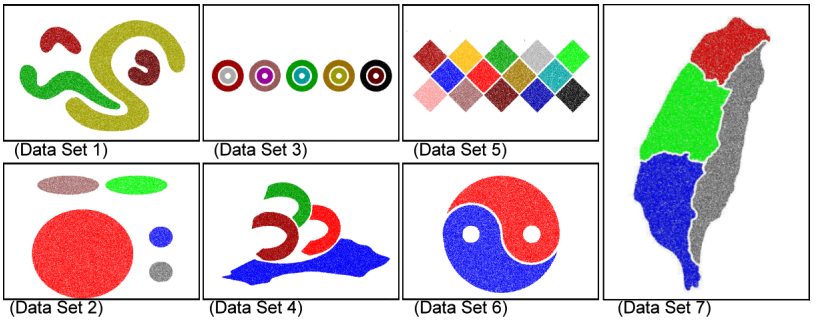


Fig. 4. The experimental results with 230,000 objects of ANGEL

Table 1. Comparisons with ANGEL, K-means, DBSCAN, CLIQUE and GDH using 230,000 data sets with 15% noise; item 1 represents time cost (in seconds); item 2 denotes the clustering correctness rate (%), while item 3 is the noise filtering rate (%)

Algorithm	Item	DataSet-1	DataSet-2	DataSet-3	DataSet-4	DataSet-5	DataSet-6	DataSet-7
K-means	1	8.406	13.782	9.718	20.829	23.891	2.75	7.344
	2	50.032%	56.241%	51.144%	58.108%	84.982%	49.957%	59.056%
	3	0%	0%	0%	0%	0%	0%	0%
DBSCAN	1	11465.96	N/A	N/A	N/A	N/A	N/A	N/A
	2	99.141%	N/A	N/A	N/A	N/A	N/A	N/A
	3	94.6%	N/A	N/A	N/A	N/A	N/A	N/A
CLIQUE	1	2.578	5.203	9.141	25.813	51.984	20.688	39.656
	2	97.934%	99.64%	97.287%	98.208%	96.861%	89.486%	94.157%
	3	96.663%	97.473%	99.013%	99.36%	98.764%	99.686%	99.666%
GDH	1	3.453	5.875	8.985	25.969	50.187	16.672	19.172
	2	99.031%	99.712%	98.009%	98.642%	96.859%	97.791%	94.431%
	3	96.036%	97.406%	98.766%	99.256%	98.764%	99.283%	99.336%
ANGEL	1	3.14	3.782	6.734	6.859	9.281	9.672	11.359
	2	99.05%	99.051%	99.03%	99.271%	98.285%	99.025%	98.412%
	3	96.683%	98.11%	98.656%	99.01%	98.115%	99.08%	99.12%

5 Conclusion

This paper presents a new clustering algorithm called ANGEL that integrates grid-based, density-based and hierarchical approaches. The proposed algorithm makes the following contributions. First, the algorithm improves the clustering performance of large databases. Second, unlike conventional clustering methods, the proposed ANGEL algorithm successfully eliminates edge indention. Finally, the proposed algorithm accurately identifies large patterns that are close to each other. Additionally, simulation results reveal that the proposed new clustering algorithm performs better than some existing well-known approaches such as the K-means, DBSCAN, CLIQUE and GDH methods.

Acknowledgments. The author would like to thank the National Science Council of Republic of China for financially supporting this research under contract no. NSC 95-2221-E-020-036.

References

1. Tsai, C. F., Tsai, C. W., Wu, H. C., Yang, T.: ACODF: A Novel Data Clustering Approach for Data Mining in Large Databases. Journal of Systems and Software, Vol. 73 (2004) 133-145
2. Tsai, C. F., Liu, C. W.: KIDBSCAN: A New Efficient Data Clustering Algorithm for Data Mining in Large Databases. Lecture Notes in Artificial Intelligence, Vol. 4029 (2006) 702-711
3. McQueen, J.B.: Some Methods of Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability (1967) 281-297
4. Ester, M., Kriegel, H. P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (1996) 226-231
5. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Proceedings of the ACM SIGMOD International Conference on Management of Data. Seattle, Washington, D C: ACM Press (1998) 94-105
6. Wang, T. P.: GDH: An Effective and Efficient Approach to Detect Arbitrary Patterns in Clusters with Noises in Very Large Databases. Degree of master at National Pingtung University of Science and Technology, Taiwan (2006)

Exploring Group Moving Pattern for an Energy-Constrained Object Tracking Sensor Network

Hsiao-Ping Tsai[1], De-Nian Yang[1], Wen-Chih Peng[2], and Ming-Syan Chen[1]

[1] National Taiwan University
{hptsai@arbor, tony@kiki, mschen@cc}.ee.ntu.edu.tw
[2] National Chiao Tung University
wcpeng@cs.nctu.edu.tw

Abstract. In this paper, we investigate and utilize the characteristic of the group movement of objects to achieve energy conservation in the inherently resource-constrained wireless object tracking sensor network (OTSN). We propose a novel mining algorithm that consists of a global mining and a local mining to leverage the group moving pattern. We use the VMM model together with Probabilistic Suffix Tree (PST) in learning the moving patterns, as well as Highly Connected Component (HCS) that is a clustering algorithm based on graph connectivity for moving pattern clustering in our mining algorithm. Based on the mined out group relationship and the group moving patterns, a hierarchically prediction-based query algorithm and a group data aggregation algorithm are proposed. Our experiment results show that the energy consumption in terms of the communication cost for our system is better than that of the conventional query/update based OTSN, especially in the case that on-tracking objects have the group moving characteristics.

Keywords: OTSN, Grouping, Data Aggregation, Prediction.

1 Introduction

Energy conservation is paramount among all design issues in the inherently resource-constrained Object Tracking Sensor Network (OTSN). However, the current energy conservation mechanisms and algorithms may not be properly suitable for WSN because WSN differs from other network in many aspects [1] [2] [3]. Energy conservation in OTSN is even a harder affair because the target objects are moving. Many researches aim for designing an energy efficient OTSN. In [4] [5], a cluster-based or tree-based local network collaborates multiple nearby sensors to handle the tracking task. By utilizing the local network, data aggregation and on-off scheduling are applied for reducing communication cost, saving energy, or prolonging the lifetime of the efficient route to the sink. However, while the object moving speed is relatively high, the frequency in rebuilding an energy efficient route and the complexity in maintaining a local network make the route and network structure no longer efficient. Prediction-based schemes

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 825–832, 2007.

predict the future movement of the on-tracking objects according to their latest detected or average velocity to allow an energy efficient wake-up mechanism [6] [7]. Probability-based prediction approaches take advantage of object moving patterns for future location prediction [8] [9] [10] [11]. While investigating the factors that dominate energy cost of OTSN, we observe that many creatures such as animals, birds, or insects have social behavior that is they usually form mass organization and migrate together for food, breeding, wintering or other unknown reasons. The famous annual wildebeest migration is an example. Many water birds such as Black-faced Spoonbill have patterned flight-path. Ultradian rhythmus phenomenon can be found in insect world. Each year, millions of monarch butterflies travel from Canada to Mexico and back again. These examples points out that the movement of creatures is correlated in space and dependent in time. Our point is that if the group relationship can be explored and the next location of a group of objects can be predicted together, the long distance network traffic for query can be reduced and the update data can be aggregated. That's the main idea of the paper and to our best knowledge, this is the first study that considers group relationship and object movement simultaneously in OTSN.

In this paper, we propose a group moving pattern mining algorithm, a hierarchically prediction-based query algorithm, and a group data aggregation algorithm. Our contribution has two folds: first, exploring group relationship among object moving patterns. Second, based on the group moving pattern, an efficient energy-constrained OTSN is designed. In our algorithm, a group probabilistic suffix tree (GPST) is used to predict objects' next locations in hierarchically prediction-based query algorithm. The group data aggregation algorithm utilizes group information in local and small scale data aggregation. The rest of the paper is organized as follows. Preliminaries, definitions and problem formulation are presented in Section 2. Algorithms for mining group moving patterns, a hierarchically prediction-based query algorithm, and an efficient group data aggregation algorithm are proposed in Section 3. Experiments are presented in Section 4. This paper concludes with Section 5.

2 Preliminaries, Definitions, and Problem Formulation

2.1 Preliminaries

Hierarchical Sensor Network. In a hierarchical WSN, nodes are heterogeneous in energy, computing and storage capacity. Higher-energy node can be used to perform high complexity computing and send data while low-energy node can be used to perform the sensing and low complexity computing. A sensor equipped with higher energy acts as a cluster head (CH) on which high complexity task are assigned. In this paper, we adopt the hierarchical cluster structure. Sensors within a cluster have a locally unique id, and CH logical represents sensors within the cluster and acts as a sensor in view from upper layer. When a sensor detects an on-tracking object, it informs the location information to the CH. The CH aggregates location information then forwards to CH of upper layer. The process repeats until sink receives the location information. Here,

location information of an object is the corresponding sensor id. The movement of an object is represented by a sequence of sensor id visited by the object.

Location Modeling. The movement dependency is that the next location in a moving sequence can be predicted from the sequence of preceding locations. We use Variable Markov Model (VMM) for learning statistic of object moving sequences and also a data structure called Probabilistic Suffix Tree (PST) is used together for mining significant moving patterns [12]. PST building algorithm is an efficient lossy compression scheme that converts a large data set into a variable length tree. The tree represents a dictionary of significant sequences that are meaningful for predicting next location.

2.2 Definitions and Problem Formulation

Definition 1. *Group Data Aggregation Radius (GDAR) is the number of hop counts between a sensor and its furthest neighbor in the participation of the group data aggregation.*

Definition 2. *Moving Sequence is a sequence of sensor id visited by one or a group of objects. Moving Speed is the number of sensor that an object crosses in a time unit.*

Given a moving sequence data set, our problem is to find the group relationship and the group moving pattern and obtain an energy efficient hierarchical OTSN.

3 Designs and Algorithms for OTSN

While a sensor detects an object, it invokes a group data aggregation and then informs a list of detected objects and a sensor id to the CH. CHs therefore collect objects' moving sequences within its cluster. In this section, we first present the group moving pattern mining algorithm and then propose an efficient object tracking sensor network.

3.1 Group Moving Pattern Mining Algorithm

The group moving pattern mining algorithm has four steps: building PST for each object, constructing a similarity graph on PSTs, extracting highly-connected components, and selecting Group Probability Suffix Tree (GPST). After the mining is performed, CH sends the group information to upper layer and gets a group id in return. The group information, group id and GPST are used in the hierarchically prediction-based query and group data aggregation.

Building PSTs for All Objects. The movement data set in the CH is a set of moving sequences collected within this cluster. In the step, the CH builds a PST for each object. The Build-PST algorithm is shown in Figure 1.

Build - PST ($P_{\min}, \alpha, \gamma_{\min}, r, L$)

1. Initialization :
   Let T consist of a single root node (with an empty label) and let $S \leftarrow \{\sigma | \sigma \in \sum \text{ and } P(\sigma) \geq P_{\min}\}$
2. Building the PST skeleton :
   While $S \neq \phi$, pick any $s \in \sum$ and do :
   (a) Remove s from S .
   (b) If there exists a symbol σ such that $P(\sigma|s) \geq (1+\alpha) \times \gamma_{\min}$
   and $\left\{ \frac{P(\sigma|s)}{P(\sigma|suf(s))} \geq r \text{ or } \frac{P(\sigma|s)}{P(\sigma|suf(s))} \leq \frac{1}{r} \right\}$, then add to T the node for s and all the nodes on the path to s from the deepest node in T that is a suffix of s .
   (c) If $|s| < L$ then add the strings $\{\sigma' s | \sigma' \in \sum \text{ and } P(\sigma' s) \geq P_{\min}\}$ to S, if any.
3. Smoothing the prediction probabilities :
   For each node s in T, let $\gamma_s(\sigma) = (1 - |\sum| \times \gamma_{\min}) \times P(\sigma|s) + \gamma_{\min}$

Fig. 1. The Build_PST Algorithm [12]

Constructing a Similarity Graph on PSTs. Given n PSTs, we want to group them according to their structural similarities. The similarity score sim_p of two PSTs is defined as follows,

$$sim_p = -\log\Big(\sum_{s \in S} \sqrt{\frac{\sum_{\sigma \in \Sigma} (P_1(s) \times P_1(\sigma|s) - P_2(s) \times P_2(\sigma|s))^2}{|\Sigma|}} \Big) \tag{1}$$

It is a combination of condition probability and Euclidean distance over the union of node symbol strings of the two PST. The union is identified by S. If sim_p of two PSTs is higher then a given threshold, we consider they are similar. After comparing each pair of the given PSTs, we have pair relationship among all objects that forms a graph. Then the problem is transformed to be a graph connectivity problem.

Extracting highly-Connected components. Connectivity $k(G)$ of a graph G is defined as the minimum number of edges whose removal results in a disconnected graph. Given a graph with n nodes, we partition the graph into subgraphs such that the subgraphs are highly connected. We partition the graph G if $k(G) < \frac{n}{2}$. The highly connected component is an induced subgraph $G' \subseteq G$ such that G' is highly connected [13]. The nodes in a highly connected subgraph has degree at least $\frac{n'}{2}$ which means that each node in the subgraph has similarity relation with at least half nodes in the subgraph. The HCS cluster algorithm is shown in Figure 2. The clustering problem is then converted to a min-cut problem. A simple min-cut algorithm [14] is utilized in partition of the nodes.

Selecting Group Probabilistic Suffix Tree. In last step, the similarity graph is partitioned into highly connected subgraphs. If the size of the subgraphs is higher than a given threshold, the objects corresponding to the subgraph are considered as an efficient group. The threshold is selected such that the efficiency gained on group based query algorithm and group data aggregation deserves the mining. After groups are found, a most representative PST named GPST is heuristically selected for each group such that the storage cost is reduced.

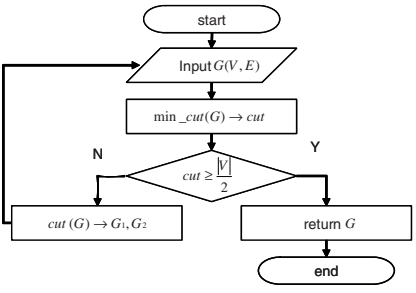


Fig. 2. The HCS Algorithm

3.2 The Energy Efficient Object Tracking Sensor Network

The energy efficient OTSN includes a hierarchically prediction-based query algorithm and a group data aggregation algorithm. The prediction-based query algorithm flexibly determines to query for an individual object or a group of objects by one query. If on-tracking objects belong to a group, a group query can be applied such that not only flooding-based network traffic is eliminated but long-distance network traffic is reduced. In addition, the total data amount for the updates is further compressed under acceptable location precision by the group data aggregation algorithm. In the best situation, only one query is required in querying for a group of objects, and only one update is required in each updating interval for a group of objects.

Hierarchically Prediction-Based Query Algorithm. After the groups and their best fit GPSTs are produced. The information is sent to the CHs such that a group query and group update can be achieved efficiently. For the query-based OTSN, while receiving a query, the sink first predicts the most possible cluster that the object is currently located by using GPST and then sends the query to the CH. While the CH receives the query, it performs another prediction to get the most possible sensor that can detect the object. After receiving the query, the sensor invites its neighbors within GDAR to participate in tracking the object.

Group Data Aggregation Algorithm. For the update-based OTSN, while an object is detected by a sensor, a group data aggregation process is initiated. The sensor performs as the master sensor that invites its neighbors within GDAR to collaborate in tracking objects and handles the local data collection for a period. After that, the master sensor reports to the CH about the detected objects and the id of the sensor that detects most objects. Finally, the CH further compresses total data amount by using group id and filters redundant data according to the specified precision.

4 Experiments

We implemented an event driven simulator in C++ with SIM [15] to evaluate the performance of our design. In the simulation, we use a Location-Dependent

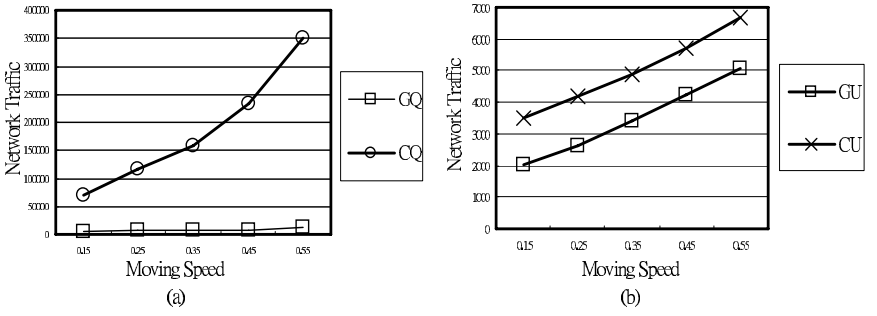


Fig. 3. Network traffic comparison

Parameterization of a Random Direction Mobility Model [16] to simulate the roaming behavior of a group leader. The other members are followers that are uniformly distributed around the leader within a specified Group Dispersion Range (GDR). The metrics used in evaluating our design are the amount of transmitted data in bytes per sensor (network traffic) and the average distance between observed location and real location of all objects (location distance).

4.1 Experiment 1

In the first experiment, we compare the performance between conventional query-based OTSN (CQ) and our group prediction query-based OTSN (GQ) as well as conventional update-based OTSN and our group aggregation update-based OTSN (GU). The sink has to persistently query for an object in query-base OTSN, and the sensors surrounding target objects persistently update the locations of objects in update-base OTSN. Figure 3(a) shows that the network traffic of CQ is about 20 times of GQ at the same location distance (˜1.0). While the moving speed is higher, the rate is higher. Figure 3(b) shows that the network cost in CU about 1.48 time of GU at the same location distance (˜1.0). While object moving speed is low, CU can achieve high location precision at a comparable network cost. However, if the object moving speed is higher, in order to achieve the same location precision, the update frequency increases and incurs much more traffic than GU.

4.2 Experiment 2

In the experiment, we study the impact of object moving speed and query interval on the location precision in GQ. The results in Figure 4(a) show that higher query frequency leads to a better location precision. Besides, at the same query interval, if the object moving speed is higher, the location precision is lower. To achieve higher location precision, higher query frequency is required.

4.3 Experiment 3

Figure 4(b) shows the impact of GDR. Higher GDR means that objects are more scattered such that a larger GDAR is required. GDAR is an important parameter

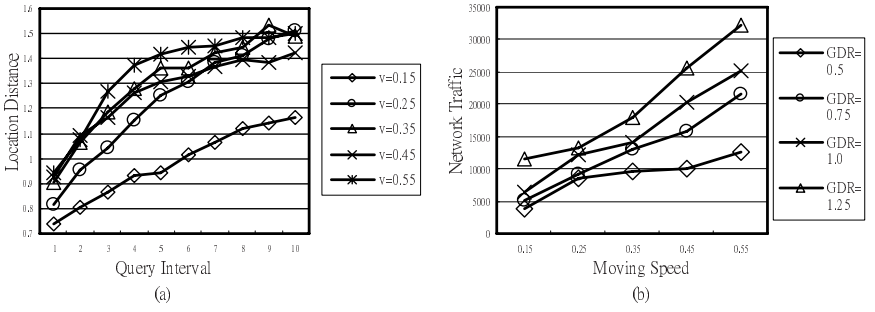


Fig. 4. (a)The impact of object moving speed on the location precision. (b)The impact of GDR on the network traffic.

that influences both the location information collecting range and object query snooping range. If GDR is higher, GDAR is set to be larger, and therefore more local in-network traffic is incurred.

5 Conclusions

In the paper, a group moving pattern mining algorithm, a prediction-based routing algorithm, and a group data aggregation algorithm are proposed. A simulation is conducted in light of the performance of the algorithms. Our contribution has two folds: first, exploring group relationship among object moving patterns. Second, based on the group moving pattern, an efficient prediction-based query algorithm and an efficient data aggregation algorithm for OTSN are provided. According to our experiments, the explored group relationship and group moving pattern are adopted to predict the group location such that the amount of query network traffic is significant reduced. The amount of update network traffic that is incurred by reporting the location of monitored objects is also greatly reduced especially while the group density of monitored objects are high. In addition, the adaptive data aggregation range improves the prediction hit rate.

Acknowledgements

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE.

References

1. Akyildiz, I.F., W. Su, Y.S., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks **38**(4) (2002) 393–422
2. Culler, D., Estrin, D., Srivastava, M.: Overview of sensor networks. IEEE Computer, Special Issue in Sensor Networks (2004)
3. Al-Karaki, J.N., Kamal, A.E.: Routing techniques in wireless sensor networks: a survey. IEEE Wireless Commun. **11**(6) (2004) 6–28

4. Zhang, W., Cao, G.: Dctc: Dynamic convoy tree-based collaboration for target tracking in sensor networks. IEEE Trans. on Wireless Commun. **3**(5) (2004)
5. Lin, C.Y., Peng, W.C., Tseng, Y.C.: Efficient in-network moving object tracking in wireless sensor networks. IEEE Trans. on Mobile Comput. **5**(8) (2006) 1044–1056
6. Xu, Y., Winter, J., Lee, W.C.: Prediction-based strategies for energy saving in object tracking sensor networks. IEEE Int. Conf. on Mobile Data Manag. (2004) 346–357
7. Yang, L., Feng, C., Rozenblit, J.W., Qiao, H.: Adaptive tracking in distributed wireless sensor networks. 13th Annual IEEE Int. Symp. and Workshop on Engineering of Computer Based Systems (2006) 103–111
8. Tseng, V.S., Lin, K.W.: Mining temporal moving patterns in object tracking sensor networks. Int. Workshop on Ubiquitous Data Manag. (2005)
9. Tseng, Y.C., Kuo, S.P., Lee, H.W., Huang, C.F.: Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. Int. Workshop on Information Processing in Sensor Networks (2003)
10. Peng, W.C., Ko, Y.Z., Lee, W.C.: On mining moving patterns for object tracking sensor networks. 7th Int. Conf. on Mobile Data Manag. (2006)
11. Ma, S., Tang, S., Yang, D., Wang, T., Han, J.: Combining clustering with moving sequential pattern mining: A novel and efficient technique. 8th PAKDD (2004) 419–423
12. Ron, D., Singer, Y., Tishby, N.: Learning probabilistic automata with variable memory length. 7th annual Conf. on Computational learning theory (1994)
13. Hartuv, E., Shamir, R.: A clustering algorithm based on graph connectivity. Information Processing Letters **76**(4-6) (2000) 175–181
14. Stoer, M., Wagner, F.: A simple min-cut algorithm. Journal of the ACM **44**(4) (1997) 585–591
15. Bolier, D.: Sim : a c++ library for discrete event simulation (1995)
16. Gloss, B., Scharf, M., Neubauer, D.: Location-dependent parameterization of a random direction mobility model. IEEE 63rd Conf. on Veh. Technol. **3** (2006) 1068–1072

ProMail: Using Progressive Email Social Network for Spam Detection

Chi-Yao Tseng, Jen-Wei Huang, and Ming-Syan Chen

Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan, ROC
{cytseng, jwhuang}@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw

Abstract. The spam problem continues growing drastically. Owing to the ever-changing tricks of spammers, the filtering technique with continual update is imperative nowadays. In this paper, a server-oriented spam detection system **ProMail**, which investigates human email social network, is presented. According to recent email interaction and reputation of users, arriving emails can be classified as spam or non-spam(ham). To capture the dynamic email communication, the progressive update scheme is introduced to include latest arriving emails by the feedback mechanism and delete obsolete ones. This not only effectively limits the memory space, but also keeps the most up-to-date information. For better efficiency, it is not required to sort the scores of each email user and acquire the exact ones. Instead, the reputation procedure, **SpGrade**, is proposed to accelerate the progressive rating process. In addition, **ProMail** is able to deal with huge amounts of emails without delaying the delivery time and possesses higher attack resilience against spammers. The real dataset of 1,500,000 emails is used to evaluate the performance of **ProMail**, and the experimental results show that **ProMail** is more accurate and efficient.

1 Introduction

Spam, the so-called junk email, has become an imperative problem to the email communication today. In fact, there are a huge percentage of all emails sent are spams. The statistics [1] shows that more than 50% of emails are spams in 2004, up from 8% in 2001. Even worse, spam not only annoys email users, but also threatens the whole email application. Some systems have been presented to prevent the disturbance of spams. However, each of them has drawbacks and no one settles the problem completely.

For data mining researchers, spam detection is brand-new and challenging. At the initial stage, they commonly treated the spam detection as a text classification task. In this way, the contents of emails are exploited for spam detection. Some conventional machine learning techniques have been conducted. Among them, Naive Bayes methods [2] and SVMs [3] were the most popular. Though they have certainly reported quite excellent results, it is obvious that the spam problem does not nearly go away. These filters generally work only when the scale

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 833–840, 2007.

is miniature. When they are employed on a large scale, spammers always find new ways to attack the filters. Moreover, the contents of spam change over time. Only filters with constant re-training can attain high performance permanently.

The other group of researchers try to look for other clues, such as the email traffic [4], the email social network [5], and so forth [6]. These approaches are categorized as "non-content-based spam filters". Compared with content-based approaches, these methods possess several advantages. First, less processing time is required. Instead of going through the entire email contents, it is much more efficient in dealing with non-content knowledge. Second, to create the counterfeit non-content information is relatively difficult. Finally, the privacy issue of the email contents has always been concerned. It is noted that non-content-based approaches evade this issue inherently.

Due to above merits, we decide to use the non-content knowledge of email social network formed by human email communication for spam detection. MailRank [5], presented by Chirita *et al*, investigated the feasibility of rating the sender addresses in the email social network. The well-known reputation algorithm, PageRank [7], is the kernel of the rating mechanism of MailRank. According to the score obtained by the algorithm, an arriving email can be classified as a spam or a ham. Although the authors claimed high performance of MailRank, there are still some drawbacks that can be improved. MailRank constructs a graph for email social network. Each node represents an email address and each arrow points from a sender to a receiver. In this way, according to the concept of PageRank algorithm, a sender gives a trust vote to a receiver. However, the key idea of PageRank is that the rank of a web page is given by the pages pointing to this page. Therefore, if an arrow points from a sender to a receiver, it means the receiver gets the trust vote from the sender. This scoring scheme is a contradiction to the real world. The trust score of an email address should be determined by the number of trusted emails it has sent instead of the number of emails received. In addition, only a single link is allowed between two nodes in MailRank. This diminishes the effect of different numbers of email communications between two nodes. In the real-world situation, the more spams a node has sent, the more suspicious to be a spammer it is. Moreover, MailRank collects merely the ham emails as training data. All links in the graph are legitimate emails. This may lack the information of communications of junk emails.

In this paper, we propose a spam detection system **ProMail**, which utilizes the non-content knowledge of the email social network. According to the latest email interaction and reputation of users, an arriving email can be classified as a spam or a ham. Three primary contributions of this paper are as follows.

(1) We design an innovative modeling graph of the email social network, whose concept is illustrated in Figure 2 and is elaborated in Section 3.1.

(2) The progressive update scheme is presented to catch the evolving feature of the real-world communication. Newly training emails are included and the obsolete emails are pruned away. This crucial scheme makes **ProMail** more suitable for modeling the latest email social network and thus with higher attack resilience.

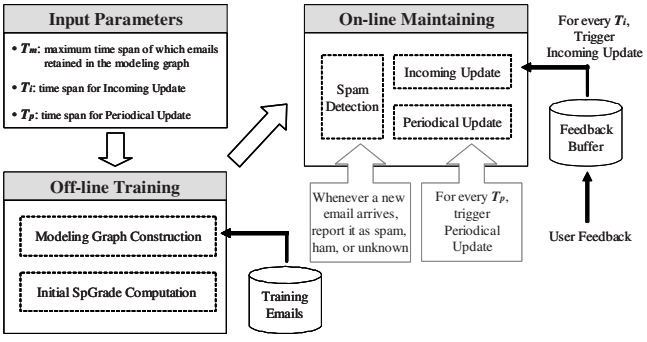


Fig. 1. System model of ProMail

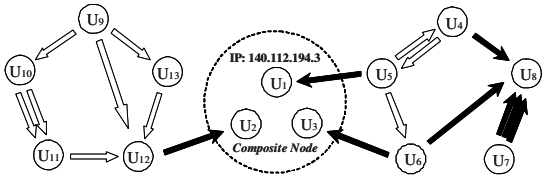


Fig. 2. The modeling graph of email social network

(3) The reputation procedure, **SpGrade**, is proposed to rate each email address in the modeling graph. **SpGrade**, which follows the concept of PageRank [7], accelerates the convergence rate without re-running the reputation algorithm and hence **ProMail** efficiently keeps the most up-to-date information for spam detection. Consequently, **ProMail** is capable of handling a great quantity of arriving emails in real time.

2 System Model of ProMail

Figure 1 demonstrates the system model of **ProMail**. Three input parameters, T_m (the maximum time span of which emails retained in the modeling graph), T_i (the time span for Incoming Update), and T_p (the time span for Periodical Update) are given. There are two major phases in the system, namely, off-line training phase and on-line maintaining phase. In the off-line training phase, **ProMail** includes initial training emails, which are collected from users for every T_m, to construct our new modeling graph. Then, the proposed reputation procedure **SpGrade** is conducted to compute the initial objective score for each node in the modeling graph. At this stage, **ProMail** is able to be put on-line to do the spam detection based on the initial objective scores.

Three individual modules, Spam Detection, Incoming Update, and Periodical Update are involved in the on-line maintaining phase. Whenever a new email arrives, Spam Detection instantly reports the result based on the objective scores obtained by **SpGrade**. It is noted that Spam Detection can be computed in real

time without delaying the delivery time of arriving emails. In addition, to keep the latest information, the feedback mechanism is introduced, and the reported feedback emails are buffered temporarily. For every T_i, the system automatically triggers the Incoming Update and includes buffered emails into the modeling graph. In the meantime, **SpGrade** progressively adjusts the objective scores of email users related to or influenced by the feedback emails. Note that **SpGrade** considers only associated nodes rather than re-rating all nodes in the modeling graph. Meanwhile, the decaying effect, which conveys the concept that the significance of emails decays over time, is also taken into account in **SpGrade**. Periodical Update is triggered for every T_p. Those nodes which were not modified in the previous time span T_p are re-rated. For efficiency concern, T_p is normally set larger than T_i by several times. Moreover, the obsolete emails which exceed the maximum time span T_m are eliminated in Periodical Update. Consequently, **ProMail** is self-adjusting and always retains the most up-to-date knowledge of human email communication for spam detection with limited memory space.

3 Using Progressive Email Social Network for Spam Detection

3.1 Modeling Graph of the Email Social Network

Figure 2 shows our new modeling graph of the email social network in **ProMail**. Each node denotes an email user represented by the email address. Each arrow indicates an email communication between two nodes. There are four significant features of the innovative modeling graph.

(1) The direction of an arrow points from a receiver to a sender. This demonstrates the situation that the receiver gives the trust vote to the sender.

(2) There can be more than one link connected between two nodes. Namely, each link is weighted in accordance with the number of emails. For the ease of readability, we put all links in the modeling graph to emphasize the weighting effect. In addition, the decaying effect, which emphasizes the decays of importance over time, is also contained on the links.

(3) Both spam and ham links are included. Solid arrows stand for spam links, while hollow arrows represent ham links. Two types of email links are distinguished with each other. Thus, two respective email social networks are fused together in the modeling graph. Each node maintains two corresponding scores for the spam network and the ham network.

(4) The composite node, which consists of senders who appeared just once and had identical IP address, is introduced. This is because the email header is easily tampered by spammers. However, the IP address of the sending machine is valid. Large amounts of spams sent from one machine may have different fake sender addresses, but their sender IP addresses are the same. As a result, these nodes should be enclosed to form a composite node. Comparatively, other nodes are named as common nodes. Though normal users may be mis-enclosed together as well, they will jump out and form a new common node by itself at the

moment they appear the second time. The dashed circle in Figure 2 illustrates a composite node.

3.2 Algorithm ProMail

Algorithmic form of **ProMail** is shown in Figure 3(a). The major objectives of **ProMail** are to maintain the progressive email social network as time goes by and to report the classification results whenever new emails arrive. There are two primary phases in **ProMail**. One is off-line training phase, and the other is on-line maintaining phase. Initially, a large set of training emails are fed into **ProMail** for the off-line phase. In this phase, from line 4 to line 6, three procedures are included. **ConstructGraph** utilizes the training email set to construct the modeling graph of the email social network. The training set contains the emails arriving in previous T_m. The reputation procedure **SpGrade** is then applied to rate each node in the graph. Two corresponding scores, ham and spam, are produced for each node. By the linear combination of two scores, each node obtains the objective value which indicates the extent of suspicion to be a spammer. The last step of the off-line phase is to determine the separation threshold S_{th} to classify emails into spams and hams. First, **ProMail** samples the nodes in the graph uniformly, and sorts the sample set of nodes by their objective values. Then, the separation threshold is determined according to the proportion of spams and hams in the training data. For example, if there are 1,000 sampled nodes and half of the training data are spams, S_{th} will be set the same as the value of the node whose rank of the objective value is 500. At this stage, **ProMail** finishes the off-line training phase and goes into the on-line maintaining phase from line 8 to line 16. Two types of update schemes, Incoming Update and Periodical Update, are involved. Incoming Update, from line 8 to line 10, includes new feedback emails from users by **ConstructGraph**. Feedback emails are buffered temporarily and Incoming Update is triggered for every T_i. Since some nodes and links are inserted into the graph, **SpGrade** is conducted in line 10 to re-rate the scores of the related nodes. To achieve the efficient update, **SpGrade** considers merely the nodes associated with the feedback nodes. On the other hand, Periodical Update, from line 11 to line 14, is triggered for every T_p. The main objectives of Periodical Update are to eliminate the obsolete links and to update the nodes which have not been updated in previous T_p. In line 13, **DeleteGraph** excludes the obsolete links which exceed the time span T_m from the graph. While **DeleteGraph** examines the obsolete links, the nodes which have not been updated in previous T_p are also marked so that these nodes can be updated in the following procedure. Finally, **SpGrade** is employed to re-rate corresponding nodes. Owing to the operation of Incoming Update and Periodical Update, **ProMail** is able to keep the most up-to-date information and delete the obsolete knowledge for spam detection with limited memory space. In addition, for efficiency concern, T_p is set larger than T_i by several times. In the following subsections, the reputation mechanism **SpGrade** will be elaborated in detail.

```
Algorithm ProMail
Input:  Tm: maximum time span which emails
            retained in the modeling graph
        Ti: time span of incoming update
        Tp: time span of periodical update
1   var Graph; // Modeling Graph
2   var currentTime;
3   var Sth; // the score separation of spam and ham;
3   // Off-line Training Phase
4   ConstructGraph( );
5   SpGrade( );
6   determine Sth;
7   // On-line Maintaining Phase
8   for (every time span Ti) // Incoming Update
9       ConstructGraph( );
10      SpGrade( );
11  for (every time span Tp) // Periodical Update
12      update currentTime;
13      DeleteGraph(Tm, Tp, currentTime);
14      SpGrade( );
15  while (new emails arrive) // Email Classification
16      classify the email as spam, ham, or unknown by Sth;
End
```

(a) Algorithm ProMail

```
Procedure SpGrade ( )
// Capital 'S' denotes the word 'score'.
1   var d_factor; // exponential decaying factor
2   var err_th; // error threshold;
3   while (there exists some marked nodes in Graph)
4       for(each marked node in Graph)
5           tmp_nodeS = node.S;
6           node.S = 0;
7           for (each inlink of the node)
8               tmp_time = currentTime - link.time;
9               node.S= node.S + inlink.S * exp(d_factor, tmp_time);
10              link.decayS = link.S * exp(d_factor, tmp_time);
11          if ((node.S - tmp_nodeS) / tmp_nodeS > err_th)
12              tmp_linkS = node.S / node.no_outlink;
13              for (each outlink of the node)
14                  tmp_time = currentTime - link.time;
15                  tmp_linkS = tmp_linkS * exp(d_factor, tmp_time);
16                  link.S = tmp_linkS;
17                  if ((tmp_linkS - link.decayS) / link.decayS > err_th)
18                      mark this link's terminal node for the next iteration;
19                  link.decayS = tmp_linkS;
End
```

(b) Procedure SpGrade

Fig. 3. Algorithm ProMail and procedure SpGrade

3.3 Details of Procedure SpGrade

SpGrade, the reputation mechanism of **ProMail**, is presented in Figure 3(b). **SpGrade** takes charge of rating and updating the scores of nodes in the modeling graph. To catch the concept of the importance of links decays over time (called the decaying effect), *d_factor*, the exponential decaying factor, is introduced to diminish the scores of links according to the existing time. In addition, to accelerate the convergence rate, **SpGrade** adopts two crucial measures. First, only the marked nodes are considered. In the first iteration, two types of marked nodes are involved. One is the nodes which are associated with the insertion and the deletion of the graph. The other is the nodes which have not been updated in T_p. During each iteration, **SpGrade** marks the nodes which are required to be updated for the next iteration. The second crucial measure is that the constraint of update propagation is included. The primary purpose of the constraint is to avoid triggering too many slight updates caused by a single marked node. The node triggers the update propagation only when the change percentage of scores in a node is greater than the error threshold *err_th*. In addition, the constraint of update propagation is enforced on the link score as well. From line 7 to line 10, the targeted marked node re-computes the score by summing all inlink scores with the decaying effect. Meanwhile, each inlink *decayS*, the decayed score which has been acquired by the terminal node, is updated. Providing that the score change percentage of this node exceeds *err_th*, the update propagation, from line 12 to line 19, is executed. The node averagely shares the score to all of the outlinks. From line 14 to line 16, **SpGrade** first updates the outlink score by the decaying effect. Then, if the score change percentage of this outlink also exceeds *err_th*, the terminal node of this link will be marked for the update in the next iteration. This measure effectively accelerates the convergence rate of

SpGrade, and does not influence the detection results. Finally, in line 19, the outlink *decayS* is updated. It is noted that the entire nodes in the graph should be processed concurrently. The newly updated scores become effective in the next iteration but not in the current iteration. Both ham and spam scores are processed in the same way.

4 Experimental Evaluation

In this section, we conduct experiments to validate the feasibility of **ProMail**. The real dataset used in the experiments is the email server log of Computer Center in National Taiwan University. There are about 200,000 emails per day in this dataset. Since the classified labels of emails are not available in practice, we utilize the detection results from the well-known existing system, SpamAssassin, as the correct labels. There are 30% of emails are spams in this dataset. The competitive approach, MailRank, is included for comparison. We implement **ProMail** and MailRank using C++ language, and execute the programs on a computer with Pentium 4 - 3GHz CPU and 1.5GB RAM.

	False Positive	False Negative	Uncertain	Unknown
2% / 4%	**0.45%**	**3.52%**	**4.26%**	**16.62%**
4% / 8%	**0.37%**	**3.47%**	**8.62%**	**16.62%**
6% / 12%	**0.19%**	**3.41%**	**12.26%**	**16.62%**
8% / 16%	**0.11%**	**3.25%**	**15.21%**	**16.62%**
10% / 20%	**0.08%**	**3.20%**	**18.93%**	**16.62%**

(a) Performance of ProMail

	False Positive	False Negative	Uncertain	Unknown
ProMail	**0.47%**	**4.07%**	**7.45%**	**15.93%**
MailRank	**8.25%**	**6.57%**	**8.26%**	**32.38%**

(b) Comparison of ProMail and MailRank

Fig. 4. Accuracy Evaluation

4.1 Accuracy Evaluation

The training dataset used in Figure 4(a) consists of emails in one week (T_m is one week), and there are totally about 1,400,000 emails. T_i and T_p are both set as one day. Another 7 days of emails are used as testing data. The procedures of the experiment are as follows. Each day of testing emails represents the arriving emails and are tested day by day. Then, all emails are viewed as the feedback emails for the system. In addition, two values in the first column of Figure 4(a) stand for the interval between the definitely ham threshold and the interval between the definitely spam threshold with S_{th}. The sender address whose score is between these two thresholds is classified as uncertain. The sender address which does not appear in the graph is classified as unknown. The false positive rate and the false negative rate are investigated. As mentioned in Section 1, the false positive costs much than the false negative for spam detection problem. As shown in Figure 4(a), the false positive rate of **ProMail** is lower than 0.5% in all cases. This validates the high performance of **ProMail**.

Figure 4(b) shows the comparison results against MailRank. The unknown rate of MailRank is much higher than **ProMail** since only hams are considered in MailRank. Moreover, **ProMail** outperforms MailRank in false positive

rate considerably. The main reason is that some email senders who have sent fewer emails possess low score, and are mis-classified as spammers by MailRank. However, **ProMail** can handle this drawback well.

5 Conclusion

In this paper, we proposed a spam detection system **ProMail**, which utilizes the non-content knowledge of the email social network. The innovative modeling graph was presented to model the human email communication. In addition, the progressive update scheme was introduced to include new feedback emails and delete the obsolete ones, and hence the most up-to-date information is always kept for spam detection. Moreover, we designed the reputation mechanism **SpGrade** to accelerate the convergence rate without re-running the reputation algorithm. The experimental results showed that **ProMail** outperforms the competitive algorithm MailRank significantly. Furthermore, the outstanding false positive rate justifies the practicability of **ProMail** in real applications.

Acknowledgements

The work was supported in part by the National Science Council of Taiwan, R.O.C., under Contracts NSC93-2752-E-002-006-PAE.

References

1. Hulten, G., Goodman, J.: Junk mail filtering. Tutorial in SIGKDD 2004 (2004)
2. Schneider, K.M.: A comparison of event models for naive bayes anti-spam e-mail filtering. In Proc. of the 10th Conference of the European Chapter of the Association for Computational Linguistics (2003)
3. Kolcz, A., Alspector, J.: Svm-based filtering of email spam with content-specific misclassification costs. In Proc. of the ICDM Workshop on Text Mining (2001)
4. Desikan, P., Srivastava, J.: Analyzing network traffic to detect e-mail spamming machines. In Proc. of the Workshop on Privacy and Security Aspects of Data Mining (2004)
5. Chirita, P.A., Diederich, J., Nejdl, W.: Mailrank: Using ranking for spam detection. In Proc. of the 14th ACM International Conference on Information and Knowledge Management (2005)
6. Sankaralingam, K., Sethumadhavan, S., Browne, J.C.: Distributed pagerank for p2p systems. In Proc. of the 12th IEEE International Symposium on High Performance Distributed Computing (2003)
7. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. In Technical Report Computer Systems Laboratory Stanford University Stanford CA (1998)

Multidimensional Decision Support Indicator (mDSI) for Time Series Stock Trend Prediction

Kuralmani Vellaisamy and Jinyan Li

Institute for Infocomm Research,
21 Heng Mui Keng Terrace, Singapore 119613
{vkmani,jinyan}@i2r.a-star.edu.sg

Abstract. This work proposes a generalized approach for predicting trends in time series data with a particular interest in stocks. In this approach, we suggest a multidimensional decision support indicator $mDSI$ derived from a sequential data mining process to monitor trends in stocks. Available indicators in the literature often fail to agree with their predictions to their competitors because of the specific nature of features each one uses in their predictions like moving averages use means, momentums use dispersions, etc. Then again, choosing a best indicator is a challenging and also expensive one. Thus, in this paper, we introduce a compact, but robust indicator to learn the trends effectively for any given time series data. That is, it introduces a simple multdimensional indicator such as $mDSI$ which integrates multiple decision criteria into a single index value that to eliminate conflicts and improve the overall efficiency. Experiments with $mDSI$ on the real data further confirm its efficiency and good performance.

Keywords: time series datamining, stock trends, multidimensional decision support indicator, technical indicator.

1 Introduction

Massive amount of financial time series data encourages investors to find new technical devices to understand a stock's behavior. Methods such as support vector machines and neural networks provide excellent support for time series data mining, particularly forecasting and predicting trends in stocks, [1,2,3], etc. On the other hand, technical analysis promotes a wide variety of indicators to predict the start of the trends (or derivatives) of prices or volumes over time. For example, an indicator, the relative strength index (RSI) tracks the trends and signals by weighing the difference between gains and losses for n days; another indicator the momentum tags similar trends with tracking price changes over n days; the moving averages (MA) use mean to see these trends; Stochastics (ST), covers same trends by adjusting its momentums; the price rate of changes (ROC) understands these with measuring the price changes; etc. However, basically

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 841–848, 2007.

these indicators use time series data points to detect the trends or shifts in supply and demand, particularly identifying regions of interests such as overbought and oversold and market conditions, namely bullish and bearish [4,5,6,7,8,9,10] are of the interests. Overbought (or oversold) is a condition in which a stock or market has recently experienced a sharp rise (or fall) in price and is vulnerable to a price drop (or rise), because few buyers (or sellers) are left to drive the price up (or down) any farther. Similarly, bullish (or bearish) signal is used to describe an optimistic (or pessimistic) sentiment toward an issue, an index, or the market in general; a bullish (or bearish) sentiment reflects a belief that prices will tend to rise (or fall).

Though these indicators seem to predict the trends with more accuracies, often they disagree in their predictions. This is due to the fact that they use different features like moving averages use means, momentums use dispersions, etc in their predictions. Choosing a best indicator or indicators is expensive and challenging. Rather, in this work, we propose a generalized approach for predicting trends in a time series data in which a univariate time series data is transformed into multidimensional time series data where it is processed with multiple data mining procedures. In the end, it suggests a multidimensional decision support indicator ($mDSI$) to use on any time series data for learning the trends. In general, the $mDSI$ integrates multiple decision criteria into a single time series index value to plot on a chart beside the price or volume in stock analysis. Experiments on the real data confirm its performance.

Rest of this paper is organized into 5 sections. Section 2 introduces the methodology while section 3 discusses experimental work. Section 4 justifies and compares the proposed methodology. Section 5 concludes the work.

2 Methodology

In this section we formally introduce our proposed multidimensional decision support indicator in the presence of multiple features. This indicator is obtained using multiple sequential data mining steps as illustrated in Fig.1. The definitions for these steps are summarized in the following sections:

2.1 Features and Decision Functions

Let x be a time series data point at time t, we denote as x_t, then this point can be represented as a function of time series points as shown below:

$$x_t = l(x_{t-1}, x_{t-2}, ..., x_{t-n}) \tag{1}$$

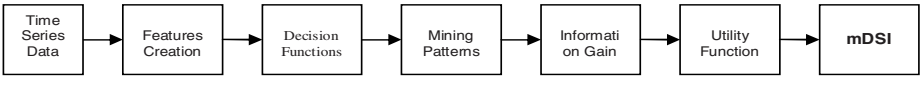


Fig. 1. Steps for constructing multidimensional decision support indicator

A feature f_i in this work is defined as an individual measurable heuristic property of the phenomena being observed over a period of time and we can write this as:

$$f_i = g_i(x_t, x_{t-1}, ..., x_{t-n}) \tag{2}$$

Also a function of time series data points, features or a feature itself can be identified with an technical indicator in stocks. Suppose θ_t is an indicator of price action in stocks at time t, then the θ_t can be written as:

$$\theta_t = \phi(f_1, ..., f_k) \tag{3}$$

where k is the number of features. Moreover, we define a decision criteria in this paper as q mutually exclusive decisions or conclusions as shown below:

$$\delta_{tj} = \begin{cases} \alpha_1 \ if \ f_j(x_t, x_{t-1}, ..., x_{t-n}) = \Delta_1 \\ \alpha_2 \ if \ f_j(x_t, x_{t-1}, ..., x_{t-n}) = \Delta_2 \\ ..... \\ \alpha_q \ if \ f_j(x_t, x_{t-1}, ..., x_{t-n}) = \Delta_q \end{cases} \tag{4}$$

where α and Δ are decision values and decision threshold levels, respectively. If f_1 and f_2 are n_1 days' moving average and standard deviation, then a possible decision criteria, say δ_{t12}, with two mutually exclusive decisions can be designed as:

$$\delta_{t12} = \begin{cases} \alpha_1 \ if \ (f_1 + x_t)/f_2 < \Delta \\ \alpha_2 \ if \ (f_1 - x_t)/f_2 \geq \Delta \end{cases} \tag{5}$$

With k features, m decision criteria can be formed ($m \geq k$) and each decision criteria can be considered, without any loss of generality, as a dimension in m-dimensional space at any given time t. That is, a matrix of multiples of α and m as its rows and columns could be constructed. And each row in this sample space can assume to follow an arbitrary cluster. Let ρ be a function such that:

$$\rho(\delta_1, \delta_2, ..., \delta_m) \in \partial(c_1, .., c_\epsilon, .., c_\nu) \tag{6}$$

with ν clusters (or classes). Let ξ be a weight vector of m decision criteria, $\xi = (\xi_1, \xi_2, ..., \xi_m)$, and decision vector, $\delta = (\delta_1, \delta_2, ..., \delta_m)$, then we write our index function, γ_t, as, $\gamma_t = \psi(\delta, \xi)$

2.2 Estimating Weights

Since the above equation has two unknowns, in this section we propose an information gain based weight evaluation. The information gain of two random variables is a quantity that measures the mutual dependence of the two variables. Intuitively, mutual information measures the information about one variable that is shared by another variable. Using this analogy, this work proposes a methodology to compute the weights through unsupervised learning. As a precursor, a clustering technique is suggested to learn the patterns from the processed multidimensional time series data. Later, a classification is recommended to verify and

identify the corresponding weights through gaining the information. Repeated process of this further encouraged to ensure that any feature is not missed out. Let τ be an information gain parameter, then we write the gain between any δ and ∂ as:

$$\tau(\delta_j;\partial) = \sum\sum p(\delta_j,\partial) log \frac{p(\delta_j,\partial)}{p(\delta_j)p(\partial)} \tag{7}$$

where $p(\delta_j,\partial)$ is the joint probability of δ_j and ∂. $p(\delta_j)$ and $p(\partial)$ are marginal probabilities of δ_j and ∂ respectively. Then the resultant gain becomes a weight of decision criteria j, that is,$\xi_j = \tau(\delta_j,\partial)$ An optimum ξ could be further obtained using the following equation:

$$\xi_j = \frac{\xi_{j1} + \xi_{j2} + ... + \xi_{j\nu}}{\nu} \tag{8}$$

where $\xi_{j\nu}$ is a value of ξ_j obtained from cluster ν. Alternatively, ξ can be computed by choosing a best cluster from the list, that is,$\xi = \xi_\epsilon$ where ξ_ϵ is a weight vector of ϵ clusters subject to $p(\gamma_\epsilon) = min\{p(\gamma_1),..,p(\gamma_\epsilon),..,p(\gamma_v)\}$ and $p(\gamma_\epsilon)$ is the misclassification rate of ϵ classes.

2.3 Our Multidimensional Decision Support Indicator

Without any loss of generality, γ_t can be closely approximated as:

$$\gamma_t = \sum_j \xi_j \alpha_{tj} \tag{9}$$

with α_{tj} as a decision value at time t and ξ_j as a weight of j^{th} decision criteria. Then I_t, the decision support indicator, could be deduced from Eqn.1 as $I_t = \varphi(\gamma_t)$ [11] and it can be written as:

$$I_t = 1 - \frac{\gamma_{max} - \gamma_t}{\gamma_{max} - \gamma_{min}} \tag{10}$$

where $\gamma_{max} = max\{\gamma_1,...,\gamma_t\}$ and $\gamma_{min} = min\{\gamma_1,...,\gamma_t\}$. At time t, the regions of interests such as overbought signal line, U_t and oversold signal line, L_t could easily be derived using statistical control limits for a specified risk as:

$$U_t = \mu(I_t) + s\sigma(I_t) \tag{11}$$

$$L_t = \mu(I_t) - s\sigma(I_t) \tag{12}$$

where μ and σ are the statistical mean and standard deviation of I and s, a scalar quantity which measures distance from the mean.

3 Experimental Evaluation

Experiments conducted with about forty features to study the effects such as short term and long term trends, and distributions of price movements. The

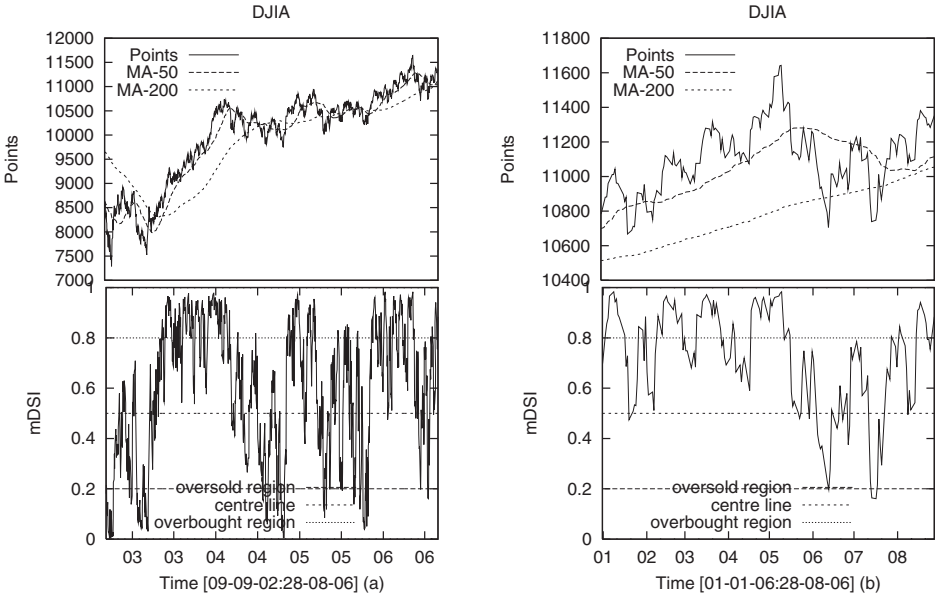


Fig. 2. Indicator points of DJIA and its corresponding mDSI chart

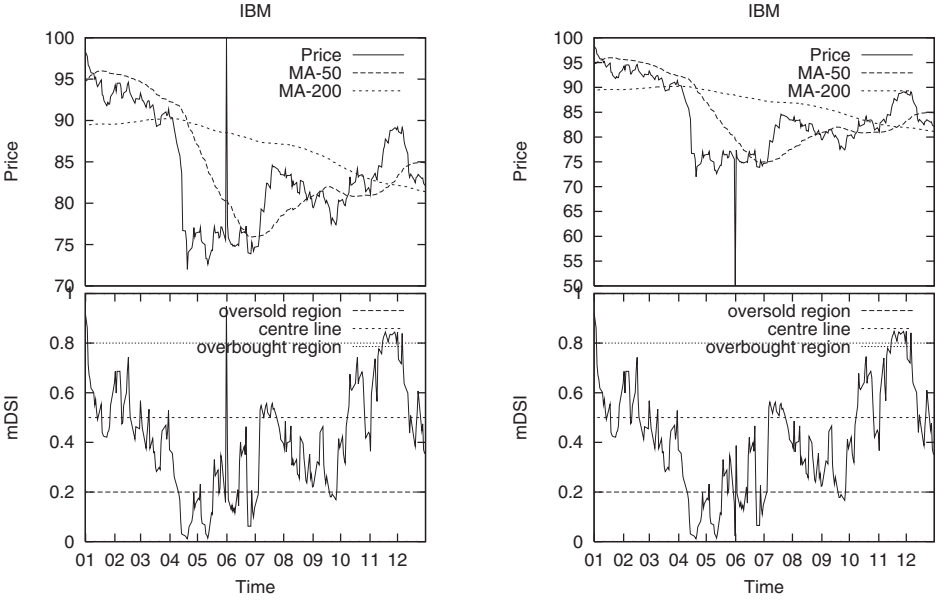


Fig. 3. A simulated example shows the sensitivity of the mDSI Chart - a change was made on 1*st* June by increasing to 100 as well reduced to 50 as shown in (a) and (b), respectively

Table 1. Estimated Weights for Each Decision Criteria Corresponding to the Each Feature

Feat	Clust1 Wts	Clust2 Wts	Feat	Clust1 Wts	Clust2 Wts	Feat	Clust1 Wts	Clust2 Wts	Feat	Clust1 Wts	Clust2 Wts
$f1$	0.1061	0.1090	$f11$	0.6606	0.7079	$f21$	0.0953	0.1939	$f31$	0.7473	0.954
$f2$	0.1518	0.1723	$f12$	0.5290	0.5502	$f22$	0.6221	0.7594	$f32$	0.4779	0.8538
$f3$	0.2458	0.2833	$f13$	0.1972	0.4960	$f23$	0.5855	0.6148	$f33$	0.2673	0.4672
$f4$	0.2918	0.3458	$f14$	0.0459	0.0789	$f24$	0.2384	0.4831	$f34$	0.0007	0
$f5$	0.3673	0.4471	$f15$	0.0863	0.1877	$f25$	0.0677	0.1200	$f35$	0.0003	0.0024
$f6$	0.4505	0.5326	$f16$	0.0095	0.1296	$f26$	0.1151	0.1191	$f36$	0.0023	0.0054
$f7$	0.4837	0.5200	$f17$	0	0.0038	$f27$	0.0974	0.0843	$f37$	0.0006	0.0030
$f8$	0.4912	0.5101	$f18$	0.5020	0.5530	$f28$	0.0493	0.0455	$f38$	0.0084	0.0077
$f9$	0.4321	0.4416	$f19$	0.5358	0.6657	$f29$	0.0217	0.0204	$f39$	0.0062	0.0131
$f10$	0.3670	0.4013	$f20$	0.3419	0.5652	$f30$	0.7865	0.9665	$f40$	0.0010	0.0052

weights of each decision criteria are estimated, from a historical data of Dow Jones industrial average [12] and Nasdaq [13] indices, and [14] using Eqn.9, for various clusters. A sample of these weights are presented in Table 1 for two groups in which group one has two clusters and group two has three clusters. With these weights, the $mDSI$ index values are computed for every period and the results are plotted on the chart as shown in Fig.2 for DJIA index values. The arbitrary limits for overbought and oversold are taken as .8 and .2 at par with RSI. The charts display many times the bullish and bearish signals by staying on top for a while as well as in the bottom. Similarly, the charts exhibit buying and selling opportunities by crossing the limits. Further examination on $mDSI$ is done for its sensitivity to sudden changes or noises in prices. Fig.3 highlights these scenario at two occasions, one with positive changes and another one with negative changes. In both cases, the $mDSI$ is quickly reacting to these changes.

4 Performance

To see its effectiveness when compared to its competitors, its cumulative distribution, correlation and cost are taken into the account. A sample stock of $CREAF$ is examined on $mDSI$, RSI, Stochastics, moving average convergence and divergence (MACD) and MA indicators. The observed trends are plotted in Fig.4. Chart 4(a) highlights trends captured by the different indicators and the graph 4(b) shows their cumulative distributions. In both cases, the $mDSI$ shows its close correlation with its competitors. This is again confirmed in Fig.5 while specifically comparing with RSI indicator. The cost comparison is done with RSI and $mDSI$ charts, for example. With different threshold levels for oversold and overbought, an investment of one unit's growth rate has been tested and its results are presented in Table 2. This table evidences $mDSI$'s good performance at every threshold levels and also at extreme levels over the RSI index. For example, for an investment of $1000 in IBM, the investor gains about $620 when he uses $mDSI$ and $230 when he uses RSI for a four year period. For

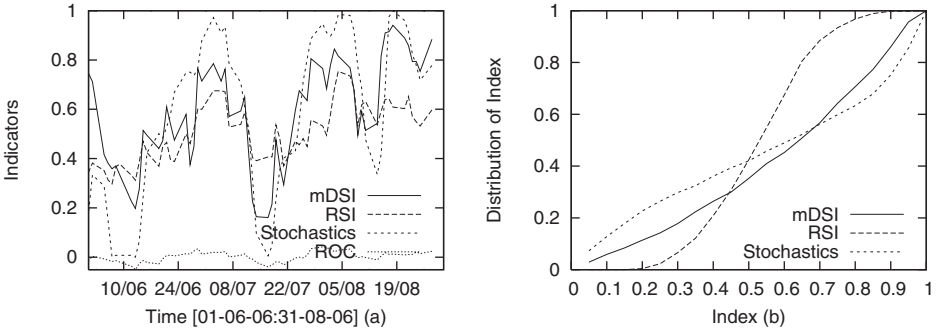


Fig. 4. (a) Indicators are compared with mDSI values for a period of two months in a sample data of DJIA components; (b)Indicators are compared with mDSI by means of distribution for a period of four years in a sample data of DJIA components

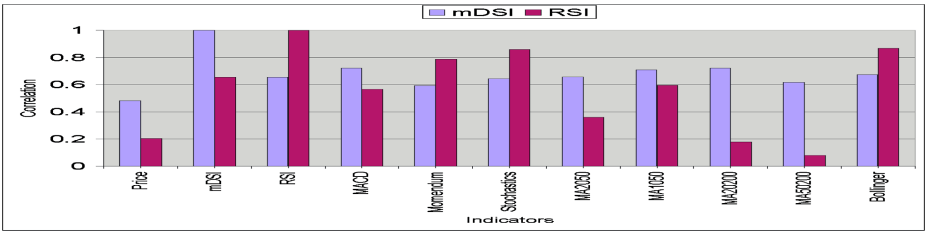


Fig. 5. Correlation levels observed against mDSI and RSI for different indicators

Table 2. Cost comparison with mDSI and RSI for a period of five years

Threshold	IBM RSI	IBM mDSI	MSFT RSI	MSFT mDSI	GE RSI	GE mDSI	Threshold	IBM RSI	IBM mDSI	MSFT RSI	MSFT mDSI	GE RSI	GE mDSI
0.1	1.23	1.62	1.11	2.18	1	1.21	0.3	1.75	1.81	1.78	2.18	1.59	1.43
0.15	1.25	1.75	1.27	1.99	1.24	1.32	0.35	1.91	1.88	2.00	1.90	1.79	1.46
0.2	1.73	1.76	1.43	1.92	1.76	1.39	0.4	1.84	2.03	1.65	2.37	1.82	1.56
0.25	1.77	2.04	1.46	2.19	1.59	1.40	0.45	2.05	2.28	1.87	2.49	1.78	1.75

both indexes, oversold and overbought regions set at 0.9 and 0.1, respectively. However, for better yields, the regions need to be adjusted with optimized limits.

5 Conclusion

This work suggests a general framework for modeling multiple decisions using a utility function and information gain. Methodologies such as classification and clustering are effectively used to calculate the weights of each decision criteria. In particular, a multidimensional decision support indicator ($mDSI$) for predicting stock trends has been proposed. This indicator function monitors stock trends closely and indicates the outliers such as overbought and oversold regions along

with other market conditions. The robustness of the indicator has been tested with real data and observed sufficient evidences for better performance over its competitors. Its preliminary experiment on cost also shows better performance than its immediate competitor such as RSI. Further work is initiated to improve the $mDSI$ and also with price and volumes. This can effectively be used for any time series data.

References

1. Huang, W., Nakamori, Y., and Wang, S.Y.: Forecasting stock market movement direction with support vector machine. Computers & Operations Research **32**(2005) 2513–2522
2. Lijuan, C: Support vector machines experts for time series forecasting. Neurocomputing **51**(2003)321–339
3. Monica, L: Neural network techniques for financial performance prediction: integrating fundamental and technical analysis. Decision Support Systems **37** (2003)567–581
4. Sotiris, Z., Skiadas, C. and Yiannis, V. Technical analysis and mutual funds:Testing Trading Rules, ESIT99 (1999)
5. Murphy, J. J: Technical Analysis of Financial Markets, New York Institute of Finance, USA (1999)
6. Martin, J. P: Technical Analysis Explained, McGraw-Hill Professional, NY (2002)
7. Edwards, R.W. and Magee, J: Technical Analysis of Stock Trends, CRC Press, USA (2001)
8. Leigh, W., Purvis, R., James, M. R: Forecasting the NYSE composite Indicator with technical analysis, pattern recognizer, neural networks, and genetic algorithm: a case study in romantic decision support. Decision Support Systems **32**(2002)361–377
9. Lo, A.W., Mamaysky, H., Wang, J: Foundations of technical analysis: computational algorithms, statistical inference, and empirical implementation. Journal of Finance **55** (2000) 1705-1770
10. Karl, N.: Stock prediction-a neural network approach. Master Thesis, Royal Institute of Technology, KTH (2004)
11. Blank, L: Statistical procedures for engineering, Management and Science, McGraw Hill, New York (1980)
12. Dow Jones & Company, 1 World Finance Center, New York
13. The Nasdaq Stock Market, One Liberty Plaza, New York
14. MSN Money, Microsoft Corporation, USA
15. Wikimedia Foundation, INC, USA
16. Yuehui, C., Ajith, A., Ju Y., and Bo Yang: Hybrid Methods for Stock Indicator Modeling, FSKD 2005, LNAI, **3614**(2005)1067–1070
17. Leung, M.T., Daouk, H., and Chen, A: Forecasting Stock Indices: A Comparison of Classification and Level Estimation Models, International Journal of Forecasting, **16**(2000)173-190
18. Brock, W. , Lakonishok, J., LeBaron, B: Simple Technical Trading Rules and the Stochastic Properties of Stock Returns, The Journal of Finance**XLVII**(1992)1731–1764.

A Novel Support Vector Machine Ensemble Based on Subtractive Clustering Analysis

Cuiru Wang[1], Hejin Yuan[1], Jun Liu[1], Tao Zhou[2], and Huiling Lu[3]

[1] Department of Computer Science, North China Electric Power University
071003 Baoding, Hebei, China
[2] Department of Maths, Shaanxi University of Technology
723000 Hanzhong , Shaanxi, China
[3] Department of Computer, Shaanxi University of Technology
723000 Hanzhong , Shaanxi, China

Abstract. This paper put forwards a novel support vector machine ensemble construction method based on subtractive clustering analysis. Firstly, the training samples are clustered into several clusters according to their distribution with subtractive clustering algorithm. Then small quantities of representative instances from them are chosen as training subsets to construct support vector machine components. At last, the base classifiers' outputs are aggregated to obtain the final decision. Experiment results on UCI datasets show that the SVM ensemble generated by our method has higher classification accuracy than Bagging, Adaboost and k-fold cross validation algorithms.

1 Introduction

Support Vector Machine (SVM) is a practical implementation of structural risk minimization which bounds the generation error by the sum of training error and a term related to *Vapnik-Chervonenkis* dimension of the learning machine. The minimization of this bound lead to a binary classifier with high generalization performance. Whereas, like neural network and other machine learning algorithms, the prediction space of SVM is only an approximation to the hypothesis space because of statistical, computational and representational reasons. Ensemble learning is a promising method for this problem and has been shown as an effective solution for difficult tasks. An ensemble is a set of classifiers whose individual decisions are combined in some way to classify new samples. The research results indicate that ensembles are usually much more accurate than the separate classifier. In recent years, ensemble learning became a hot topic in the fields of machine learning and was regarded as the first direction of current machine learning [1] [2]. The idea of ensemble has been applied for SVM by Kim [3] [4]. Now, SVM ensemble is widely used in many fields, such as news audio classification [5], gene expression analysis [6], cancer recognition [7] and fault diagnoses [8], etc.

In order to constructing a good SVM ensemble, two main problems should be solved: how to generate accurate and diverse base classifiers and how to fuse their outputs effectively. As well known, the first one is the key problem for ensemble learning, so this paper mainly focuses on it. Many methods of constructing ensembles

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 849–856, 2007.

have been developed, such as Bagging [9], Adaboost [10] and k-fold cross validation method. These techniques are very effective for unstable learning algorithms, such as neural network, decision tree and rule-based learning algorithm. However, these methods have some randomness and unrationality because they don't consider the instance distribution information when sampling. Once the training subsets are not appropriate, the ensemble's performance may be even worse than before. To avoid these drawbacks, we put forward a new SVM ensemble based on clustering analysis. In this method, the training samples are firstly been clustered with subtractive clustering algorithm [11]. Then the training subsets are generated by choosing a small quantity of representative samples from each clustering subset. Finally, the SVM components are combined with majority voting method to classify new examples. Since the training examples are chosen from the results of clustering analysis, the training subsets generated by this method may bitterly reflect the actual sample distribution comparing to existing methods. So the SVM ensemble constructed by our method has higher accuracy.

The rest of this paper is organized as follows: section 2 provides a brief review for the basic concepts of support vector machine; section 3 presents an introduction of subtractive clustering algorithm; the main idea and detailed steps for SVM ensemble based on clustering analysis are given in Section 4; section 5 shows the experiment results and analysis of our method on synthetic and UCI datasets; finally, we draw a conclusion in section 6.

2 The Basic Concepts of Support Vector Machine

Suppose we are given a set S of labeled training samples $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$. Each training sample $x_i \in R^m$ belongs to either of two classes and is given a label $y_i \in \{-1, 1\}$ where $i = 1, 2, \cdots, n$. To classify these samples, a SVM will search for a separating hyper-plane with the largest margin. The optimal hyper-plane problem is then regarded as the solution to the following problem

$$\begin{aligned} &\min \frac{1}{2} w \cdot w + C \sum_{i=1}^{n} \xi_i \\ &\text{subject to } y_i (w \cdot x_i + b) \geq 1 - \xi_i \quad i = 1, 2, \cdots, n \\ &\qquad\qquad \xi_i \geq 0 \quad i = 1, 2, \cdots, n \end{aligned} \tag{1}$$

where C is a constant and can be regarded as a regulation parameter. Tune this parameter can make balance between margin maximization and classification violation. The solution of this problem can be obtained by solving its dual formulation:

$$\begin{aligned} &\max W(a) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ &\text{subject to } \sum_{i=1}^{n} \alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C \quad i = 1, 2, \cdots, n \end{aligned} \tag{2}$$

where $\alpha = (a_1, a_2, \cdots, a_n)$ is the vector of nonnegative Lagrange multipliers. The vectors with nonzero coefficient are called support vectors. If the training points are not linear separable in the input space, we need to map them into a high dimensional feature space and find the optimal decision hyper-plane in it with kernel techniques. For this purpose, we only need to change (2) into the forms as following:

$$\begin{aligned} &\max W(a) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i \cdot x_j) \\ &\text{subject to } \sum_{i=1}^{n} \alpha_i y_i = 0 \quad 0 \le \alpha_i \le C \quad i = 1,2,\cdots,n \end{aligned} \tag{3}$$

where $k(\cdot,\cdot)$ is the kernel function.

3 Introduction of Subtractive Clustering Algorithm

Clustering analysis is an unsupervised learning method, by which we can master some basic knowledge about the sample distribution. There are many clustering methods, such as *k* means and competitive learning neural networks [12] [13]. But for these algorithms, it is a hard problem to decide the number of *k* or the number of competitive neurons in prior. Subtractive clustering proposed by Chiu[11] is a simple and effective clustering method. It needn't specify the cluster number in prior. At the same time, this algorithm can have a large reduction on the number of training samples based on the density of surrounding data points.

The algorithm goes as follows:

Step 1: Let X^m be a set of n data points $x_1, x_2, ..., x_n$. Normalize each point into a unit hyper box to make each dimension identical.

Step 2: Compute the potential value for each x_i as:

$$P(x_i, X^m) = \sum_{j=1}^{n} \exp\left(-\frac{\left\| x_i - x_j \right\|^2}{(r_a / 2)^2} \right), i = 1, 2, ..., n \tag{4}$$

where $0 < r_a \le 1$ is the clustering parameter, which defines the neighborhood radius of each point.

Step 3: Select the data point x_i with the highest $P_i (i = 1,2,..., n)$ as the first cluster center. Note the first cluster center as x_1^* and its potential is P_1^*.

Step 4: Reduce the potential value of remaining data points using

$$P_i = P_i - P_1^* \exp\left(-\frac{\left\| x_i - x_1^* \right\|^2}{(r_b / 2)^2} \right) \tag{5}$$

where $r_b > 0$ defines the neighborhood of a cluster center with which the existence of other cluster centers are discouraged. To avoid closely space centers, usually $r_b = 1.5 r_a$.

Step 5: Select the highest potential P_k^* from the reduced potential P_i, and x_k^* is the next candidate cluster center.

Step 6: Compute the potential for the remaining data

$$P_i = P_i - P_k^* \exp\left(-\frac{\left\| x_i - x_k^* \right\|^2}{(r_b / 2)^2} \right) \tag{6}$$

Step 7: Repeat step 5 and step 6 until $P_k^* \Big/ P_k^1 < \delta$. This means the algorithm ends when the current highest potential P_k^* is far smaller than P_1^*.

4 SVM Ensemble Based on Clustering Analysis

4.1 The SVM Ensemble Method Based on Clustering Analysis

Aiming at aforesaid problems, we put forward a new support vector machine ensemble method based on clustering analysis (Abbreviated as SVMECA). Firstly, the positive and negative examples are respectively clustered with subtractive clustering algorithm. And the clustering centers are denoted as $\{C_1^+, C_2^+, \cdots, C_p^+\}$ and $\{C_1^-, C_2^-, \cdots, C_q^-\}$. Then the samples are relabeled according their distance to the cluster centers. Finally we can get p positive subsets, denoted as $\{TR_1^+(x), TR_2^+(x), \cdots, TR_p^+(x)\}$ and q negative subsets, denoted as $\{TR_1^-(x), TR_2^-(x), \cdots, TR_q^-(x)\}$.

The main idea of ***SVMECA*** is as follows:

1) Generating individual training subset and constructing SVM components

Using Bagging, Adaboost, k-fold cross-validation or any other methods to choose certain number of examples from $\{TR_1^+(x), TR_2^+(x), \cdots, TR_p^+(x)\}$ and $\{TR_1^-(x), TR_2^-(x), \cdots, TR_q^-(x)\}$ respectively to generate training subsets, denoted as $\{TR_1^B(x), TR_2^B(x), \cdots, TR_L^B(x)\}$. Then, training support vector machines on each of them to obtain base classifiers.

2) Aggregating the base classifiers' outputs

For any input sample x, using each SVM generated above to classify it and then fusing their outputs in some way to obtain the final decision. Many classifier combination methods have been proposed, such as majority voting, Borda count, Bayes theory, evidence theory, fuzzy integral, LSE-based weighting, double-layer hierarchical support vector net, and so on. Since there is no obvious evidence to evaluate the

SVM components' performance, majority voting method may be a good selection in aggregating step. So in this paper, we use majority voting method to fuse the base classifiers' outputs.

The details about SVMECA are shown as follows:

Algorihtm of SVMECA

Input:

Training Set: $TR = \{(x_i, y_i), i = 1,2,\cdots,n\}$, $x_i \in R^m$, $y_i \in \{-1,+1\}$

L: number of SVM Components

Num: Size of training subsets by which to generate SVM Component and it is smaller than the number of examples included in the original training set

$r_a^+, r_b^+, r_a^-, r_b^-, \delta^+, \delta^-$: Parameters of subtractive clustering algorithm for positive and negative examples, these parameters are set by the user manually.

Output:

$Svm = \{Svm_i, i = 1,2,\cdots,L\}$

Notation:

Svm_i :The support vector machine generated by subset TR_i^B

$Svm_i(x)$: Classification result of x by Svm_i, its value is -1 or 1

Procedure SVMECA

Begin

$\{TR_1^+(x), TR_2^+(x), \cdots, TR_p^+(x)\}$=SubClustering($TR^+, r_a^+, r_b^+, \delta^+$);

$\{TR_1^-(x), TR_2^-(x), \cdots, TR_q^-(x)\}$=SubClustering($TR^-, r_a^-, r_b^-, \delta^-$);

for i=1: L

begin

$TR_i^B = \phi$;

$n' = \left\lceil num / (p+q) \right\rceil$;

for j=1: p

for m=1: n'

begin

$r = rand(\|TR_j^+\|)$;

// r is a random integer between 1 and $\|TR_j^+\|$

$TR_i^B = TR_i^B + \{x_r^+\}$;

// x_r^+ is the rth positive example in TR_j^+

end

for j=1: q

```
        for m=1: n'
        begin
          r = rand(||TR_j^-||) ;
          //r is a random integer between 1 and ||TR_j^-||
          TR_i^B = TR_i^B + {x_r^-} ;
          //x_r^- is the rth negative example in TR_j^-
        end
      Training Svm_i on TR_i^B ;
    end
end
```

For any input example x, the classification result is:

$$label(x) = sign\left(\sum_{i=1}^{L} Svm_i(x)\right) \{|x \in R^m\} \tag{7}$$

Here the samples in the training subsets are chosen from $TR_i^+(x)(i = 1,2,\cdots, p)$ and $TR_j^-(x)(j = 1,2,\cdots, q)$ with equal probability, so they have some representative property and can reflect the actual distribution.

Through constructing smaller training sets and combining different classifier components, we can not only improve the generalization ability and classification accuracy, but also the time and space efficiency of the learning algorithms. Ref [14, 15] used the cluster centers to construct SVM instead of the whole training data set, through which the time and space complexities of the learning algorithm can be greatly decreased. However, this is obtained on the cost of the classification accuracy. Comparing to these methods, we think our algorithm has such two main advantages: 1) SVMECA can improve the generalization ability through ensemble of different base classifiers; 2) The classifier components of ensemble are not constructed by the set of cluster centers but by the representative examples, so the ensemble's accuracy is higher and its complexity is a tradeoff between the set of cluster centers and the whole training dataset.

5 Experiments and Analysis

In order to verify the effectiveness of the algorithm proposed in this paper, we do experiments on sonar, ionoshpere and handwritten digit recognition datasets from UCI machine learning repository. Besides of our method, we also test Bagging, Adaboost and k-fold cross validation algorithms. Here the majority voting is used to aggregate the outputs of the base classifier. The experiments are carried out on PC machine with Pentium 2.0 CPU and 256M memory.

Sonar, ionoshpere and handwritten digits recognition datasets from UCI machine learning repository [16] are used to verify our algorithms. Sonar dataset includes 208 samples from two classes and each sample includes 60 features. One half of them are

Table 1. The statistical results of the different algorithms

Algorithm	Accuracy(%) of different data sets		
	sonar data	ionoshpere	handwritten digits
Single SVM Method	75.96	91.45	89.75
Single SVM(with subtractive clustering centers as traing data)	71.15	67.52	86.15
Bagging method	83.65	94.02	92.80
Adaboost method	77.88	93.16	93.35
k-fold Cross Validation method	84.61	92.31	94.18
SVMECA method	86.65	93.60	96.40

randomly selected as training set, and the remainders are used as testing dataset. Ionosphere dataset includes 351 samples and each sample includes 34 features. Here 2/3 of them are randomly chosen as training set, and the remainders are used as testing dataset. Handwritten digits recognition dataset includes 3823 training samples and 1797 testing samples from 10 classes with 64 features. And the samples of digits '6' and '9' including 750 training samples and 361 testing samples are used for experiments. In the experiments, we take 9 individual SVM components to ensemble with Bagging, Adaboost, k-fold cross validation and out algorithm. For Bagging and Adaboost algorithms, the training subset size of sonar, ionoshpere and handwritten digits are 104, 200 and 600 respectively. While for SVMECA, they are only 50, 78 and 200. The statistical results of classification accuracy are shown in table 1. It can be obviously found that the SVM ensembles generated by SVMECA has better performance than Bagging, Adaboost and k-fold cross validation algorithms.

6 Conclusion

This paper presented a novel construction method for support vector machine ensemble based on clustering analysis. The experiments results on synthetic and UCI datasets show that our method is effective for SVM ensemble. Comparing to existing algorithms, the superiority of SVMECA is that the instances distribution information is considered when generating training subsets by clustering analysis. So, the SVM ensemble generated in this way has better performance since the samples in the training subsets have good representative property.

References

1. Dietterich TG, Machine Llearning Research: Four Current Directions, AI Magazine, 1997, 18(4):97-136.
2. Dietterich TG, Ensembles in Machine Learning , Multiple Classifier Systems, Lecture Notes in Computer Science,2000:1-15.
3. Kim H C, Pang S, Je H M, *et al*, Constructing Support Vector Machine ensemble, Pattern Recognition, 2003,36(12):2757-2767.

4. Kim H C, Pang S, Je H M, *et al*, Pattern Classification Using Support Vector Machine Ensemble, Los Alamitos CA: Proceedings of the 16th International Conference on Pattern Recognition,2002:160-163.
5. Bing Han, Xinbo Gao, Hongbing Ji, Automatic News Audio Classification Based on Selective Ensemble SVMs, Second International Symposium on Neural Networks, Chongqing, China, , 2005: 363-368.
6. Valentini G, Muselli M, Ruffion F, Bagged Ensembles of Support Vector Machines for Gene Expression Data Analysis, Proceedings of the International Joint Conference on Networks,2003
7. Valentini G, Museli M, Ruffino F, Cancer Recognition with Bagged Ensembles of Support Vector Machines , Neural Computing ,2004:461-466.
8. Li Ye, Cai Yun-ze, Xu Xiao-ming, Fault Diagnosis Based on SVM Ensemble, Control Engineering of China, 2005,7:170-173.
9. Breiman L, Bagging Predictors, Mach. Learning, 1996, 24(2):123-140.
10. Freund Y, Schapire R, A Decision Theoretic Generalization of Online Learning and an Application to Boosting, Journal of Computer System Science, 1997,55(1):119-139.
11. Chiu S, Fuzzy Model Identification Based on Cluster Estimation.,Journal of Intelligent and Fuzzy Systems, 1994,2(3):267-278.
12. Kohonen T, The Self-Organization Map,Proc. IEEE, vol 78, 1990:1464-1480.
13. Desieno D,Adding a Conscience to Competitive Learning, Proc IEEE Internal Conference on Neural Networks,1988:117-124.
14. Zhang YN, Zhao RC, Leung Y, An Efficient Target Recognition Method for Large Scale Data, Acta Electronica Sinica, 2002, 30(10):1533-1535.
15. Wu S, Niu XX, Liu HB, Support vector Machines Based on Subtractive Clustering, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 2005:4345-4350.
16. http://www.ics.uci.edu/~mlearn/MLRepository.html

Keyword Extraction Based on PageRank

Jinghua Wang, Jianyi Liu, and Cong Wang

Beijing University of Posts and Telecommunications 100876 Beijing China
{wangjh, liujy, wangc}@nlu.caai.cn

Abstract. Keywords are viewed as the words that represent the topic and the content of the whole text. Keyword extraction is an important technology in many areas of document processing, such as text clustering, text summarization, and text retrieval. This paper provides a keyword extraction algorithm based on WordNet and PageRank. Firstly, a text is represented as a rough undirected weighted semantic graph with WordNet, which defines synsets as vertices and relations of vertices as edges, and assigns the weight of edges with the relatedness of connected synsets. Then we apply UW-PageRank in the rough graph to do word sense disambiguation, prune the graph, and finally apply UW-PageRank again on the pruned graph to extract keywords. The experimental results show our algorithm is practical and effective.

1 Introduction

Keyword extraction is an important technology in many areas of document processing, such as text clustering [1], text summarization, and text retrieval [2,3]. Keywords are viewed as the words that represent the topic and the content of the whole text [4]. A most popular algorithm for keyword extraction is *tfidf* measure, which extracts keywords that appear frequently in a document while seldom in the remainder documents of the corpus [2]. But *tfidf* measure has two disadvantages: Sometimes, there is no corpus for computing *idf*. Synonym's *tf* is viewed independent and may decrease the precision.

This paper presents a new algorithm that represents the text as a semantic graph with synset from WordNet, disambiguates the words, and finally extracts keywords from the text based on UW-PageRank. It needs no corpus, has the ability to disambiguate all the words in the text, and extracts keywords by analyzing the semantic structure of the whole text. The experiment result shows that our algorithm is effective and practical.

2 PageRank on Semantic Networks

PageRank is an algorithm of deciding the importance of vertices in a graph. WordNet can be viewed as an undirected weighted graph, which defines synsets as vertices and relations of synsets as edges and assigns the weight of edges by the relatedness of connected synsets. For PageRank formula is defined for directed graph, a modified PageRank formula is applied to use on the undirected weighted graph from WordNet.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 857–864, 2007.

2.1 PageRank

PageRank [5] which is widely used by search engines for ranking web pages based on the importance of the pages on the web is an algorithm essentially for deciding the importance of vertices within a graph. The main idea is that: in a directed graph, when one vertex links to another one, it is casting a vote for that other vertex. The more votes one vertex gets, the more important this vertex is. PageRank also takes account the voter: the more important the voter is, the more important the vote itself is. In one word, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertex casting these votes. So this is the definition:

Let $G=(V,E)$ be a directed graph with the set of vertices V and set of edges E, when E is a subset of $V \times V$. For a given vertex V_i, let $In(V_i)$ be the set of vertices that point to it, and let $Out(V_i)$ be the set of edges going out of vertex V_i. The PageRank score of vertex V_i is

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{S(V_j)}{\left|Out(V_j)\right|} \tag{1}$$

d is a damping factor that can be set between 0 and 1,and usually set at 0.85 which is the value we use in this paper [5].

PageRank starts from arbitrary values assigned to each vertex in the graph, and ends when the convergence below a given threshold is achieved. Experiments proved that it usually stops computing within 30 iterations [6].

PageRank can be also applied on undirected graph, in which case the out-degree of a vertex is equal to the in-degree of the vertex.

2.2 WordNet as Semantic Networks

WordNet is an online lexical reference system. English nouns, verbs, adjectives and adverbs are organized into synonym sets or synsets related by defined relations such as hypernymy/ hyponymy and holonymy / meronymy.

2.2.1 The Relatedness of Sense

Glosses of synset meanings in WordNet and the networked arrangement of synsets are both utilized as sources to determine the relatedness of a pair of synsets. Lesk [7], Wilks [8], Banerjee and Pedersen [9][10] use the glosses of the words, while Rada [11], Wu & Palmer[12], Leacock & Chodorow [13] computed the relatedness of two senses based on the structure of WordNet.

2.2.2 PageRank on WordNet

WordNet can be represented as a graph, in which synsets are defined as vertices, and relations of synsets are defined as edges. The graph can be constructed as an undirected graph. The edges can be weighted by the “strength” of the connection between two vertices, i.e. synsets, and computed by the measures of semantic relatedness. We applied PageRank on the undirected weighted graph from WordNet with a modified formula

$$S_{UW}(V_i) = (1-d) + d * \sum_{j \in C(V_i)} \frac{weight(E_{ij}) S_{UW}(V_j)}{\left| D(V_j) \right|} \quad (2)$$

$C(V_i)$is the set of edges connecting with V_j, $weight(E_{ij})$is the weight of edge E_{ij} connecting vertex V_i and V_j, and $D(V_j)$ is the degree of V_j. This formula is named UW-PageRank.

3 Keyword Extraction Based on PageRank

To extract keywords from the text, firstly the text is converted to an undirected weighted semantic graph based on WordNet. All the senses for all words defined in WordNet form the vertices of the graph and the relations are the edges of the graph. At that time, one word usually has more than one corresponding vertices in the graph, but only one is actually the real meaning while others are noise. So secondly, we disambiguate all the words in the text with UW-PageRank (Formula (2)). After that, by deleting all the other vertices and their connected edges, one word corresponds to one synset and its corresponding vertex is left in the graph. Finally, we use UW-PageRank again to find the most important vertices in the graph and the corresponding words in the text are the keywords.

3.1 Text Representation as a Graph

To use PageRank algorithm to exact keyword of the text, a graph which represents the text and interconnects the words with meaningful relations should be build first. We made a hypothesis that "The same word in a text segment has the same sense", which is proved to be acceptable in our experiments in Section 4. All the words in the text should be POS tagged first, and then find all the synsets pertaining to the word in WordNet with its POS. Synsets form the vertices of the graph. Edges are added between the vertices which have a relation in WordNet between them. The weights of the edges which represent the relatedness of two synsets can be computed by the algorithm of Pedersen introduced in Section 2.2.1.

There is a special situation while building the graph, that *co-lexical synsets* which is defined as synsets that represent senses of the same word, have a relation defined in WordNet [14]. Co-lexical synsets are competing for one word. Therefore, only one of the co-lexical synsets is the "correct" one to be left and the others are all noise. So no edges would be added between co-lexical synsets.

Assumed that a text containing "*word1 word2 word3*" is to be represented in a graph.*Word1* has three synsets defined in WordNet which are S1, S2 and S3 represented in graph; *Word2* has one synset S4 and *Word3* has two, S5 and S6. None co-lexical synsets are linked together with the weight of relatedness. For example, S1 and S4 are connected with an edge weighted 0.2. The graph represented the text is in Fig.1(a).

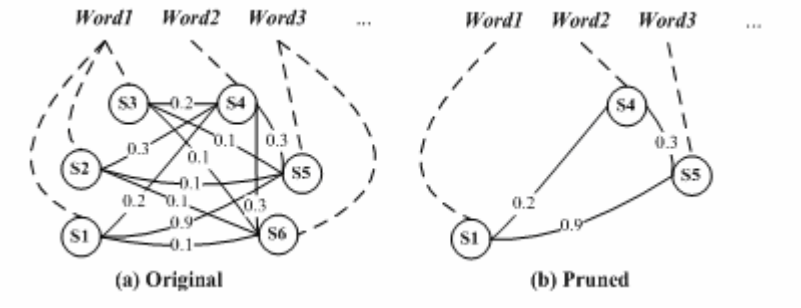


Fig. 1. Text represented in graph

3.2 Word Sense Disambiguation Based on PageRank

The ambiguity problem is reflected in our model as extra senses pertaining to co-lexical synsets. It is a great obstacle for the task of keyword extraction. So our first step is to do word sense disambiguation. And our goal is to find the one and only synset for each word, thus leave the vertices pertaining to the exclusive synset in the graph while deleting all the others. There are two disambiguation approaches: knowledge-based and corpus-based methods [15]. Knowledge-based method disambiguates words by matching context with information from a prescribed knowledge source, such as WordNet. Agirre and Figau [16] present a method for the resolution of the lexical ambiguity of nouns using the WordNet noun taxonomy and the notion of conceptual density. Magnini and Strapparava [17] explored the role of domain information in word sense disambiguation using WordNet. Mihealcea [14,18] use a PageRank-style algorithm applied on a WordNet-based concepts graph to do word sense disambiguation for free text.

We use UW-PageRank (Formula (2)) to score all the vertices in synset graph built from the text based on WordNet. The higher score one vertex (synset) gets, the more important it is in the graph, therefore more likely the word tends to choose the vertex, i.e. synset. The UW-PageRank score of synsets in Fig.1 is as following.

Table 1. UW-PageRank score of vertices in Fig.1

vertex	S1	S2	S3	S4	S5	S6
UW-PageRank score	0.203	0.169	0.166	0.211	0.222	0.176

Co-reference sense and the serial number of the sense defined in WordNet are also taken into consideration while assign a sense to a word as well as the UW-PageRank score. Co-reference sense is defined as the sense that pertaining to more than one words from the text (recall that different word shapes from the same word are viewed as exactly the same word). For example, “task” pertains 2 senses, and “job” pertains 13. For job#2 and task#2 are the same sense “{00708623}—a specific piece of work repaired to be done as a duty or for a specific fee”. If word “task” and word “job” are both in the text, sense {00708623} is the co-reference sense of word “task” and “job”.

If other conditions of this sense and other sense pertaining to the word are the similar, this sense should have more chance to be assigned to the word “task” and “job”. Yet this aspect of the sense can not be reflected in the structure of the group, so additional priority should be taken into consider for this situation.

WordNet makes a statistical comparison of different sense of the same word on a large sense-annotated corpus and the senses are ordered from most frequent to least. Therefore, words trend to choose more frequent sense as its correct sense.

Our algorithm combines UW-PageRank score, co-reference sense priority and the frequency priority together, and gives an integrated evaluation to each sense. For each word in the text, it will choose the sense having the highest score as its correct synset. Words of graph in Fig.1 are disambiguated that *Word1* has the sense of S1, *Word2* has S4 and *Word3* has S5. So the graph is pruned to Fig.1(b).

3.3 Keyword Extraction Based on UW-PageRank

Word sense disambiguation step assigns one sense to one word, and deletes all the “wrong” vertices bonding to the “wrong” sense and the edges connecting to them. Therefore, there are only vertices bonding to chosen sense and original edges between them in the graph. Consequently, the word, sense and the vertex forms a many-one-one relation. Recall that the essential meaning of UW-PageRank algorithm is to judge the importance of vertices in a graph, so we use UW-PageRank on the pruned undirected weighted graph to measure all the vertices in the graph. The vertex that have higher UW-PageRank score is considered to be more important in the graph, so the sense bonding to the vertex is considered to be more important in the text. We can choose the top “n” words as keywords of the text or choose all the words above a defined threshold. Actually, UW-PageRank finds the most important sense of the text, however there may be more than one words bonding to the sense and these words are all synonymous words, so we just choose the most frequent one of them as the keyword. The UW-PageRank score of the synset vertices in Fig.2 is in Table 2.

Table 2. UW-PageRank score of vertices in Fig.2

vertex	S1	S4	S5
UW-PageRank Score	0.276	0.209	0.282

From the result, we can see that S5 is the most important vertex while S4 is the least. The importance serial of the words in text is *Word3*, *Word1* and *Word2*.

4 Experiment and Evaluation

Our task is to extract keywords from a single document without any training sets. There is an important procedure in our algorithm to disambiguate word sense for all the words in the document. The performance of word sense disambiguation is tested on SemCor 2.1, which contains 186 documents containing more than 2000 words for each document. Words in the documents are manually annotated with part-of-speech and corresponding senses in WordNet.

The hypothesis that "the same word in a text segment has the same sense" is tested with a definition of word-sense-uniform-ratio.

$$word\text{-}sense\text{-}uniform\text{-}ratio = \frac{the\ number\ of\ word\ having\ the\ "main\ sense"}{the\ total\ number\ of\ the\ word\ in\ the\ text\ segments} \quad (3)$$

"main sense" is defined as that for each word the most usually sense it choose in the text segment. We choose sentence, paragraph and whole text as the unit of text segment, and get the word-sense-uniform-ratio.

Table 3. Word-sense-uniform-ratio

Text segment unit	sentence	paragraph	Whole text
Word-sense-uniform-ratio	99.19%	96.60%	87.47%

Word-sense-uniform-ratio gets lower while the text segment unit gets larger. But even the lowest ratio still could be accepted. So we can accept the hypothesis at any text segment unit from above.

Our word sense disambiguation algorithm is tested on SemCor2.1. We choose sentence, paragraph and whole text as text segment units to build the semantic graph. Each word is POS tagged and found in WordNet2.1 to get all the senses. If one word can not be found in WordNet2.1, usually is a new word, we just have to discard it. All the senses form the vertices of the semantic graph and the weight of the edges are computed with the algorithm mentioned in section 2.2.1. We also tag all the words with the most frequent sense in WordNet and take this performance as the baseline.

Table 4. Word sense disambiguation precision

Text Segment unit	Sentence	paragraph	whole text	Baseline
Precision	78.1%	80.2%	75.0%	76.7%

Paragraph as text segment unit gets the highest precision, and "whole text" gets lowest. It is proved that the text segment unit should choose a reasonable size: if the unit is too large, there are too many irrelevant words and senses involved in the graph, which become noise for the disambiguation of other words. If the unit is too small, there are not enough words and senses to build a compact graph, therefore the computation of UW-PageRank is not reliable. From the result, we can see that the paragraph as a unit is just suitable for word sense disambiguation, for there are several sentences containing tens of words about an event or concept just able to build a suitable scale of semantic graph.

Keywords are attached to the content of the text; however they are not defined in a consistent way. Therefore, we used author-based evaluation. Fifty technical papers about artificial intelligence of CISTR (Center of Intelligent Science and Technology Research) are involved in the experiments as test corpus for keyword extraction evaluation. We choose 5 of the keywords (just words, no phrase) assigned by the author for each paper as the results. Then, we use our algorithm to extract top 10 words from the paper. As a comparison, *tf* is also used to extract keywords.

Elimination of stop words and stemming are processed ahead and 10 most frequent words of the text are extracted as keywords. Precision is the result of the number of correct keyword divided by the number of extracted words .Precision has a limitation of 0.5 for authors only assigned 5 while our algorithm extracts 10. Coverage is the result of the number of correct keyword divided by the number of keywords the author has assigned.

Table 5. Keyword extraction comparison result

	Coverage	Precision
***tf* without word sense disambiguation**	0.49	0.24
***tf* with word sense disambiguation**	0.54	0.27
UW-PageRank with word sense disambiguation	0.69	0.34

Results are shown in Table 5. *tf* selects terms which appear frequently in a document, however there are many synonyms in the document and *tf* views these synonyms as different words, which weakens the performance of *tf*. After disambiguating each word, synonyms' term frequencies are added together and the coverage and precision both increase, which proves that our algorithm for word sense disambiguation is necessary and effective. *tf* ignores semantic structure of a document, transforms the document form a string of characters into a sequence of words, and assumes the words is independent. While UW-PageRank represents a text as semantic graph with synset from WordNet, disambiguates all the words in the text, decides the importance of vertices within the semantic graph, and regards those top "n" important vertices as keywords. Therefore, UW-PageRank can detect some "hidden" keywords even if they do not appear frequently, which are not been extracted by *tf*.

5 Conclusion

Keywords are viewed as the words that represent the topic and the content of the whole text. This paper proposed a keyword extraction algorithm based on WordNet and PageRank. Firstly, a free text is represented as an undirected weighted semantic graph based on WordNet which defines synsets as vertices and relations of vertices as edges, and assigns the weight of edges with the relatedness of connected synsets. The second step is to disambiguate words referring to UW-PageRank score, co-reference sense priority and the frequency priority. Then, graph is pruned leaving only the "correct" synset vertices. Finally, UW-PageRank is used again to extract key synset vertices in the graph, and the corresponding words are assigned as the keywords. Our algorithm is tested on SemCor2.1 and corpus of CISTR, and the experiment results proves our algorithm to be practical and effective.

Acknowledgement

This study is supported by National Natural Science Foundation of China (60575034).

References

1. Liu, Y., Wang, X., Liu, B.,Zhong, B., The clustering analysis technology for information retrieval, Vol. 28 No.4, Journal of Electronics & Information Technology, (2006) 606-609
2. Matsuo, Y., Ishizuka, M., Keyword Extraction from a Single Document using Word Cooccurrence Statistical Information, Proceedings of the 16th International FLAIRS Conference, St. Augustine, Floridam (2003)
3. Li, S., Wang, H., Yu, S., and Xin, C., Research on maximum entropy model for keyword indexing, Vol.27 No.9, Chinese Journal of Computers, (2004) 1192-1197
4. Ramakrishnan, G., Bhattacharyya, P., Using Wordnet Based Semantic Sets for Word Sense Disambiguation and Keyword Extraction, International Conference on Knowledge Based Computer Systems (KBCS 2002), Mumbai, India, (2002)
5. Page, L., Brin, S., Motwani, R., and wingorad, T., The pagerank citation ranking: Bringing order to the web Technical report, Stanford Digital Library Technologies Project, (1998)
6. Paul Tarau, Rada Mihalcea and Elizabeth Figa, Semantic Document Engineering with WordNet and PageRank, in Proceedings of the ACM Conference on Applied Computing (ACM-SAC 2005), New Mexico, (2005)
7. Lesk, M., Automatic sense disambiguation using machine readable dictionaries: how to tell a pine code from an ice cream cone, in: Proceedings of the 5th annual international conference on systems documentation, ACM Press, (1986) 24-26
8. Wilks, Y., Fass, D., Guo, C., MeDonald, J., Plate, T., Slator, B., Providing machine tractable dictionary tools, Machine Translation 5 (1990) 99-154
9. Banerjee, S., Pedersen, T., An adapted Lesk algorithm for word sense disambiguation using WordNet in Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, (2002)136-145
10. Pedersen, T., Banerjee,S., Patwardhan,S., Maximizing Semantic Relatedness to Perform Word Sense Disambiguation, Research Report UMSI 2005/25, University of Minnesota Supercomputing Institute, March (2005)
11. Rada, R., Mili,E,.Bicknell, Blettner, M., Development and application of a metric on semantic nets, IEEE Transactions on Systems, Man and Cybernetics 19(1) (1989)17-30
12. Wu, Z., Plamer, M., Verb semantics and lexical selection, in 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, (1994) 133-138
13. Leacock, C., Chodorow, M., Combing local context and WordNet Similarity for word sense identification, in: C.Fellbaum (Ed.), WordNet: An electronic lexical database, MIT Press, (1998) 305-332
14. Mihalcea, R., Tarau, P., Figa, E., PageRank on Semantic Networks, with application to Word Sense Disambiguation, in Proceedings of The 20st International Conference on Computational Linguistics (2004)
15. Montoyo, A., Suarez, A., Rigau, G. and Palomar, M. Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods, Volume 23, Journal of Machine learning research , (2005) 299-330.
16. Agirre, E., Rigau,G., Word Sense Disambiguation using Conceptual Density. In Proceedings of the International Conference on Computational Linguistic COLLING'96 Copenhagen, Denmark (1996)
17. Magini, B., Strapparava, C., Pezzulo, G., &Gliozzo, A., The Role of Domain Information in Word Sense Disambiguation. Natural Language Engineering, 8(4), (2002)359-373
18. Mihalcea,R., Graph-based ranking algorithms for sentence extraction applied to text summarization. In Proceedings of 42ed Annual Meeting of the Association for Computational Linguistics (2004)

Finding the Optimal Feature Representations for Bayesian Network Learning

LiMin Wang[1], ChunHong Cao[2], XiongFei Li[1], and HaiJun Li[3]

[1] Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, JiLin University, ChangChun 130012, China
jeffreywlm@sina.com
[2] College of Information Science and Engineering, Northeastern University, ShenYang 110004, China
[3] College of Computer Science, YanTai University, YanTai 264005, China

Abstract. Naive Bayes is often used in text classification applications and experiments because of its simplicity and effectiveness. However, many different versions of Bayes model consider only one aspect of a particular word. In this paper we define an information criterion, Projective Information Gain, to decide which representation is appropriate for a specific word. Based on this, the conditional independence assumption is extended to make it more efficient and feasible and then we propose a novel Bayes model, General Naive Bayes (GNB), which can handle two representations concurrently. Experimental results and theoretical justification that demonstrate the feasibility of our approach are presented.

1 Introduction

With the ever-increasing growth of the on-line information and the permeation of Internet into daily life, web document classification plays an important role in natural language processing (such as E-mail filtering [1], news filtering [2], prediction of user preferences [3]) and information retrieval applications from the linguistic point of view. Because of the variety of languages, applications and domains, machine learning techniques are commonly applied to infer a classification model from instance documents with known class labels. The inferred model can then be used for web document classification, i.e. classification based on text content.

Naive Bayes [4] [5] has been found particularly attractive for the task of text categorization because it performs surprisingly well in many application areas despite its simplicity. Naive Bayes makes the strong assumption that the predictive variables ("Features" or "Words") are conditionally independent given the class. From the linguistic point of view, a document is made up of words, and the semantics of the document is determined by the meaning of the words and the linguistic structure of the document. The generative model underlying the Naive Bayes can be characterized with respect to the amount of information it captures about the words in a document. In information retrieval and text categorization, the most popular probabilistic models are: the multinomial model [7] [8] and the binary independence model [6].

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 865–870, 2007.

The binary independence model specifies that a document is represented by a vector of binary features indicating which words occur and do not occur in the document. It captures the information of which words are used in a document, but not the number of times each words is used, nor the order of the words in the document. This describes a distribution based on a multi-variate Bernoulli event model. This approach is more appropriate for tasks that have a fixed number of features. The multinomial model specifies that a document is represented by the set of word occurrences from the document. And we call these words multinomial features for clarity. This approach is more traditional in statistical language modeling for speech recognition, where it would be called a "unigram language model."

Before learning, the characteristic of an feature should be analyzed to decide which kind of representation is appropriate. On the other hand, the multinomial model and the binary independence model can not handle a much more complex situation, that is, both representations are required to appear in the same model. In this paper, we extend the independence assumption to make it more efficient and feasible and then propose General Naive Bayes (GNB) model, which can handle both representations concurrently. The remainder of this paper is organized as follows. Sect. 2 defines an information gain criterion to decide which representation is appropriate for a given feature. Sect. 3 describes the basic idea of General Naive Bayes. Sect. 4 presents the corresponding experimental results of compared performance with regarding to the multinomial model and the binary independence model. Sect. 5 wraps up the discussion.

2 The Projective Information Gain $PI(C; V_i)$

In this discussion we use capital letters such as V_i, V_j to denote feature names, and lower-case letters such as v_i, v_j to denote specific values taken by those features. Let $P(\cdot)$ denote the probability, $p(\cdot)$ refer to the probability density function. Here d_i denotes the ith training document and c_i is the corresponding category label of d_i. A document d is normally represented by a vector of n features or words $d = (v_1, v_2, \cdots, v_n)$.

Entropy is commonly used to characterize the impurity of an arbitrary collection of instances. But Entropy has limitation when dealing with multinomial representation. In the case of binary features, the set of possible values is a numerable set. To compute the conditional probability we only need to maintain a counter for each feature value and for each class. In the case of multinomial features, the number of possible occurrences is infinite, thus make it impossible to compute conditional entropy.

We begin by adding two implicit features $\hat{C}$ and $\tilde{C}$. Let V_i represent one of the predictive features. According to Bayes theorem, if V_i is a binary feature, there will be

$$P(c|v_i) = \frac{P(c)P(v_i|c)}{P(v_i)} \tag{1}$$

Since $P(v_i)$ is the same for all classes, and does not affect the relative values of their probabilities, it can be ignored. When some instances satisfy $V_i = v_i$, their class labels are most likely to be:

$$\hat{C} = \arg\max_{c \in C} P(c|v_i) = \arg\max_{c \in C} P(c)P(v_i|c) \tag{2}$$

Correspondingly, if V_i is a multinomial feature we will have

$$\tilde{C} = \arg\max_{c \in C} P(c|v_i) = \arg\max_{c \in C} \frac{P(c)p(v_i|c)}{p(v_i)} = \arg\max_{c \in C} P(c)p(v_i|c) \tag{3}$$

Then, the corresponding relationship between V_i, C, $\hat{C}$ and $\tilde{C}$ may be:

Table 1. The relationship between V_i, C, $\hat{C}$ and $\tilde{C}$

V_i	C	$\hat{C}$	$\tilde{C}$
v_{i1}	c_1	c_1	c_1
v_{i2}	c_2	c_1	c_2
.	.	.	.
.	.	.	.
.	.	.	.
v_{iN}	c_2	c_2	c_1

Accordingly, the conditional entropies of C are:

$$H(C|\hat{C}) = -\sum_{\hat{c} \in C} P(\hat{c}) \sum_{C} P(c|\hat{c}) \log P(c|\hat{c})$$

and

$$H(C|\tilde{C}) = -\sum_{\tilde{c} \in C} P(\tilde{c}) \sum_{C} P(c|\tilde{c}) \log P(c|\tilde{c})$$

On the other hand, the entropy of C is

$$H(C) = -\sum_{c \in C} P(c) \log P(c) \tag{4}$$

The Projective Information Gain $PI(C; V_i)$ is defined as

$$\begin{aligned} PI(C; V_i) &= \arg\max\{I(C; \hat{C}), I(C; \tilde{C})\} \\ &= \arg\max\{H(C) - H(C|\hat{C}), H(C) - H(C|\tilde{C})\} \end{aligned} \tag{5}$$

Where $I(\cdot)$ denotes the mutual information. $I(C; \hat{C})$ and $I(C; \tilde{C})$ describe the extent to which the model constructed by feature V_i fits class feature C when V_i is treated as binary feature or multinomial feature, respectively. Then we compare them to choose the right representation and that is what $PI(C; V_i)$ means.

3 General Naive Bayes (GNB)

Naive Bayes is one of the most straightforward and widely used method for probabilistic induction. This scheme represents each class with a single probabilistic summary and is based on one assumption that predictive features $V_1, \cdots, V_n$ are conditionally independent given the category label C, which can be expressed as:

$$P(v_1, \cdots, v_n|c) = \prod_{i=1}^{n} P(v_i|c) \tag{6}$$

But if multinomial features are required to represent words, the situation is different. Since the independence assumption described above is applicable to binary features only, we should extend it to make it much more effective. For simplicity, we first just consider two features: V_1 (multinomial) and V_2 (binary). Suppose the word frequency values of V_1 have been normalized and discretized, then the independence assumption should be:

$$P(v_1 \leq V_1 \leq v_1 + \triangle, v_2|c) = P(v_1 \leq V_1 \leq v_1 + \triangle|c)P(v_2|c). \tag{7}$$

where $[v_1, v_1 + \triangle]$ is arbitrary interval of the values of feature V_1. This assumption, which is the basis of GNB, supports very efficient algorithms for both classification and learning. By the definition of a derivative,

$$\begin{aligned} P(c|v_1 \leq V_1 \leq v_1 + \triangle, v_2) &= \frac{P(c)P(v_1 \leq V_1 \leq v_1 + \triangle|c)P(v_2|c)}{P(v_1 \leq V_1 \leq v_1 + \triangle|v_2)P(v_2)} \\ &= \frac{P(c)p(\zeta|c)\triangle P(v_2|c)}{p(\eta|v_2)\triangle P(v_2)} \\ &= \frac{P(c)p(\zeta|c)P(v_2|c)}{p(\eta|v_2)P(v_2)} \end{aligned} \tag{8}$$

where $v_1 \leq \zeta, \eta \leq v_1 + \triangle$. When $\triangle \to 0$, $P(c|v_1 \leq V_1 \leq v_1 + \triangle, v_2) \to P(c|v_1, v_2)$ and $\zeta, \eta \to v_1$, hence

$$\lim_{\triangle \to 0} P(c|v_1 \leq V_1 \leq v_1 + \triangle, v_2) = P(c|v_1, v_2) = \frac{P(c)p(v_1|c)P(v_2|c)}{p(v_1|v_2)P(v_2)} \tag{9}$$

Suppose the first m of n features are multinomial and the remaining features are binary. Similar to the induction process of (9), we will have

$$P(c|v_1, \cdots, v_n) = \frac{P(c)\prod_{i=1}^{m} p(v_i|c)\prod_{j=m+1}^{n} P(v_j|c)}{p(v_1, \cdots, v_m|v_{m+1}, \cdots, v_n)P(v_{m+1}, \cdots, v_n)} \tag{10}$$

Based on Eq. (10), maximum a posterior (MAP) classifier can be constructed by seeking the optimal category which maximizes the posterior $P(c|d)$, then the classification rule of GNB is:

$$C^* = \arg\max_{c \in C} P(c|v_1, \cdots, v_n) = \arg\max_{c \in C} P(c) \prod_{i=1}^{m} p(v_i|c) \prod_{j=m+1}^{n} P(v_j|c) \tag{11}$$

4 Experiments

We compare classification accuracy with and without word frequency information on two datasets: 20 Newsgroups [9] and Reuters-21578 [10] collection. We produce 20 train/test splits using stratified random sampling with 90% of the data for training and 10% for testing. Experiments are done using 20 trials over these splits and averaging the results. We carried the experiment to compare the General Naive Bayes model, the binary independence model [11] and the multinomial model [12] when different number of unlabelled test data are incorporated into the training set. Fig. 1 shows the average results on Reuters-21578 corpus, with various vocabulary sizes. The effectiveness of Bayes classifier was measured by classification accuracy.

All the three models do best at the maximum vocabulary sizes. Multinomial model achieves 81% accuracy and the binary independence model achieves 68% accuracy. Using GNB improves performance by more than 3 percentage points. The binary independence model is more sensitive to query length than the multinomial model. The multinomial model treats each occurrence of a word in a document independently of any other occurrence of the same word. It can be argued that the binary statistics used in the binary independence model are more robust than the ones used in the multinomial model. The violations of the multinomial model's independence assumption may be more detrimental than the corresponding violations for the binary independence model. In particular, there is one effect to which the multinomial model is subject which has no counterpart in the binary independence model. It is likely that many terms, especially the kind that are likely to be good discriminators, are strongly dependent on their own occurrence. Overall, however, the results show that performance of the GNB model on this task is substantially better than that of the other two Bayes models. We hypothesize that this is primarily due to the independence assumption of the GNB model.

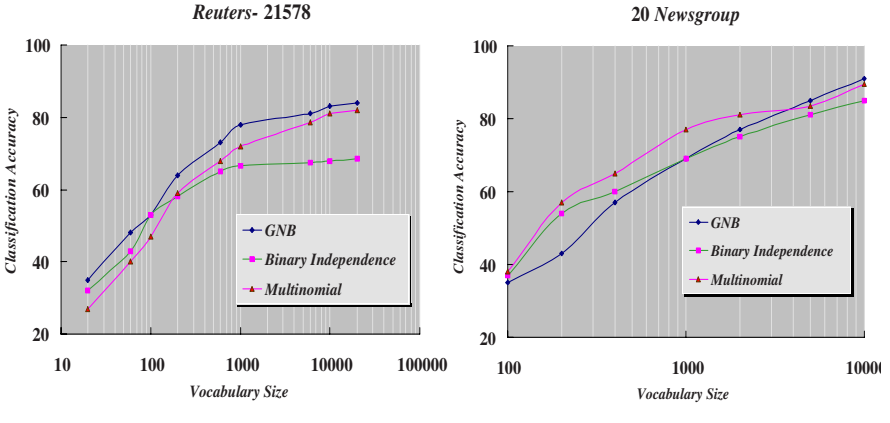


Fig. 1. The Experimental Result on Dataset **Reuters-21578 and 20 Newsgroup**

We conducted similar experiment on another dataset: 20-Newsgroups collection. we can also see similar improvement. Richer vocabulary and semantics, which can give more positive information for classification, may be one reason. On the other hand, the binary independence model and multinomial model represent only one aspect of a given word. But GNB can integrate them into one model on text documents, this should be the main reason.

5 Conclusions

In this paper, we presented a novel web mining approach named General Naive Bayes (GNB). By defining Projective Information Gain $PI(C; V_i)$, GNB can easily decide which representation is appropriate for a given word, thus overcome the restrictiveness of the binary independence model and multinomial model. Furthermore, the classification rule applied by GNB can directly handle both representations concurrently while not suffering from the user's bias. Our results indicate that GNB constitutes a promising addition to the induction algorithms from the viewpoint of classification performance.

References

1. Sahami M, Dumais S, Heckerman D, Horvitz E. A Bayesian approach to filtering junk e-mail. In Proceedings of the AAAI Workshop, 55–62, 1998.
2. Androutsopoulos I, Paliouras G. Learning to filter spam e-mail: A comparison of a Naive Bayesian and a memory-based approach. In Proceedings of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, 1–13, 2000.
3. Pazzani M, Billsus D. Learning and revising user profiles: The identification of interesting web sites. Machine Learning, Vol 27, 313–331, 1997.
4. Ricardo V, Irina R. A Decomposition of Classes via Clustering to Explain and Improve Naive Bayes. Lecture Notes in Computer Science, Vol 2837, 444–455, 2003.
5. Liangxiao J, Harry Z, Zhihua C. Dynamic K-Nearest-Neighbor Naive Bayes with Attribute Weighted. Lecture Notes in Computer Science, Vol 4223, 365–368, 2006.
6. Karl-Michael S. On Word Frequency Information and Negative Evidence in Naive Bayes Text Classification. Lecture Notes in Computer Science, Vol 3230, 474–485, 2004.
7. T Kalt, W B Croft. A new probabilistic model of text classification and retrieval. Technical Report IR-78, University of Massachusetts Center for Intelligent Information Retrieval, 1996.
8. Shi Z. Semi-supervised model-based document clustering: A comparative study. Machine Learning, Vol 65, 3–29, 1998.
9. http://people.csail.mit.edu/people/jrennie/20Newsgroups/
10. http://www.daviddlewis.com/resources/testcollections/reuters21578/
11. Daphne K, Mehran S. Hierarchically classifying documents using very few words. In Proceedings of the 14th International Conference on Machine Learning, 329–387, 1997.
12. Mehran S. Learning limited dependence Bayesian classifiers. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 335–338, 1996.

Feature Extraction and Classification of Tumor Based on Wavelet Package and Support Vector Machines

Shulin Wang[1,2], Ji Wang[1], Huowang Chen[1], and Shutao Li[3]

[1] School of Computer Science, National University of Defense Technology, Changsha, Hunan 410073, China
jt_slwang@hnu.cn
[2] School of Computer and Communication, Hunan University, Changsha, Hunan 410082, China
[3] College of Electrical and Information Engineering, Hunan University, Changsha, Hunan 410082, China

Abstract. DNA microarray experiments provide us with huge amount of gene expression data, which leads to a dimensional disaster for extracting features related to tumor. A wavelet package decomposition based feature extraction method for tumor classification was proposed, by which eigenvectors are extracted from gene expression profiles and used as the input of support vector machines classifier. Two well-known datasets are examined using the novel feature extraction method and support vector machines. Experiment results show that the 4-fold cross-validated accuracy of 100% is obtained for the leukemia dataset and 93.55% for the colon dataset.

Keywords: gene expression profiles; tumor classification; feature extraction; support vector machines; wavelet package decomposition.

1 Introduction

High throughput gene expression techniques make it possible to measure the expression levels of thousands of genes simultaneously, so it is usually an effective tool for tumor research, because many or even all human diseases may be accompanied by specific changes in the expression levels of some genes. However, due to its characteristics such as high dimensionality and small sample size, how to select tumor-related genes and to extract integrated features to drastically reduce the dimensionality of gene expression data constitutes a challenging analytical problem.

Many researchers have been studying many problems of tumor classification based on gene expression profiles. For example, unsupervised methods such as clustering [1] and self-organizing maps [2] and supervised methods such as artificial neural networks [3], support vector machines (SVM) [4, 5] and multi-layer perceptrons [6] have been successfully applied to classify tumor tissues. However, feature extraction plays a key role in the problem of tumor classification without doubt.

The goal of feature extraction is to eliminate redundancies in gene expression profiles to obtain the integrated attributes which can correctly classify the tumor

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 871–878, 2007.

dataset. For example, Xuewu Zhang *et al* [7] applied independent component analysis (ICA) to extract independent components (ICs) from gene expression data to be used as classification information. Wang X.H. *et al* [8] use stationary wavelet transform to denoise the microarray image before further image processing. In this paper we propose a novel feature extraction method which combines gene ranking and wavelet package decomposition (WPD) to extract tumor-related features which are used as the input of SVM classifier.

2 The Tumor Classification Model

2.1 Representation of Gene Expression Profiles

Let $G = \{g_1, \cdots g_n\}$ be a set of genes and $S = \{s_1, \cdots s_m\}$ be a set of samples. The corresponding gene expression profiles can be represented as matrix $X = (x_{i,j})$, $1 \le i \le m$, $1 \le j \le n$, where $x_{i,j}$ is the expression level of sample s_i on gene g_j, and usually $n >> m$.

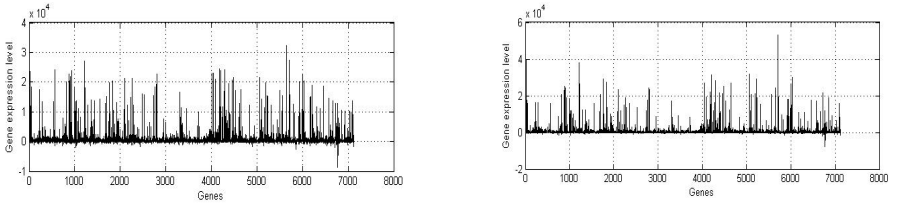


Fig. 1. The gene expression levels of two leukemia samples: ALL(left) and AML(right) sample

The matrix X is composed of m row vectors $s_i \in R^n$. Each vector s_i can be viewed as a signal, so we can apply the signal processing method to treat gene expression data. The expression levels of two samples selected randomly in leukemia dataset are shown in Fig. 1 in which one is an ALL sample and another is an AML sample.

2.2 The Algorithm Model

Our task is to classify all samples into two types, which is a binary classification problem. To achieve this goal, the framework of classification algorithm is designed as follows, and then the detailed descriptions and rationale for each step are introduced in the following sections.

Step 1. Gene selection: for each gene $g_i \in G$, we firstly calculate its score according to the feature score criterion (FSC) [9] and the revised feature score criterion (RFSC) [10], and then rank all genes in their scores. After gene ranking, we simply take the top-ranked genes with the highest scores as the selected gene subset G_{top}, usually satisfying $|G_{top}| \ll |G|$.

Step 2. Applying three-layer WPD to the top-ranked gene subset G_{top} for every sample to extract its eigenvectors.

Step 3. Splitting the obtained eigenvector set into training set and testing set, and then training SVM classifier using training set to get a classification model for tumor classification.
Step 4. Testing the obtained classification model using testing set to get the predictive accuracy of tumor classification.

2.3 Gene Selection (Step 1)

Gene selection is necessary for performing the tumor classification with gene expression profiles. In measuring the classification information of genes, Golub et al [9] proposed FSC as gene selection method. For each gene $g_i \in G$, the FSC method 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, and then calculate its score with the FSC formula $FSC(g_i) = \left|(\mu_i^+ - \mu_i^-)/(\sigma_i^+ + \sigma_i^-)\right|$. All genes are ranked in their score values. However, when the mean values of gene g_i in normal and tumor tissues are equal, there is a fault in this formula because this gene g_i is removed as noise from informative gene subset due to $F(g_i) = 0$, so RFSC which is the revised FSC was proposed [10]. RFSC consists of two parts. The first part is the original FSC formula, and the second part concerns the classification performance of variance.

$$RFSC(g_i) = 0.5\left|(\mu_i^+ - \mu_i^-)/(\sigma_i^+ + \sigma_i^-)\right| + 0.5\ln((\sigma_i^{+2} + \sigma_i^{-2})/(2\sigma_i^+\sigma_i^-)) \quad (1)$$

2.4 Wavelet Package Decomposition (Step 2) [11]

Wavelet package decomposition decomposes not only low frequency, but also high frequency. Therefore, it is the more precise method than wavelet decomposition. As it does, wavelet package analysis is a more widespread wavelet method. It applies to various signals including signal decomposition. The procedure of the eigenvector extraction of a sample is described as follows.

Step 2.1. Sample viewed as signal is decomposed into three-layer wavelet package to obtain eight signal characteristics of frequency composition from low frequency to high frequency in the third layer. The decomposition structure is shown as Fig. 2 in which node (*i*,*j*) denotes the *j*-th node in the *i*-th layer, where *i*=0,1,2,3 and *j*=0,1,…,7.
Step 2.2. Wavelet package decomposition coefficients are reconstructed to obtain signals of various frequency scopes. S_{30} denotes reconstruction signal of X_{30} (corresponding to node (3,0)). S_{31} denotes reconstruction signal of X_{31} (node (3,1)). Others are deduced similarly. If all nodes in the third layer are analyzed, the overall signal S can be expressed as follows: $S = S_{30} + S_{31} + S_{32} + S_{33} + S_{34} + S_{35} + S_{36} + S_{37}$.
Step 2.3. Calculating the overall energy of all frequency signals, and then constructing eigenvectors for each sample. Let E_{3j} be energy corresponding to S_{3j}, then $E_{3j} = \int \left|S_{3j}\right|^2 dt = \sum_{k=1}^{n} \left|x_{jk}\right|^2$, where x_{jk} denotes range value of S_{3j} and $j = 0,1,\cdots,7$. An eigenvector F can be constructed by means of energy as elements and shown as follows: $F = [E_{30}, E_{31}, E_{32}, E_{33}, E_{34}, E_{35}, E_{36}, E_{37}]$.

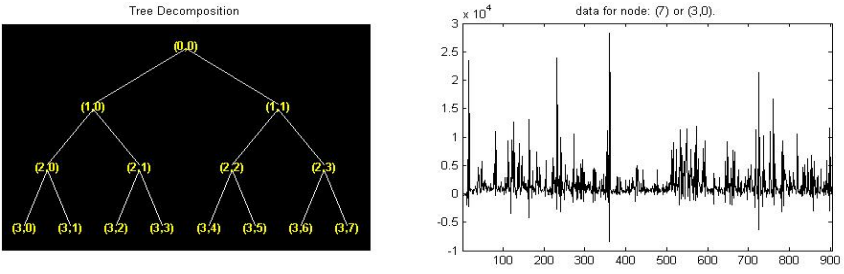


Fig. 2. Three-layer WPD structure and the lowest frequency signal of a leukemia sample

2.5 Support Vector Machines (Step 3) [12]

SVM is a relatively new type of statistic learning theory, originally introduced by Vapnik. SVM builds up a hyper-plane as the decision surface to maximize the margin of separation between positive and negative samples. Given a labeled set of m training samples $S = \{(F_i, y_i) \mid (F_i, y_i) \in R^8 \times \{\pm 1\}, i = 1,2,\cdots m\}$, where $F_i \in R^8, y_i \in \{\pm 1\}$ is a label of sample eigenvector F_i, and the discriminant hyper-plane is defined by formula (2).

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i K(F_i, x) + b \tag{2}$$

where $K(F_i, x)$ is a kernel function and the sign of $f(x)$ determines which class the unknown sample eigenvector x belongs to. Constructing an optimal hyper-plane is equivalent to finding all the support vectors α_i and a bias b.

3 Experiments

3.1 The Descriptions of Two Tumor Datasets

We have experimented with two well-known datasets: the leukemia dataset [9] and the colon dataset [13]. The descriptions of the two datasets are shown in Table 1.

Table 1. Descriptions of two tumor datasets in our experiments

Tumor Dataset	#Gene	#Sample	Subtype 1	Subtype 2
Leukemia Dataset	7,129	72	47(ALL)	25(AML)
Colon Dataset	2,000	62	40(Tumor)	22(Normal)

3.2 Experiment Methods

For our tumor classification algorithm, extracting eigenvectors from gene expression data using WPD is implemented in MATLAB 7.1, and the corresponding source code is shown in Fig. 3.

```
MG=load('LeukemiaDataset.txt');        %load gene expression profiles from the text file
s=zeros(72,8);                %initialize eigenvector for every leukemia sample
for i=1:72;                   %compute eigenvector for every sample in leukemia dataset
   %decompose a sample with wavelet db9 using three-layer wavelet package
   t=wpdec(MG(i,:),3,'db9','shannon');
   s130=wprcoef(t,[3,0]); s131=wprcoef(t,[3,1]); %s130 denotes the reconstruction coefficient
   s132=wprcoef(t,[3,2]); s133=wprcoef(t,[3,3]);
    s134=wprcoef(t,[3,4]); s135=wprcoef(t,[3,5]); s136=wprcoef(t,[3,6]); s137=wprcoef(t,[3,7]);
   s(i,1)=norm(s130); s(i,2)=norm(s131);   %compute variance of reconstruction coefficient
   s(i,3)=norm(s132); s(i,4)=norm(s133); s(i,5)=norm(s134); s(i,6)=norm(s135);
   s(i,7)=norm(s136); s(i,8)=norm(s137);
end          %s(i,:) denotes the eigenvector of the i-th sample
save('WaveletFeature.txt','s','-ascii','-double'); %save eigenvector of all sample into the file
```

Fig. 3. MATLAB source code for extracting eigenvector from gene expression profiles

We firstly apply FSC and RFSC to roughly select the top-ranked gene subset, respectively, and then apply the source code in Fig. 3 to the selected gene subset to extract eigenvectors which are used as the input of the SVM software LIBSVM [14] to classify the two tumor datasets. Training SVM requires specifying the type of kernel and the regularization parameter C. Generally, the recommended kernel for nonlinear problems is the Gaussian radial basis kernel $K(x, y) = \exp(-\gamma\|x - y\|^2)$ that is also used in our experiments. However, finding the best combination for the parameter pair (C, γ) can be challenging when applied to real datasets. Notice that given the input data used by SVM are already normalized. Finally, the 4-fold cross-validated (CV) accuracy can be used to measure the classification performance of SVM classifier.

3.3 Experiment Results

The CV accuracy of the same dataset is sensitive to different wavelet package and different gene selection methods. Experiments show that FSC is better than RFSC to the leukemia dataset in our classification method and in contrast RFSC is better than FSC to the colon dataset, which is obviously validated in Fig. 4. Table 2 partly shows

Table 2. Comparison of the classification using different wavelets and 200 top-ranked genes

Dataset(method)	Wavelet	CV Acc.	Wavelet	CV Acc.	Wavelet	CV Acc.
Leukemia (FSC+WPD+SVM)	db1	95.83%	db8	98.61%	db15	94.44%
	db2	94.44%	db9	98.611%	db16	94.44%
	db3	97.22%	db10	97.22%	db17	97.22%
	db4	95.83%	db11	94.44%	db18	95.83%
	db5	94.44%	db12	93.06%	bior1.1	95.83%
	db6	93.06%	db13	93.06%	bior2.8	95.83%
	db7	94.44%	db14	93.06%	rbio1.1	95.83%
Colon (RFSC+WPD+SVM)	db1	79.03%	db8	87.1%	db15	80.65%
	db2	90.32%	db9	91.94%	db16	80.65%
	db3	85.48%	db10	88.71%	db17	83.87%
	db4	80.65%	db11	85.48%	db18	85.48%
	db5	83.87%	db12	85.48%	bior1.1	83.87%
	db6	80.65%	db13	85.48%	bior2.8	91.94%
	db7	85.48%	db14	82.26%	rbio1.1	83.87%

the classification accuracy for the leukemia and colon datasets using different wavelet and 200 top-ranked genes which are selected with FSC to the leukemia dataset and RFSC to the colon dataset, respectively. It is obvious that the classification accuracy is different to different wavelets. From Table 2 we can conclude that wavelet db9 is relatively suitable for the two datasets simultaneously.

The 4-fold CV accuracy of 94.44% can be achieved when extracting eigenvectors from raw leukemia samples using wavelet db9 without gene ranking, and correspondingly 82.26% for raw colon samples. Why the CV accuracy is not enough high results from gene redundancies. Fig. 4 shows the 4-fold CV accuracy of tumor classification when using wavelet db9 from 20 to 400 top-ranked genes. From Fig. 4 we can see that the highest CV accuracy of 100% is achieved when using 375 top-ranked genes for the leukemia dataset and 93.55% when using 35 top-ranked genes for the colon dataset.

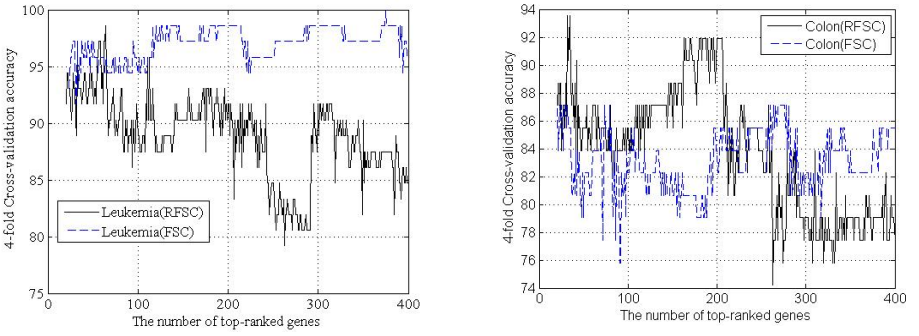


Fig. 4. Comparison of tumor classification using two gene ranking methods

Table 3. The classification accuracy comparison among different classification approaches on the colon dataset and the leukemia dataset

Feature Extraction	Classifier	Dataset	CV Acc.	Reference
Signal to noise ratio	SVM	Colon	90.30%	[15]
		Leukemia	94.10%	
Genetic Algorithm (GA)	k-nearest neighbor (k-NN)	Colon	94.10%	[16]
		Leukemia	84.60%	
All genes, TNoM score	SVM with quadratic kernel	Colon	74.20%	[17]
		Leukemia	94.40%	
Principal component analysis (PCA)	Logistic discriminent	Colon	87.10%	[18]
		Leukemia	94.20%	
Partial least square	Logistic discriminant	Colon	93.50%	[18]
		Leukemia	95.90%	
	Quadratic discriminant analysis	Colon	91.90%	[18]
		Leukemia	96.40%	
Independent component analysis (ICA)	Calculating the ratio of tumor and normal ICs	Colon	91.90%	[7]
RFSC and WPD	SVM with RBF kernel	Colon	93.55%	This paper
FSC and WPD		Leukemia	100%	

3.4 Comparison of the Classification Accuracy

Many feature extraction and machine learning approaches have been successfully applied to the tumor classification based on gene expression data. Comparison of the classification accuracy among different tumor classification approaches are shown in Table 3 from which we can conclude that our approach obtains almost the best results and the validity of our approach is also evaluated. In fact, among the published tumor datasets, classifying the colon dataset is more difficulty than doing others.

4 Conclusions

A novel feature extraction method to extract eigenvectors from high dimensional gene expression profiles using three-layer WPD is proposed, which deal with gene expression profiles from the view of signal processing. Our aim is to explore the feasibility of WPD in tumor classification based on gene expression profiles. Two well-known tumor datasets are examined to assess the classification performance, and the experiment results show that the 4-fold CV accuracy of 100% is obtained for the leukemia dataset and 93.55% for the colon dataset. Experiments prove that our method can meet real-time application requirements in clinical domain because the feature extraction algorithm is computationally efficient and the eigenvector of a sample from patient can be extracted independently.

Acknowledgement

This research is supported by the Program for New Century Excellent Talents in University and the Excellent Youth Foundation of Hunan Province (06JJ1010).

References

1. Heping Zhang, Chang-Yung Yu, Burton Singer, and Momiao Xiong. Recursive partitioning for tumor classification with gene expression microarray data. PNAS, 2001, 98(12):6730-6735.
2. David G. Covell, Anders Wallqvist, Alfred A. Rabow, and Narmada Thanki. Molecular classification of cancer: unsupervised self-organizing map analysis of gene expression microarray data. Molecular Cancer Therapeutics, 2003, 2:317-332.
3. Khan J., Wei J.S., Rignér M., Saal L.H., Ladanyi M., Westermann F., Berthold F., Schwab M., Antonescu C.R., Peterson C., and Meltzer P.S.. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine, 2001, 7(6):673-679.
4. Guyon I., Weston J., Barnhill S., and Vapnik V.. Gene selection for cancer classification using support vector machines. Machine Learning, 2002, 46:389-422.
5. 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.
6. Peter Antal, Geert Fannes, Dirk Timmerman, Yves Moreau, Bart De Moor. Bayesian applications of belief networks and multilayer perceptrons for overian tumor classification with rejection. Artificial Intelligence in Medicine, 2003, 29(1):39-60.

7. Xuewu Zhang, Yee Leng Yap, Dong Wei, Feng Chen, and Antoine Danchin. Molecular diagnosis of human cancer type by gene expression profiles and independent component analysis. European Journal of Human Genetics, 2005, 05(9):1018-4813.
8. Wang X.H., Robert S.H. Istepanian, and Yong Hua Song. Microarray image enhancement by denoising using stationary wavelet transform. IEEE Transactions on Nanobioscience, December 2003, 2(4):184-189.
9. 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.
10. Yingxin Li and Xiaogang Ruan. Feature selection for cancer classification based on Support Vector Machine. Journal of Computer Research and Development, 2005, 42(10):1796-1801.
11. Tao Li, Qi Li, Shenghuo Zhu, and Mitsunori Ogihara. A survey on wavelet applications in Data Mining. SIGKDD Explorations, 2003, 4(2):49-48.
12. Vapnik V.N.. Statistical learning theory. Springer, New York, 1998.
13. 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.
14. 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.
15. Furey T.S., Cristianini N., Duffy N., Bednarski D.W., Schummer M., and Haussler D.. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics, 2000, 16(10):906-914.
16. 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.
17. Ben-Dor A., Bruhn L., Friedman N., Nachman I., Schummer M., and Yakhini N.. Tissue classification with gene expression profiles. Journal of Computional Biology, 2000, 7:559-584.
18. Nguyen D.V. and Rocke D.M.. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics, 2002, 18(1):39-45.
19. 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.
20. Yuhang Wang, Fillia S. Makedon, James C. Ford, and Justin Pearlman. HykGene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. Bioinformatics, 2005, 21(8):1530-1537.

Resource Allocation and Scheduling Problem Based on Genetic Algorithm and Ant Colony Optimization

Su Wang and Bo Meng

Computer School, Wuhan University, Wuhan, Hubei Province, China
zzwangsu@163.com, bmengwhu@sina.com

Abstract. Faced with the increasing growth of container throughput and more large ships in shorter time, a key factor of success is to generate the best resource allocation plan for the future. This paper discusses a heuristic GA-ACO method which combines Genetic Algorithm and Ant Colony Optimization for resource allocation and scheduling problem in container terminals. In the first phase GA uses character string to represent chromosome for allocation plans and finds the best allocation by self-learning. In the second phase, an improved ACO algorithm is introduced to optimize the scheduling jobs based on the allocation plan from GA. We examine the performance of tugboat allocation optimization in container terminals and obtain satisfactory results.

Keywords: Resource Allocation, Scheduling, Genetic Algorithm, Ant Colony Optimization, Container Terminal.

1 Introduction

Since containers have been introduced to the world trade over thirty years, the container transportation becomes the most important worldwide transportation. An increasing number of goods are put into containers, loaded and unloaded onto large ships to their respective destinations throughout the world. Container terminals are continuously facing the increasing growth of container throughput and more large ships in shorter time. This leads to the necessity of scheduling highly expensive terminal equipments as efficiently as possible. A key factor of success is to allocate these resources at the optimal level.

Many researchers have developed allocation optimization approaches for container terminal logistics. Chan in [1] and Etsuko Nishimura in [2] adopted GA to determine a dynamic berth allocation in the public berth system. During the process of berth allocation, the berthing time and berthing position of a containership are determined. W.C.Ng in [3] discussed quay crane allocation problem and developed a dynamic programming-based heuristic to solve the scheduling problem.

This paper discusses the allocation of another important resource tugboat. There are limited studies on tugboat allocation problem. Liu Zhixiong in [4] simulated tugboat operation, but didn't optimize this problem. In this paper, we develop a new mixed heuristic algorithm GA-ACO combined with Genetic Algorithm and Ant Colony Optimization Algorithm. Our aim is to employ the satisfactory allocation plan

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 879–886, 2007.

for the future from the former records, which would serve the increasing throughput and improve the productivity of container terminals.

The rest of paper is organized as follows: in section 2, we present the tugboat allocation problem and illustrate why GA and ACO are used in this problem. In section 3, GA-ACO algorithm for allocation optimization is discussed. In section 4, we describe the experiment and analyze the results. Finally, we make conclusions and introduce the future work in section 5.

2 Allocation Problem

There are many factors such as the arriving time of vessels and the number of containers to be handled influencing the allocation decisions. When ships arrive, they can not enter into the berth directly and need to be tugged by the tugboats. Moreover, the moving and leaving of ships also need to be tugged. Tugboat operation is a stochastic, dynamic and discrete process because the arriving time of vessels is stochastic and discrete. Vessels with different sizes and types should schedule tugboats with different horsepower, avoiding big tugboat for small vessel and small tugboat for big vessel. If the number of tugboat allocation is low, the arriving vessels can not enter into the berth for operation in time so that long waiting time and waiting queue decrease the efficiency of ports. On the contrary, if the number of tugboat allocation is high, it will cause the low utilization rate and the waste of money because some tugboats leave unused. Proper allocation of tugboats can improve operation efficiency and economic benefits of container terminals.

2.1 Allocation Model

The allocation optimization process includes three models shown in Fig.1. The allocation of tugboat is influenced by the investment money, arriving vessels and other constraints. They are input into the allocation optimization module, which is responsible for generating allocation solutions and finding the best by Genetic Algorithm. In this process, the determination of fitness is based on three parameters: average utilization rate of tugboat, average waiting time of vessels and maximum length of waiting queue. Because of the discrete and dynamic characteristics of tugboat operation, it is difficult to formulate a mathematic function of fitness. These parameters must be attained by simulation. But only simulation technology can not get optimal scheduling results, the optimization method would be used to gain better solution. We have already studied the effectiveness of Ant Colony Optimization for scheduling problem in container terminals in [5]. Hence, ACO is applied to optimize the simulation of tugboat operation and outputs the three parameters for generating the fitness evaluation. The evaluation module uses Bayes Net method described in [6] to calculate the fitness for GA.

2.2 Reasons for GA in Allocation Model and ACO in Scheduling Model

Meta-heuristic methods are explored to resolve this complex allocation problem. Compared with other algorithms, Genetic Algorithm is robust, self-learning and can

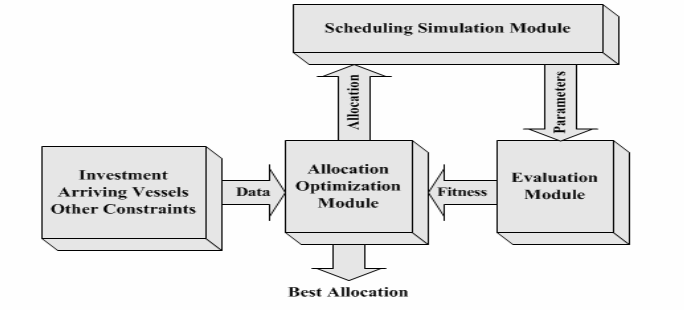


Fig. 1. Tugboat Allocation Simulation And Optimization Model

get preferable solutions. Moreover, the most important is tugboat allocation plan can be easily represented by chromosome and we can get satisfactory solution through self-learning mechanism. So GA is adopted for allocation model.

ACO was first proposed for combinatorial optimization. The basic idea is inspired by the way ants explore the environments in search of food . It is expanded to resolve scheduling problem in [7], [8]. Jobs are defined as ants and resources are defined as nodes. We consider that the nodes and ants can change according to the different numbers of resources and jobs. So ACO is suitable for dynamic tugboat scheduling instead of GA, in which the changeable chromosome is difficultly represented.

GA-ACO Algorithm is described as Fig.2.

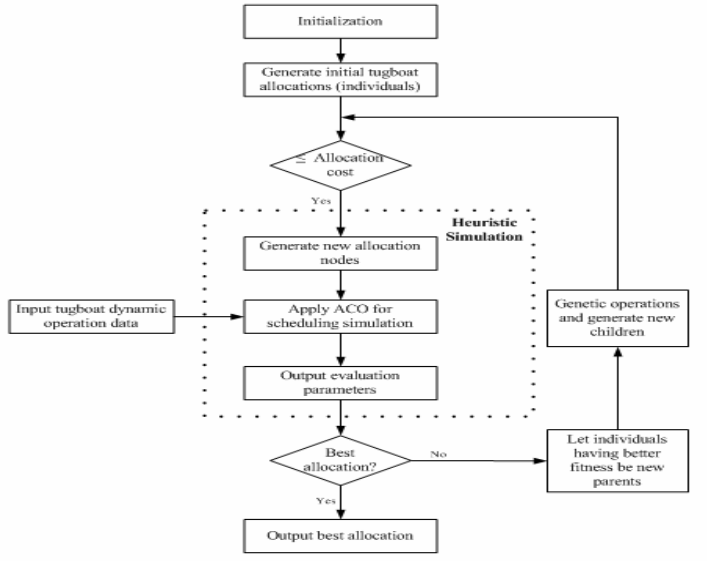


Fig. 2. GA-ACO Algorithm

3 GA-ACO for Allocation Optimization

3.1 Genetic Algorithm

In GA application, instead of using the classical binary bit string representation, the chromosomes are represented as character strings. The length of string represents how many types of tugboats and each character is the number of tugboats with different horsepower. We give one tugboat allocation of six types in Fig.3 including 1 tugboat with 1200ps, 6 tugboats with 2600ps, 2 tugboats with 3200ps, 3 tugboats with 3400ps and 2 tugboats with 4000ps.

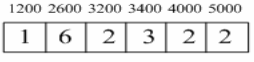


Fig. 3. Chromosome Representation

The underlying fundamental mechanism of GA consists of three main operations:

(1) Reproduction: Reproduction is a process in which individual chromosomes are copied according to their fitness values. The chromosomes with a higher fitness value would have more copies in the next generation.

(2) Crossover: Crossover is performed to introduce new chromosomes by recombining current genes. We employ 2-point crossover operator shown in Fig. 4. First, two cutting sites i and j are randomly selected in parents, in this example i=2 and j=4. Then, the substring between i and j of A is interchanged with B so that two children are formed.

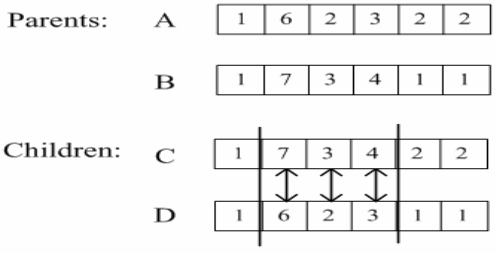


Fig. 4. Example of Crossover

(3) Mutation: Mutation introduces random changes to the chromosomes by altering the value to gene. In Fig.5, the third character changes from 2 to 5.

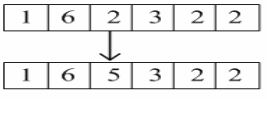


Fig. 5. Example of Mutation

We describe the flowchart of GA in Fig.6.

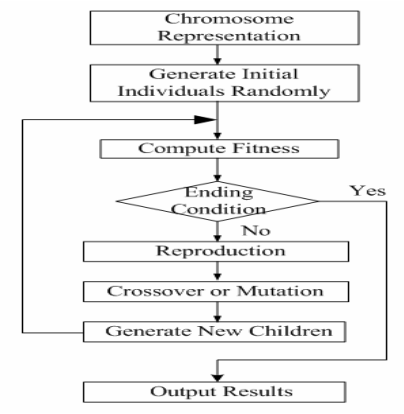


Fig. 6. Genetic Algorithm Flowchart

3.2 ACO Algorithm

As scheduling optimization method, each ant is assigned to a job T_i and each node represents one tugboat R_j. *Ti* schedules tugboats, then ant k_i deposits pheromone on the path. $\tau_{ij(t)}$ is the pheromone of path l_{ij}, which represents the cost of scheduling R_j after R_i.

Different from basic ACO, we introduce a new path selecting method to avoid local optima. We consider that ants as a simply animal has its own consciousness. At the beginning of selecting process, the pheromones on paths are few and have little influence on selecting paths so that ants move at the individual level, selecting path randomly. With the pheromone accumulating, the communication among ants becomes more and more. Then ants move at the collective level and they will select the path with more pheromone. According to this feature, we introduce the concept of Pheromone Influence (*PI*). This method can add the path diversity of selecting and avoid fast convergence into the local best.

The formulation of *PI*: n is the number of ants; ($L_1, L_2, \ldots, L_m$) is m solutions found by ants randomly and for scheduling problem L_i is the cost of scheduling solution; λ is the integer parameter usually 3 or 4.

$$PI = \frac{\sum_{i=1}^{m} (1 / L_i)}{m} \times \lambda \tag{1}$$

The node transition rule is as follows: when the average pheromone of all the paths is fewer than *PI*, ants select path randomly; otherwise, ants select path by pheromone. An ant k in R_i chooses R_j to move to following the rule:

$$s = \begin{cases} \arg \max_{j \notin tabu_k} \left\{ \tau_{ij}^{\alpha} \bullet \eta_{ij}^{\beta} \right\} & q \leqslant q_0 \\ S & q > q_0 \end{cases} \tag{2}$$

Where q is a random variable distributed between [0,1], q_0 is a tuneable parameter between [0,1]. For the selection of a resource the ant uses heuristic information as

well as pheromone information. The pheromone information is denoted by τ_{ij} and the heuristic information is denoted by η_{ij}. The heuristic value is defined as tugboat ability generated from the container terminal operations, which has been discussed in [5]. With probability q_0, the ant chooses R_j from the set of tugboats that have not been scheduled so far which maximizes $\tau^{\alpha}_{ij} \cdot \eta^{\beta}_{ij}$, where α and β are constants that determine the relative influence of the pheromone values and the heuristic values on the decision of the ant. Otherwise the next tugboat is chosen according to the probability over S determined by p^{k}_{ij}.

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \bullet \eta_{ij}^{\beta}}{\sum_{u \notin tabu_k} \tau_{iu}^{\alpha} \bullet \eta_{iu}^{\beta}} & j \square allow\ k \\ 0 & otherwise \end{cases} \tag{3}$$

In ACO, all ants are allowed to deposit pheromone after completing their tours. The updating rule is as follows:

$$\tau_{ij}^{new} = \rho \tau_{ij}^{old} + (1-\rho)\Delta\tau_{ij} \tag{4}$$

$$\Delta\tau_{ij} = \begin{cases} \dfrac{Q_{min}}{Q_k} & ant\ k\ select\ node\ i\ and\ node\ j \\ 0 & otherwise \end{cases} \tag{5}$$

Where (i,j) is the best tour, ρ is a parameter governing pheromone decay. The reason for ρ is that old pheromone should not have a too strong influence on the future. Q_{min} is the minimal cost for task k, Q_k is the real cost of completing task k.

The algorithm stops when some stopping criterion is met, e.g. a certain number of generations have been done or the best found solution has not changed for several generations.

4 Experiment Results

In this section we describe a scenario of tugboat allocation problem at a Chinese container terminal. The old tugboat allocation is shown in Table.1. Because of the increasing throughput and more and more large ships, it is necessary to increase the number of present tugboats and add up new tugboat with 5000 horsepower.

Table 1. The old distribution of tugboat

Horsepower(PS)	1200	2600	3200	3400	4000
Port 1	1	5	2	1	2

We simulate about 2800 records of arriving vessels in the year of 2003. There are two constraints: (1) the available number of every type of tugboat is between [1,10];

(2) the maximum investment money.

The parameters in GA-ACO Algorithm can be determined by experiments. For GA, the size of individual colony is 6, crossover rate is 0.8, mutation rate is 0.1 and the max generation (NC_{max}) is 100. For ACO, the number of ants is changed according to the number of jobs. Other parameters are: $\tau_0 = 0.5$, $\alpha = 1$, $\beta = 3$, $\rho = 0.8$, $NCmax$=1000.

The allocation fitness of generation 1, 20, 40 and 60 are given in Table.2. The evaluation results of scheduling of some allocations are shown in Table.3.

Table 2. Allocation results of generations

Initial Genes	F(0)	1	F(1)	20	F(20)	40	F(40)	60	F(60)
163342	0.924	162332	0.883	172322	0.872	163352	0.962	172342	0.879
152421	0.852	155213	0.625	182221	0.653	154222	0.573	163343	0.948
164222	0.876	163342	0.931	172223	0.894	163333	0.872	163352	0.967
165123	0.722	172253	0.957	163342	0.936	172342	0.886	163352	0.967
174211	0.673	152221	0.564	172342	0.862	173221	0.824	163352	0.967
172342	0.901	172323	0.869	163352	0.951	163343	0.931	172342	0.879

Table 3. Scheduling results of allocation

Allocation	Utilization Rate	Average Waiting Time	Max Waiting Queue
152421	36.2%	2.2 m	6 vessels
172322	42.8%	1.5 m	5 vessels
163333	43.6%	1.1m	5 vessels
172342	47.6%	0.9 m	4 vessels
163342	48.2%	0.7 m	4 vessels
163352	50.7%	0.6 m	4 vessels

The GA algorithm converges after about 80 generations. The best allocation is shown in Table.4. The comparison of the two allocations is shown in Table.5.

Table 4. The best allocation of tugboat

Horsepower(PS)	1200	2600	3200	3400	4000	5000
Port 1	1	6	3	3	5	2

Table 5. The comparison results of tugboat allocations

Allocation	Utilization Rate	Average Waiting Time	Max Waiting Queue
Old	20.3%	4.8 m	9 vessels
Best	50.7%	0.6 m	4 vessels

Compared with the old plan, the new allocation plan can improve the utilization rate of tugboat, reduce the waiting time and waiting queue of vessels. Using the new allocation, container terminal can improve the efficiency of equipment and serve more ships and larger ships in the future.

5 Conclusions

In this paper, we develop GA-ACO Algorithm for resource allocation problem in container terminals. GA is responsible for generating allocation plans and ACO is responsible for simulating the plans for scheduling. We examine the performance of optimization of tugboat allocation in container terminals and obtain satisfactory results.

In the future, we will explore how to apply this method for other equipments or resources in container terminals, such as quay cranes, trucks and so on. Moreover, we are prepared to improve this method deeply for the larger scale allocation problem and better performance.

Acknowledgments. This research was supported by China Scholarship Council and the Natural Science Foundation of Hubei Province(2006ABA218, 2001ABB058).

References

1. W.T.Chan, A.Imai: The Berth Allocation Problem: Heuristic Method Using Genetic Algorithms. Proceedings of the first JSPS-NUS Seminar on Integrated Engineering. (1996) 109-114.
2. Estuko Nishimura, Akio Imai, Stratos Papadimitriou: Berth Allocation Planning in the Public Berth System by GeneticAlgorithms. European Journal of Operational Research, Vol. 131. (2001) 282-292.
3. W.C.Ng, K.l.Mak: Yard Crane Scheduling in Port Container Terminals. Applied Mathematical Modeling, Vol.29. (2005) 263-276.
4. Liu Zhixiong, Wang Shaomei: The Computer Simulation Study of Port Tugboat Operation. Journal of System Simulation, Vol.16, No.1. (2004) 45-47.
5. Su Wang, Bo Meng: An Improved Ant Colony Optimization Algorithm for Tugboat Scheduling Planning in Container Terminals. The Fifth Wuhan International Conference on E-Business. (2006) 872-879.
6. Hu Wenbin: Research on the Key Technology of Distributed Intelligent Group Decision Supported System Based on Multi-agent. PHD dissertation. Wuhan University of Technology. (2004).
7. Christian Blum, Michael Sampels: An Ant Colony Optimization Algorithm for Shop Scheduling Problems. Journal of Mathematical Modelling and Algorithms, Vol.3. (2004) 285-308.
8. Chia Jim Tong, Hoong Chuin Lau, Andrew Lim: Ant Colony Optimization for the Ship Berthing Problem. The 5th Asian Computing Science Conference. Lecture Notes in Computer Science 1742. (1999) 359-370.

Image Classification and Segmentation for Densely Packed Aggregates

Weixing Wang

School of Computer Science & Technology, Henan Polytechnic University,
Post code: 454000, Henan, China
znn525d@yahoo.com

Abstract. This paper presents a methodology for delineating densely packed aggregate particles based on aggregate image classification. There is no earlier work on segmentation of aggregate particles has exploited these two building blocks for making robust object delineation. The proposed method has been tested experimentally for different kinds of densely packed aggregate images, which are difficult to detect by a normal edge detector. As tested, the studied algorithm can be applied into other applications too.

1 Introduction

The size, shape and texture of aggregates are very important characteristics of the physical properties for the geology research and aggregate production industry and mining industry. In mining, the size and shape distributions of fragments affect not only rock blasting, but also the whole mining production sequence. In the quarry manufacture, the size, shape and texture of aggregate must fit the requirements of customers, such as high-way and rail way construction companies, the different companies in the building industries, etc. In geology, the size, shape and texture of gravel and sedimentary deposits are often used for analyzing and describing local geological properties in a certain region. Hence, aggregate size, shape and texture are widely applied and studied in both industries and research organizations.

The traditional way of determining size distributions of aggregate material is by sieving. Sieving has some problems. Think of elongated particles that may pass a sieve size smaller than their intermediate diameter. According to the thesis work [1] image analysis gives better information on the true size (and shape!) grading of aggregate. In industry and the lab image analysis is increasingly used to measure size and shape of aggregate transported on container belts. The method is accurate, because many particles are being processed by the images (> 10.000 data are needed to obtain sufficient accuracy).

Image interpretation is especially helpful when dealing with large size aggregates or rock blocks, those are not always easy to weigh or sieve. Image analysis is currently used to assess rock aggregates by blasting. Therefore, this research subject becomes a hot topic in the world during last thirty years. Today, a number of image systems have been developed for measuring aggregates in different application environments (Fig. 1) such as aggregates on/in gravitational flows, conveyor belts,

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 887–894, 2007.

muckpiles, and laboratories. The research and development has been and is being carried out in many countries, the detailed information can be found in [2-5].

The common problem for the research and development is to delineate every rock aggregate, but in most industrial cases, rock aggregate images are difficult to segment due to rough surface, overlapping, and size variation etc. Hence, it is crucial to extract qualitative information about a rock aggregate image to characterize images before starting segmentation. In this paper, characterization of rock aggregate images have been thoroughly investigated by extensive tests on hundreds of images, using several packages of commercial software for image segmentation, and some previous image segmentation algorithms [6-10] coded by the author.

Image classification was essential in segmentation of rock aggregates. Therefore, it was started by developing procedures for crude determination of number of rock aggregates in an image, the basic idea being that "edge density" is a rough measure of average size in images of densely packed aggregates. This work is an essential component in the current segmentation process. The studied segmentation algorithm is based on grey value valleys which is a grey-value structure occurring more frequently than traditional step edges. However, without knowledge of scale (approximate size of rock aggregates in the image) such an approach would be hard to realize. In fact, this goes for any segmentation technique which normally "handles" the problem by adjustment of various smoothing parameters, thresholds etc. Since it needs an automatic image segmentation process to perform "image classification" first, and to avoid making "smoothing parameters" crucial for good results.

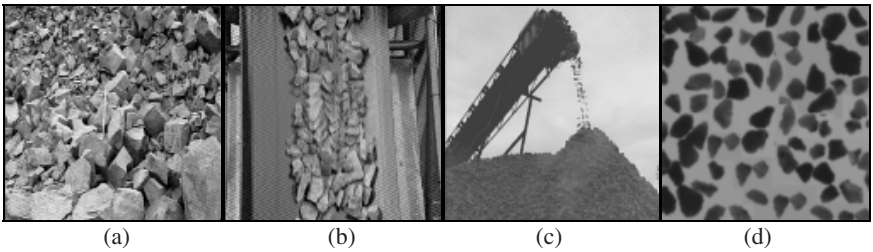


Fig. 1. Examples of aggregate images in engineering applications: (a) Muckpile; (b) Belt; (c) Falling flow; and (d) Road surface

2 Aggregate Image Classification

When an image analysis system is used for automatic monitoring of a fast moving conveyor belt, one important issue is the automatic grabbing of aggregate images. The quality of the image affects the result of the analysis. The automatic system should avoid interaction performed by an operator. Hence, when the system grabs one image frame, the system should judge if the image can be processed. If the image quality is poor, it is not possible or desirable to conduct analysis, and therefore the image should be omitted and the system should wait for the next image frame. So, the classification of aggregate images should be a first step of development.

Aggregate images taken from a fast moving conveyor belt vary so much that the quality of any two successive images are not the same, i.e. some images might include about 80% fine materials which is difficult to recognize by the system; some images consist of only a few rocks which is of less interest to analyze; some images are very dark or very light with a poor contrast of gray values, which may be due to the illumination condition suddenly changed, in which case the wrong image information are obtained, the result of image analysis will be affected seriously; some images are quite blurry, caused by raining or the increase of the speed of conveyor belt, which can also be difficult to processing, etc. For an inspection task, the classification should be done in real time, and complex and time consuming texture analysis can not be used. Thus we exclude features for texture segmentation.

If a frame of image is of a good quality, the edges of the aggregates are sharp, the contrast between the aggregate and background is acceptable and the number of the aggregates is in a certain level, In such a situation, the image is easy to process by the existing system.

Based on the above discussion, we define classes of images as follows:

Class 1: when contrast between edges and no-edges is lower than a specified value σ_0, the image might be a blurred image with a certain degree, the image belongs in class 1.

Class 2: when the density of edges is lower than a specified value λ,, and the image with a lower average gray-value, i.e. less than a specified value v_0. the image might include only few visual particles together with an empty conveyor belt, the image belongs in class 2.

Class 3: when the density of edges is lower than a specified value λ,, and the image with a average gray-value, i.e. greater than a specified value v_0. the image might include only few visual particles together with fine material, the image belongs in class 3.

Class 4: when contrast between edges and no-edges is greater than a specified value σ_0 and the density of edges is greater than a specified value λ, the image is an accepted aggregate image. There is certain relationship between the density of edges and average size of aggregate or number of aggregate in a image. The class 4 can be sub-classified based on the average size of aggregate in an image.

The question is how to determine the contrast and density, as usual, we see that gradient magnitude image includes some different noise, all these noises will affect the contrast and density, in order to minimum the noises and process in real time, image pre-processing has be applied.

3 Aggregate Image Segmentation

The most important, and the hard part of computer vision for aggregates, is segmentation. Segmentation can be divided into two steps, one is segmentation based on gray levels (called image binarization, sometimes) in which a gray level image is processed and converted into a binary image. Another is segmentation based on particle shapes

in a binary image, in which overlapping and touching particles will be split, and over-segmented particles will be merged based on some prior knowledge such as shape and size etc.

Segmentation algorithms for monochrome (gray level) images generally are based on one of two basic properties of gray-level values: similarity and discontinuity. The principal approaches in the first category are based on thresholding, region growing, and region splitting and merging. In the second category, the approach is to partition an image based on abrupt changes in gray level. The principal areas of interest within this category are detection of isolated points and detection of lines and edges in an image.

The choice of segmentation of aggregate images based on similarity or discontinuity of the gray-level values depends on both developed sub-algorithms and applications.

Aggregate images have their own characteristics compared to other particle images. Generally speaking, under the front-lighting illumination condition which is common case, aggregate images have the characteristics: (1) uneven background and foreground for which a simple thresholding algorithm cannot be applied to segment the images; (2) each aggregate particle may possess a textured surface and multiple faces, which often causes an over-segmentation problem; (3) particles overlapping each other, which hides parts of a particle, or causes breaks of the boundaries of particles; (4) touching particles forming a large cluster; (5) rain, snow, or much fine material making aggregate images clump together.

Aggregates may be densely packed or be separated mostly on a background. The former case is more difficult to process than the latter. As mentioned in Section 1.1, most systems for aggregate images were developed based on simple thresholding algorithms (some of them combined with morphological segmentation algorithm) and boundary detection algorithms. The segmentation algorithm designing is application (here, the type of aggregate images) dependent. In this section, I summarize my own segmentation approaches for aggregate images, they are: (1) an algorithm based on edge detection; (2) an algorithm based on region split-and-merge; (3) an adaptive thresholding algorithm; and (4) an algorithm for splitting touching particles in a binary image.

The whole segmentation procedure consists of the two parts in this study: image classification, and aggregate delineation. Since the delineation algorithm needs thin edges of aggregates, the classification algorithm first classify image into different classes, then based on the classification results, the procedure shrink the image into a certain scale size, provided for aggregate particle delineation. After the delineation, the delineated image is converted to the original image size, re-mapping the contours of aggregates. The rock aggregate image classification algorithm was developed for general-purpose of rock aggregate image segmentation. The algorithm evaluates image quality and produces image class labels, useful in subsequent image segmentation. Because of the large variation of rock aggregate patterns and quality, the image classification algorithm produces five different labels for the classes: (1) images in which most of the aggregates are of small size; (2) images in which most of the aggregates are of medium size; (3) images in which most of the aggregates are of relative large size; (4) images with mixed aggregates of different sizes; and (5) images with many void spaces. If most aggregates in an image are very small, the fine-detail information in the image is very important for image segmentation, and the segmentation algorithm

must avoid destroying the information. On the contrary, if aggregates are large, it is necessary to remove the detailed information on the rock aggregate surface, because it may cause image over-segmentation. If most aggregates are of relative large size (e.g. medium size), the segmentation algorithm should include a special image enhancement routine that can eliminate noise of rock aggregate surface, while keeping real edges from being destroyed. Since the delineation algorithm was developed for densely packed aggregates, the void spaces have to be removal.

Assume that P is surrounded by strong negative and positive differences in the diagonal directions: $\nabla_{45} < 0$, and $\Delta_{45} > 0$, $\nabla_{135} < 0$, and $\Delta_{135} > 0$, whereas, $\nabla_0 \approx 0$, and $\Delta_0 \geq 0$, $\nabla_{90} \approx 0$, and $\Delta_{90} \approx 0$. Where Δ are forward differences: $\Delta_{45} = f(i+1, j+1) - f(i, j)$, and ∇ are backward differences: $\nabla_{45} = f(i, j) - f(i-1, j-1)$, etc. for other directions. It uses $\max(\Delta_\alpha - \nabla_\alpha)$ as a measure of the strength of a valley point candidate. It should be noted that sampled grid coordinates are used, which are much more sparse than the pixel grid $0 \leq x \leq n$, $0 \leq y \leq m$. f is the original grey value image after weak smoothing.

What should be stressed about the valley edge detector are: (a) It uses four instead of two directions; (b) It studies value differences of well separated points: the sparse $i \pm 1$ corresponds to $x \pm L$ and $j \pm 1$ corresponds to $y \pm L$, where $L >> 1$, in our case, $3 \leq L \leq 7$. In applications, if there are closely packed particles of area > 400 pixels, images should be shrunk to be suitable for this choice of L. Section 3 deals with average size estimation, which can guide choice of L; (c) It is nonlinear: only the most valley-like directional response $(\Delta_\alpha - \nabla_\alpha)$ is used. By valley-like, it means $(\Delta_\alpha - \nabla_\alpha)$ value. To manage valley detection in cases of broader valleys, there is a slight modification whereby weighted averages of $(\Delta_\alpha - \nabla_\alpha)$ - expressions are used. $w_1\Delta_\alpha(P_B) + w_2\Delta_\alpha(P_A) - w_2\nabla_\alpha(P_B) - w_1\nabla_\alpha(P_A)$, where, P_A and P_B are two end points in a section. For example, $w_1 = 2$ and $w_2 = 3$ are in our experiments; (d) It is one-pass edge detection algorithm (Fig. 2); the detected image is a binary image, no need for further thresholding; (e) Since each edge point is detected through four different directions, hence in the local part, edge width is one pixel wide (if average particle area is greater than 200 pixels, a thinning operation follows boundary detection operation); and (f) It is not sensitive to illumination variations.

Without image classification, there is a substantial difficulty choosing an appropriate L, the spacing between sampled points. Let L refer to spacing in an image of given resolution, where the given resolution may be a down sampling of the original image resolution. Since the image classification described earlier leads to an automatic down-sampling, the choice of L is not critical.

After valley edge point detection, there are pieces of valley edges, and a valley edge tracing subroutine, filling gaps is needed (Some thinning is also needed). As a background process, there is a simple gray value thresholding sub-routine, which before classification creates a binary image with quite dark regions as the bellow-threshold class. If this dark space covers more than a certain percentage of the image, and has few holes, background is separated from aggregates by a Canny edge detector along the between-class boundaries.

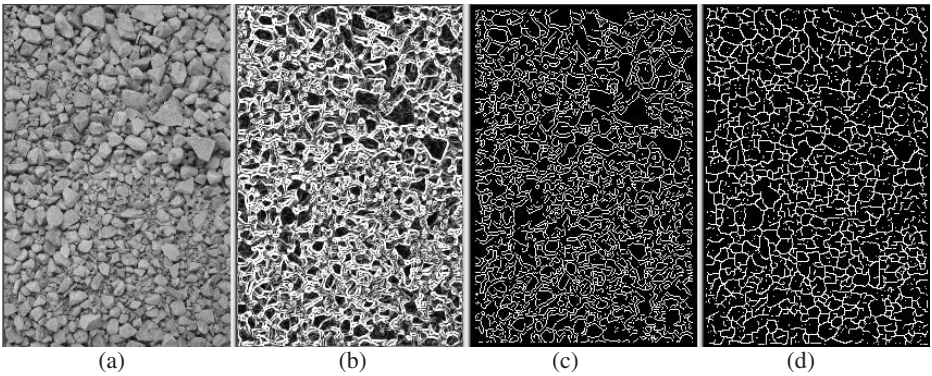


Fig. 2. Edge detection results: (a) original image, (b) Sobel edge detection result, (c) Canny edge detection result with a low threshold value, and (d) new algorithm detection result

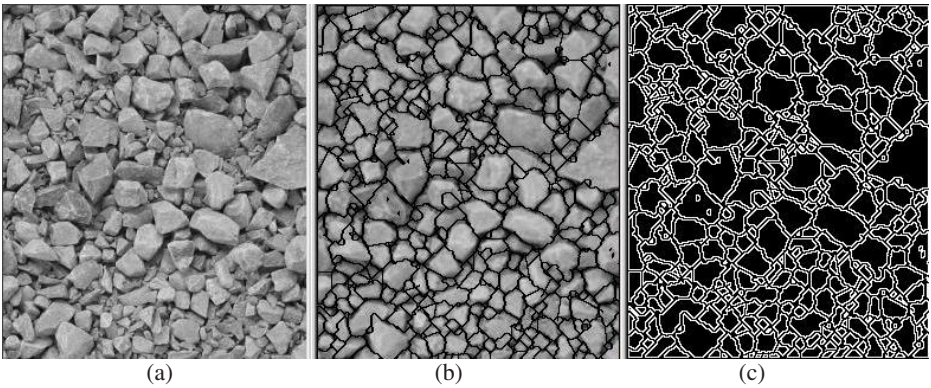
4 Experiments

To test the segmentation algorithm, we have taken a number of different aggregate particle images from a laboratory, a muckpile, and a moving conveyor belt. It is often that there is a lot of noise on the surface of fragments, which gives problems for image segmentation, over-segmentation and under-segmentation. Since surface noise and 3D geometry of rock fragments create step edges in most cases, and our new algorithm is studied based valley edge detection, it disregards step edges. Therefore it works not only for less surface noise image, and also works for the images of serious surface noise. Figs. 3-4 illustrated image segmentation results.

The image in Fig. 4(a) was taken from a laboratory with bad illumination; the surfaces of the fragments include a lot of texture and noise. The existing algorithms have been used for the segmentation of rock fragments; all of them produce over-segmentation. By using the new segmentation algorithm, this image is classified into the class of medium fragment size (class 2): the segmentation algorithm reduces the scale of image two times, then delineates fragments, finally uses the original image to re-map delineation results. The resulting segmentation is satisfactory (Fig. 4(b)).

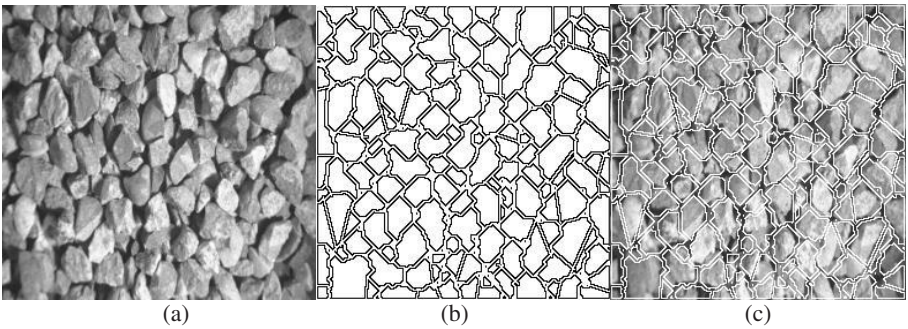
When one acquires (or takes) rock fragment images in the field, the lightning is uncontrolled; therefore, it cannot be avoided having uneven illumination images. Uneven illumination is a serious problem for image processing and image segmentation not only for rock fragments and also for other object. Uneven illumination correction is a hot topic in the research of image processing. In general, the regular shadows can be removed by using some standard filters, but for the random shadows, there is no standard filter or algorithm can be used for uneven illumination correction.

Rock fragments are in field, lightning is from the natural sun (light strength varies from time to time), some natural objects (e.g. clouds, forest, mountains) and large man-made objects (e.g. trucks, trans) maybe nearby the area one wants to take images, which may create uneven illumination (i.e. shadows) on the images. Some times, in a fragment image, it includes high lightning area and dark shadows, which



(a) (b) (c)

Fig. 3. Aggregate delineation for image of less surface noise: (a) original image; (b) aggregate delineation result; and aggregate boundaries or contours



(a) (b) (c)

Fig. 4. Aggregate delineation for image of serious surface noise: (a) original image; (b) boundaries or contours of aggregates; and (c) aggregate delineation result

make image segmentation extremely difficult. It is not possible to use the segmentation algorithms based on grey level similarity. In the newly studied fragment delinea tion algorithm, since it uses valley edges as cues for object delineation, it is not affected by uneven illumination much.

5 In Conclusion

In this paper, a methodology for image segmentation of densely packed aggregate particles is presented, studied and tested; Image classification is very important for aggregate particle delineation. The classification algorithm produces image class labels, useful in subsequent image segmentation. The aggregate delineation algorithm studied is actually based on both valley-edge detection and valley-edge tracing. The presented rock aggregate delineation algorithm seems robust for densely packed complicated particles; it is also suitable for other similar applications in the areas of biology, medicine and metal surface etc.

References

1. Gallagher, E: Optoelectronic coarse particle size analysis for industrial measurement and control. Ph.D. thesis, University of Queensland, Dept. of Mining and Metallurgical Engineering (1976).
2. Norbert H. Maerz and Tom W. Palangio, Post-Muckpile, Pre-Primary Crusher, Automated Optical Blast Fragmentation Sizing, Fragblast,, Volume 8, Number 2 / June, 2004, pp. 119 – 136, Publisher: Taylor & Francis.
3. Schleifer J, Tessier B: Aggregateation Assessment using the FragScan System: Quality of a Blast. Int. J. Fragblast, Volume 6, Numbers 3-4 (2002), 321 – 331.
4. Kemeny J, Mofya E, Kaunda R, Lever P: Improvements in Blast Aggregateation Models Using Digital Image Processing. Int. J. Fragblast, Volume 6, Numbers 3-4 (2002), 311 – 320.
5. Wang, W.X. and Fernlund, J., Shape Analysis of Aggregates. KTH-BALLAST Report no. 2, KTH, Stockholm, Sweden (1994).
6. Pal, N.R, Pal, S.K: A review of image segmentation techniques. Int. J. Pattern Recognition, Vol. 26, No. 9 (1993), 1277-1294.
7. Wang, W.X., 1999, Image analysis of aggregates, J Computers & Geosciences, No. 25, 71-81.
8. Wang, W.X:, Binary image segmentation of aggregates based on polygonal approximation and classification of concavities. Int. J. Pattern Recognition, 31(10) (1998), 1503-1524.
9. Otsu, N: A threshold selection method from gray-level histogram. *IE*EE Trans. Systems Man Cybernet, SMC-9(1979), 62-66.
10. Suk, M., Chung, SM: A new image segmentation technique based on partition mode test. Int. J. Pattern Recognition Vol. 16, No. 5 (1983), 469-480.

Mining Temporal Co-orientation Pattern from Spatio-temporal Databases

Ling-Yin Wei[1] and Man-Kwan Shan[2]

[1] Institute of Computer Science and Engineering, National Chiao Tung University, Taiwan
[2] Department of Computer Science, National Chengchi University, Taiwan
{g9315, mkshan}@cs.nccu.edu.tw

Abstract. A spatial co-orientation pattern refers to objects that frequently occur with the same spatial orientation, e.g. left, right, below, etc., among images. In this paper, we introduce temporal co-orientation pattern mining which is the problem of temporal aspects of spatial co-orientation patterns. A temporal co-orientation pattern represents how spatial co-orientation patterns change over time. Temporal co-orientation pattern mining is useful for discovering tactics from play sequences of sports video data, because the most tactic patterns of basketball competition are constituted of such spatial co-orientation patterns in time order. We propose the three-stage approach, which transforms the problem into sequential pattern mining, for mining temporal co-orientation patterns. We experimentally evaluate the performance of the proposed algorithm and analysis the effect of these stages.

1 Introduction

Spatial data mining is an important task to discover interesting patterns from spatial or image datasets. There exist three basic types of spatial relationships: distance, topological, and directional relationship. Many works have focused on spatial co-location pattern mining concerning with the distance relationship. While little attention has been paid on patterns concerning the directional relationship, we have proposed the **spatial co-orientation pattern mining** from a set of spatial images which discover the spatial objects that frequently occur and collocate with the same orientation among each other [6].

Recently, some research extended from spatial patterns to spatio-temporal patterns [1,3,4,5,7]. In this paper, we extend the concept of spatial co-orientation patterns to **temporal co-orientation patterns**. For a spatio-temporal database consisting of sequences of images, temporal co-orientation patterns refer to the common patterns of changes of spatial co-orientation among sequences. For example, Fig. 1 shows a spatio-temporal database, and Fig. 2 is one of the temporal co-orientation patterns with occurrences no less than two. We propose three-stage algorithm, TCPMiner, for mining the temporal co-orientation patterns.

One of the applications of temporal co-orientation patterns is sports video analysis. Sports video analysis aims to provide assistance for training. Much research has been

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 895–903, 2007.

done on analyzing sports video. One of the important tasks in sports analysis is to summarize the play tactics from sports video. While regarding a video as an ordered sequence of images, tactics summarization of play tactics is, in fact, the problem of our proposed temporal co-orientation pattern mining.

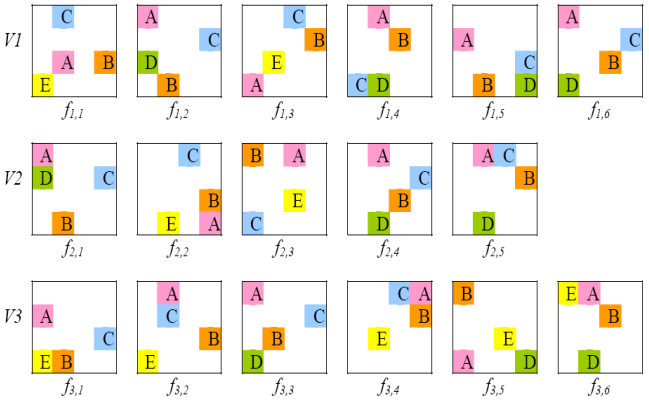


Fig. 1. A spatio-temporal database *STDB*

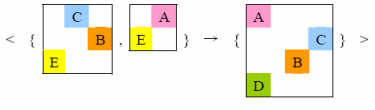


Fig. 2. A temporal co-orientation pattern in Fig. 1

2 Related Work

Mobile group pattern miming is to discover group patterns of users, determined by distance threshold and minimum time duration [1,7]. The group members are physically close to one another and stay together for some meaningful duration when they act as a group. This research reveals the neighboring relation over time without relative direction over time and the change of spatial relation among objects over time. J. Wang et al. proposed some kinds of mining spatio-temporal patterns in spatio-temporal databases. In [3], the work is discovering the topological patterns satisfying not only the spatial proximity relationships but also the temporal proximity relationships. Briefly, the topological patterns occur near in a region and near in time. The main idea of algorithm in [3] is to generate the projected database for mining topological patterns and this idea is similar to the pattern-growth approach. In [5], the work is to discover generalized spatio-temporal patterns which are intended to describe the repeated sequences of events occurring in small neighborhoods, and similarly does [4]. [4] and [5] utilize the concept of Apriori-like approach and pattern-growth approach for mining generalized spatio-temporal patterns, respectively. But these works focus on neighboring relation over time without dealing with relative direction among objects over time.

3 Problem Definition

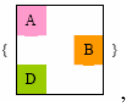
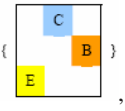
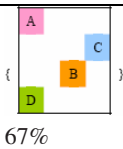
Definition 1. Let a spatio-temporal database $STDB=\{V_1, V_2, \ldots, V_N\}$ be a set of image sequences. Each image sequence V_i is an ordered list of images, sorted by time in increasing order. We denote a sequence V_i by $\{f_{i,1}, f_{i,2},\ldots,f_{i,n_i}\}$, where $f_{i,j}$ is the j-th image of V_i and n_i is the total number of images in V_i. Each image $f_{i,j}=\{(O_1, X_1, Y_1), (O_2, X_2, Y_2),\ldots, (O_{m_k}, X_{m_k}, Y_{m_k})\}$ is a set of triples, where $\forall h$, $1\le h\le m_k$, O_h is an object type, and (X_h,Y_h) is the location of an object O_h.
Note that we regard each image of sequences as a symbolic picture [6]. In this paper, we shorten the notation $V_i=\{f_{i,1}, f_{i,2},\ldots,f_{i,n_i}\}$ to $V=\{f_1, f_2,\ldots,f_{n_i}\}$ if recognizing the different image sequences is not necessary.

Definition 2. A symbolic picture g is said to **be contained in** a sequence of symbolic pictures $V=\{f_1, f_2,\ldots,f_{n_i}\}$ iff there exists an integer $1\le j\le n_i$ such that g is a subpicture [6] of f_{n_j}, denoted as $g\subseteq f_{n_j}$.

Definition 3. Given a spatio-temporal database *STDB*, the **global support** of a symbolic picture g is the percentage of sequences in *STDB* that contain the symbolic picture g. If the global support of a symbolic picture g is greater than or equal to a given minimum global support threshold, ***minGSup***, g is a **spatial co-orientation pattern of *STDB*.** The size of the spatial co-orientation pattern g is the number of objects occurring in g. A spatial co-orientation pattern of size n is called a **size-*n* spatial co-orientation pattern**. A spatial co-orientation pattern g is **maximal** if g is not contained in any other spatial co-orientation pattern of *STDB*.

Example 1. Fig. 1 shows a spatio-temporal database *STDB* of three symbolic picture sequences. If *minGSup* is 50%, some of the size-2, size-3, and size-4 spatial co-orientation patterns are shown in Table 1.

Table 1. Some of spatial co-orientation patterns of Fig. 1

	Size-2		Size-3		Size-4
Pattern					
GSup.	100%	67%	100%	100%	67%

Definition 4. Let f^s_1 and f^s_2 be two sets of symbolic pictures. A set of symbolic pictures f^s_1 is said to **be contained in** a set of symbolic pictures f^s_2 , iff for each symbolic picture $g\in f^s_1$, there exists a symbolic picture $g' \in f^s_2$ such that $g\subseteq g'$.

Definition 5. Let $F^s_1=\{f^s_{1,1}, f^s_{1,2},\ldots, f^s_{1,m}\}$ be a sequence of sets of symbolic pictures. That is, each element of F^s_1, $f^s_{1,j}$, $1 \le j \le m$, is a set of symbolic pictures. A sequence F^s_1 is said to **be contained in** a sequence $F^s_2=\{f^s_{2,1}, f^s_{2,2},\ldots, f^s_{2,n}\}$, iff there exist integers $1\le i_1<i_2<\ldots<i_m\le n$ such that for each $j\in\{1,\ldots,m\}$, $f^s_{1,j}\subseteq f^s_{2,i_j}$.

Example 2. Fig. 3 shows two sequences $F^s_1=\{f^s_{1,1}, f^s_{1,2}\}$ and $F^s_2=\{f^s_{2,1}, f^s_{2,2}, f^s_{2,3}\}$. F^s_1 is contained in F^s_2, because there exist integers $1\le i_1<i_2\le 3$, where $i_1=1$ and $i_2=3$, such that for each symbolic picture g in $f^s_{1,1}$, there exists a symbolic picture g' in $f^s_{2,1}$ such that $g\subseteq g'$, and for each symbolic picture g in $f^s_{1,2}$, there exists a symbolic picture g' in $f^s_{2,3}$ such that $g\subseteq g'$.

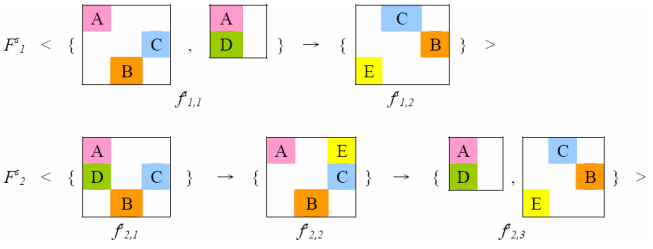


Fig. 3. Example of two sequences of sets of symbolic pictures

Definition 6. Given a spatio-temporal database *STDB*, let $F^s=\{f^s_1, f^s_2,\ldots, f^s_m\}$ be a sequence of sets of symbolic pictures. The **global support** of F^s is the percentage of sequences in *STDB* that contain F^s. F^s is the **temporal co-orientation pattern** of *STDB* if (1) $\forall i\in\{1,\ldots,m\}$, each symbolic picture in f^s_i is a spatial co-orientation pattern of *STDB* and the size of f^s_i is greater than or equal to two, and (2) the global support of F^s is greater than or equal to a given minimum global support threshold, ***minGSup***.

Definition 7. Given a temporal co-orientation pattern $F^s=\{f^s_1, f^s_2,\ldots, f^s_n\}$, the **length** of the temporal co-orientation pattern F^s is the number of sets of symbolic pictures of F^s. A temporal co-orientation pattern of length n is called a **length-*n* temporal co-orientation pattern**. A temporal co-orientation pattern F^s is called a **maximal temporal co-orientation pattern** if F^s is not contained in any other temporal co-orientation pattern of *STDB*.

For instance, Fig. 2 is a length-2 temporal co-orientation pattern. Note that a spatial co-orientation pattern is also a length-1 temporal co-orientation pattern if the size of this pattern is greater than or equal to two.

4 Proposed Algorithm

To solve the problem of mining temporal co-orientation patterns, we also employ the 2D representation [6] to represent symbolic pictures. Given a spatio-temporal database of sequences of symbolic pictures, the problem of temporal co-orientation patter mining thus becomes the discovery of the frequent sequences of sets of 2D strings among a database of sequences of 2D strings. Each such frequent sequence of sets of 2D string is a temporal co-orientation pattern.

We propose the three-stage algorithm, TCPMiner, to discover the temporal co-orientation patterns. The proposed approach consists of three stages of processes. The first stage is discovering all size-2 spatial co-orientation patterns from a given spatio-temporal database *STDB*. The second stage transforms the problem into that of

sequential pattern mining. The last stage reconstructs the symbolic pictures from the discovered patterns of the second stage. Fig. 4 shows the detail of TCPMiner algorithm for discovering temporal co-orientation patterns.

In Fig. 4, at first, we preprocess the given spatio-temporal database *STDB* to transform each symbolic picture of sequences in *STDB* to a 2D string, and then

Input: A spatio-temporal database *STDB*, and a minimum global support threshold *minGSup*
Output: A set of temporal co-orientation patterns *FP*

1. $STDB_{2D}$=preprocess(*STDB*); //$STDB_{2D}$: a set of sequences of sets of 2D strings
2. // 1st stage
3. Scan $STDB_{2D}$ to generate O_1;//O_k: a set of size-*k* spatial co-orientation patterns
4. CO_2=GenCandidate(O_1);
5. For each *c* in CO_2 //CO_k:a set of candidate size-*k* spatial co-orientation patterns
6. Scan $STDB_{2D}$ to count the global support of *c*;
7. If *c* is frequent then insert *c* to O_2;
8. // 2nd stage
9. P_{int}=UID(O_2); //P_{int}: a set of integers
10. $STDB_{int}$ =TranI($STDB_{2D}$, P_{int}, O_2); //$STDB_{int}$: a database of sequences of sets of integers
11. FP_{int}=SeqMiner($STDB_{int}$); //FP_{int}: a set of frequent sequences of sets of integers
12. // 3rd stage
13. *FP*=TranII(FP_{int}, P_{int}, O_2);
14. Return *FP*;

Fig. 4. TCPMiner algorithm

$STDB_{2D}$

TID	Sequence
V1	< (E<A=C<B, E<A=B<C) (A=D<B<C, B<D<C<A) (A<E<C<B, A<E<B<C) (C<A=D<B, C=D<B<A) (A<B<C=D, B=D<C<A) (A=D<B<C, D<B<C<A) >
V2	< (A=D<B<C, B<C=D<A) (E<C<B=A, A=E<B<C) (B=C<A=E, C<E<A=B) (A=D<B<C, D<B<C<A) (A=D<C<B, D<B<A=C) >
V3	< (A=E<B<C, B=E<C<A) (E<A=C<B, E<B<C<A) (A=D<B<C, D<B<C<A) (E<C<A=B, E<B<A=C) (A=B<E<D, A=D<E<B) (E<A=D<B, D<B<A=E) >

P_{int} ↓**UID**

UID	Pattern	UID	Pattern	UID	Pattern	UID	Pattern
1	(A<B,B<A)	2	(A=B,A<B)	3	(A<C,C<A)	4	(C<A,C<A)
5	(A=D,D<A)	6	(A<E,A<E)	7	(A=E,E<A)	8	(E<A,A=E)
9	(E<A,E<A)	10	(B<C,B<C)	11	(C<B,B<C)	12	(D<B,B<D)
13	(D<B,D<B)	14	(B<E,E<B)	15	(E<B,E<B)	16	(D<C,D<C)
17	(E<C,E<C)						

$STDB_{int}$ ↓**TranI**

TID	Sequence
V1	<(9,11,15,17)(1,3,5,10,12,16)(6,11,15,17)(1,4,5,13)(1,3,10)(1,3,5,10,13,16)>
V2	<(1,3,5,10,12)(2,8,11,15,17)(4,7,14)(1,3,5,10,13,16)(1,5,11,13,16)>
V3	<(1,3,7,10,17)(1,9,11,15,17)(1,3,5,10,13,16)(9,11,15,17)(2,6,14)(1,5,8,13)>

Fig. 5. Example of transformation in TCPMiner

generate the database $STDB_{2D}$. The detail of GenCandidate function is shown in [6]. In step 9, each spatial co-orientation pattern in O_2 is assigned a unique item number, and a hash table of item numbers P_{int} with respect to all patterns in O_2 is generated. According to P_{int}, we transform each sequence of $STDB_{2D}$ to a sequence of itemsets, that is, each symbolic picture is transformed into an itemset. For each symbolic picture, each size-2 spatial co-orientation pattern contained in it constitutes the transformed itemset. After the transformation, sequential pattern mining algorithm [2] is utilized to generate all frequent sequence of itemsets. For the patio-temproal database *STDB* in Fig. 1, Fig. 5 shows the outcome of TranI in TCPMiner if *minGSup* is 50%.

The last stage generates the temporal co-orientation patterns by transforming each itemset into a set of symbolic pictures. Note that each item corresponds to a size-2 symbolic picture. It seems straightforward to transform each itemset into a set of size-2 symbolic pictures. However, the patterns what we wish are the maximal temporal co-orientation patterns. It is essential that the transformation should be performed such that the reconstructed symbolic picture is as maximal as possible. This stage proceeds similarly to the Apriori-based approach for frequent itemset mining. The detailed procedures of the function TranII is shown in Fig. 6.

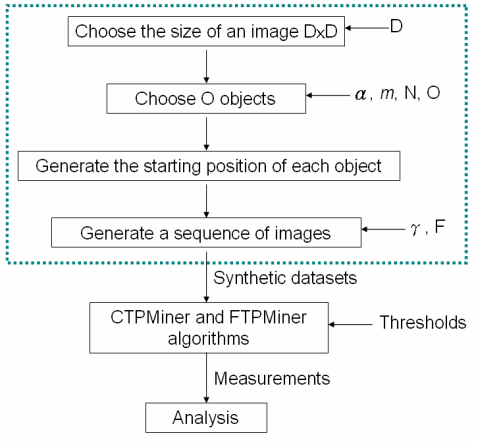
Input: A set of frequent sequences of itemsets FP_{int}, a set of integers P_{int}, and a set of size-2 spatial co-orientation patterns O_2
Output: A set of temporal co-orientation patterns *FP*
1. Let S_{int}=< I_1, I_2,…, I_m > where I_i is a set of integers; // S_{int}: a sequence of sets of integers
2. Let S_{2D}=< I'_1, I'_2,…, I'_m > where I'_i is a set of 2D strings;
3. For each S_{int} in FP_{int}
4. Generate S_{2D} from S_{int};
5. For each I_i in S_{int}
6. j=3;
7. While(|I_i|≥ (j)*(j-1)/2){
8. L=GenCombination((j)*(j-1)/2, I_i);
9. For each l in L
10. F_j =GenFP(l, T); //T: a table
11. Insert F_j into I'_i; //F_j: a size-j spatial co-orientation pattern
12. j++; }
13. Max(I'_i);
14. FP=∪S_{2D};

Fig. 6. Algorithm of TranII function

Example 3. Given the spatio-temporal database *STDB* in Fig. 1, FP_{int} is generated from the database $STDB_{int}$ in Fig. 5 if *minGSup*=50%. For instance, S_{int}={I_1}={(9,11,15,17)} is a pattern in FP_{int}. In Fig. 6, the step 10 generates size-j spatial co-orientation pattern according to l in L. If j=3 and l=(11,15,17), we join size-2 spatial co-orientation patterns 11 and 15 to generate size-3 spatial co-orientation pattern, UID=18(11,15,17):(E<C<B,E<B<C), and record it in T. Finally, S_{int}={(9,11,15,17)} is restored as {(E<A,E<A),(E<C<B,E<B<C)}.

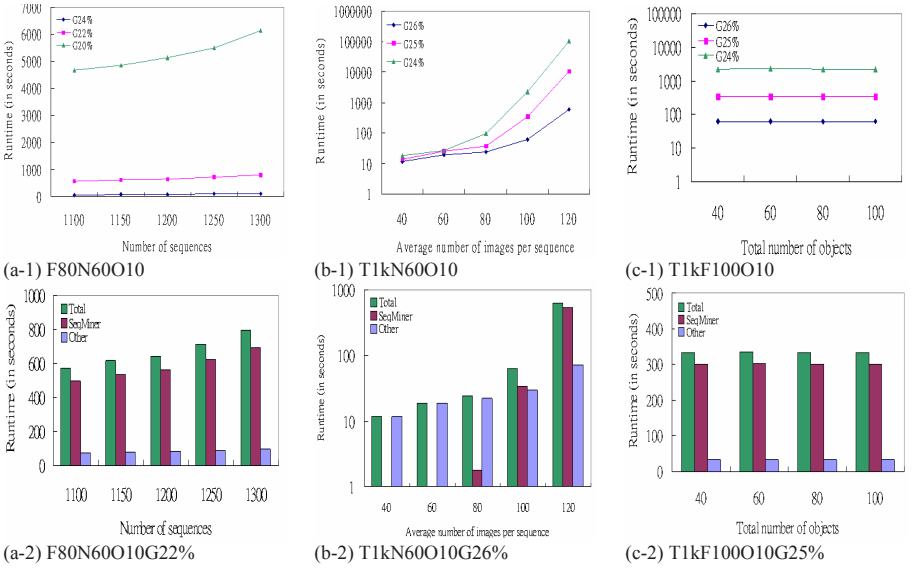
5 Experiment

We evaluate the performance of TCPMiner using synthetic datasets. Fig. 7 illustrates the flow of our experimental design and parameters. In order to simulate the sports games for a given team against others, we assume the objects (players) appearing in



Symbol	Description
T	Number of sequences
D x D	Size of an image
F	Average number of images per sequence
N	Total number of objects
O	Number of objects per image
G	Minimum global support

Fig. 7. Flow of experimental design and parameters



(a-1) F80N60O10 (b-1) T1kN60O10 (c-1) T1kF100O10

(a-2) F80N60O10G22% (b-2) T1kN60O10G26% (c-2) T1kF100O10G25%

Fig. 8. (a) Runtime v.s. number of sequences. (b) Runtime v.s. average number of images per sequence. (c) Runtime v.s. total number of objects.

each video are selected from two distinct teams, the user-specified team α and the other team by choosing randomly from (N/*m*)-1 teams, where *m* is the number of players per team. We chose O objects while O/2 objects were selected from α and the others from the randomly selected team. Finally, we generate a sequence of images with the length of $F\times(1\pm r)$ images by random walk in uniform probability distribution. In our experiments, we generated all images by setting D=6, *m*=10, O=10, and *r* =0.2. The convention of T1kF250N60O10G22% means that the data set includes 1000 sequences, average number of images per sequence is 250, total number of objects is 60, average number of objects per image is 10, and minimum global support is 50%. We implement TCPMiner in C++. The experiments were performed on AMD Opteron, 2.39GHz, 3.93G main memory PC.

We measure the efficiency of proposed algorithm as the function of other parameters. Moreover, we compare relative execution time between the stage of sequential pattern mining and the other stages (the first stage and the third stage denoted as SeqMiner and Other, respectively.) The execution times of TCPMiner scale up gradually with the number of sequences in (a-1). (a-2) illustrates that the effect of number of sequences of SeqMiner is more obvious than the effect of Other as the number of sequences increases. (b-1) demonstrates that the execution time of TCPMiner increases dramatically as the average number of per sequence increase. This is because that the size of database and the length of temporal co-orientation pattern increase. In (b-2), the execution time of SeqMiner increases more rapidly than Other as the number of images per sequence is increasing. As shown in (c-1), the effect of the total number of objects is not obvious while the total number of objects increases. (c-2) indicates that the effects of total number of objects are not obvious with respect to SeqMiner and Other.

6 Conclusion and Future Work

We proposed the temporal co-orientation pattern, and the algorithm, TCPMiner, for mining temporal co-orientation patterns. One of the core steps of the proposed algorithm is the sequential pattern mining. The experimental results showed that the step of sequential pattern mining dominates the execution time due to the characteristic of temporal co-orientation patterns. It is essential to develop algorithms for mining sequential patterns with respect to the characteristic of temporal co-orientation patterns. Moreover, it is worth to explore the applications of the proposed temporal co-orientation pattern mining.

References

1. S. Hwang, Y. Liu, J. Chiu, and E. Lim. Mining Mobile Group Patterns: A Trajectory-Based Approach. *Proc. of 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD* (2005).
2. J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, and Q. Chen. Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 11 (2004).

3. J. Wang, W. Hsu, and M. L. Lee. A Framework for Mining Topological Patterns in Spatio-temporal Databases. *Proc. of 14th ACM Conference on Information and Knowledge Management CIKM* (2005).
4. J. Wang, W. Hsu, and M. L. Lee. Mining Generalized Spatio-Temporal Patterns. *Proc. of 10th International Conference on Database Systems for Advanced Applications DASFAA* (2005).
5. J. Wang, W. Hsu, M. L. Lee, and J. Wang. FlowMiner: Finding Flow Patterns in Spatio-Temporal Databases. *Proc. of 16th IEEE International Conference on Tools with Artificial Intelligence ICTAI* (2004).
6. L. Y. Wei and M. K. Shan. Efficient Mining of Spatial Co-orientation Patterns from Image Databases. *Proc. of 3rd IEEE International Conference on Systems, Man, and Cybernetics SMC* (2006).
7. Y. Wang, E. Lim, and S. Hwang. On Mining Group Patterns of Mobile Users. *Proc. of 14th International Conference on Database and Expert Systems Applications DEXA* (2003).

Incremental Learning of Support Vector Machines by Classifier Combining

Yi-Min Wen[1,2] and Bao-Liang Lu[1,⋆]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dong Chuan Road, Shanghai 200240, China
{wenyimin; bllu}@sjtu.edu.cn
[2] Hunan Industry Polytechnic, Changsha 410007, China

Abstract. How to acquire new knowledge from new added training data while retaining the knowledge learned before is an important problem for incremental learning. In order to handle this problem, we propose a novel algorithm that enables support vector machines to accommodate new data, including samples that correspond to previously unseen classes, while it retains previously acquired knowledge. Furthermore, our new algorithm does not require access to previously used data during subsequent incremental learning sessions. The proposed algorithm trains a support vector machine that can output posterior probability information once an incremental batch training data is acquired. The outputs of all the resulting support vector machines are simply combined by averaging. Experiments are carried out on three benchmark datasets as well as a real world text categorization task. The experimental results indicate that the proposed algorithm is superior to the traditional incremental learning algorithm, Learn++. Due to the simplicity of the proposed algorithm, it can be used more effectively in practice.

1 Introduction

The brain of human beings has powerful ability of incremental learning. Therefore, how to develop brain-like computing model, how to implement incremental learning is one challenge problem in machine learning research. In real world applications, there are three scenarios need incremental learning: all training data cannot be gathered at one time for the cost of collecting data. As a result the data are acquired batch by batch; some real world applications need instant learning once some training data obtained; all training data cannot be loaded into the memory of computers if the training set is very large. According to Jantke [1], incremental learning is to construct new hypothesis by using only the hypothesis before and the recent information on hand. Zhou and Chen [2] distinguished three kinds of incremental learning tasks: Example-incremental learning

⋆ To whome correspondence should be addressed. This work was supported in part by the National Natural Science Foundation of China under the grants NSFC 60375022 and NSFC 60473040, and the Microsoft Laboratory for Intelligent Computing and Intelligent Systems of Shanghai Jiao Tong University.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 904–911, 2007.

(E-IL); Class-incremental learning (C-IL); and Attribute-incremental learning (A-IL). However, C-IL and A-IL have not been received much attention so far. Syed *et al.* [3] introduced two types of incremental learning methods: instance learning, which uses one example at a time, and block by block learning, which uses a suitable-size subset of samples at a time.

At present, however, the essence of the training algorithms of various kinds of artificial learning systems is an optimization procedure that aims to ensure the generalization ability based on the current learning environment. Therefore, all the current machine learning algorithms don't adapt for incremental learning in nature. The non-adaption lies in that the computation model lacks the ability to get new knowledge or cannot retain the knowledge learned before [4]. The training of artificial neural networks is a gradient descent process, and therefore the modification of connection weights will damage the learned knowledge. The training of SVMs is a global optimization based on all training data. As a result, new added training data will make support vectors change [5].

Classifier combining is a useful method for machine learning [6] [7] [8]. Many scholars have applied classifier combining techniques to incremental learning. Polikar *et al.* proposed Learn++ based on AdaBoost algorithm [9]. Lu and Ichikawa proposed an incremental learning model based on emergence theory [10]. Macek proposed incremental learning algorithms based on bagging and boosting and successfully applied them to EEG data classification [11]. Wang *et al.* used weighted ensemble classifiers to mine concept-drifting data stream [12]. Like bagging, a model of incremental learning by classifier combining (ILbyCC) is proposed in this paper.

2 Incremental Learning by Classifier Combining

2.1 Definition of Batch Incremental Learning

Definition 1. *Given a sequence of training datasets* $S_1, S_2, ..., S_m$*, where* $S_i = \{(x_{i_j}, c_{i_j}) | x_{i_j} \in R^n, c_{i_j} \in L_i \subseteq \{1, 2, ..., k\}, 1 \leq j \leq n_i\}, 1 \leq i \leq m$*.* L_i *indicates the set of class label in training dataset* S_i*. Lets* E_1 *denotes the classifier trained on* S_1*, the batch incremental learning procedure* IL *can be illustrated as:* $IL(S_i, E_{i-1}) = E_i,\ 2 \leq i \leq m.$

In this paper, we only consider the case where the number of class labels don't decrease, i.e., $L_1 \subseteq L_2 \subseteq ... \subseteq L_m$.

ILbyCC takes a frame of modular architecture. Modular architecture can make classifier easy adapt to incremental learning. ILbyCC trains a new classifier on an incremental batch and saves it. All the classifiers trained by far are combined into one combined classifier. The training algorithm of ILbyCC can be illustrated as: $M(f_1, f_2, ..., f_{i-1}, f_i) = E_i$, where M denotes the strategy for classifier combining, and E_i denotes the current combined classifier.

Table 1. The problem statistics and the parameters used in SVMs

Data set	#attributes	#training data	#test data	#class	C	γ
Optical Digits	1024	1200	4420	10	128	0.002
Vehicle Silhouette	18	630	216	4	1500	0.00001
Concentric Circle	2	1200	500	5	128	0.125
Yomiuri News Corpus	5000	424310	87268	9	64	0.125

2.2 Combining Classifiers by Averaged Bayes

Given m classifiers that can output posterior probability information, when a test input x comes, the j-th classifier outputs the posterior probability of x belonging to all the classes:

$$P_j(y = i|x), i \in \{1, 2, ..., k\}, j = 1, 2, ..., m \tag{1}$$

According to Averaged Bayes, the combined classifier E_m computes the posterior probability of x belonging to all classes as follows:

$$P_{E_m}(y = i|x) = \frac{1}{m} \sum_{j=1}^{m} P_j(y = i|x), \ i \in \{1, 2, ..., k\} \tag{2}$$

According to Bayes rule, x can be classified as the i-th class:

$$i = \arg\ max_{i=1}^{i=k} P_{E_m}(y = i|x) \tag{3}$$

2.3 Incremental Learning Algorithm by Classifier Combining

ILbyCC algorithm is described as Fig.1.

3 Experiments

3.1 Datasets

In order to evaluate the performance of ILbyCC algorithm, experiments are run on four data sets. The first three data sets, Optical Digits Database, Vehicle Silhouette Database, and Concentric Circle Database, are took from Poliker's paper [9] and used as Poliker's strategy. The fourth data set is a part of Yomiuri News Corpus database. We select all the instances of nine classes, such as crime, sport, Asian-Pacific, North-South-American, health, accident, by-time, society, and finance, which will be called as class 1 through class 9. The training data set is randomly divided into 9 incremental batches, S_1 through S_9, where S_1 through S_3 have instances from classes 1, 2, and 3; S_4 through S_6 contain instances from classes 1 through 6; and S_7 to S_9 have instances from classes 1 through 9. The statistics of the tasks are illustrated in Table1. The parameters used in SVMs are selected by cross-validation.

Algorithm: ILbyCC

Input: given two example-incremental learning sequences: $List_1 = \{S_1^1, S_1^2, ..., S_1^m\}$ and $List_2 = \{S_2^1, S_2^2, ..., S_2^n\}$, where $L_1^1 = L_1^2 = ... = L_1^m = L1$, $L_2^1 = L_2^2 = ... = L_2^n = L2$, $L1 \subset L2$. Let $n = 0$, if there is only one example-incremental learning sequence.

Steps:

1. For $t = 1, 2, ..., m$
   (a) Take cross-validation on S_1^t to select the optimal parameters of training algorithm and train a classifier f_1^t on the incremental batch S_1^t.
   (b) Save classifier f_1^t and S_1^t can be discarded.
2. For $t = 1, 2, ..., n$
   (a) Take cross-validation on S_2^t to select the optimal parameters of training algorithm and train a classifier f_2^t on the incremental batch S_2^t.
   (b) Save classifier f_2^t and S_2^t can be discarded.
3. Testing:
   (a) Import a test input x into each f_2^t, $1 \le t \le n$, and calculate the posterior probability of x belonging to all classes: $P_t^j, 1 \le t \le n, j \in L2$.
   (b) Take the rule of classifier combining M to combine f_2^t, $1 \le t \le n$, and get the combined classifier $E_n = M(f_2^1, f_2^2, ..., f_2^n)$, where E_n outputs the posterior probability of x belonging to all classes: $P_{E_n}^j, j \in L2$.
4. If $\arg max_{j \in L2} P_{E_n}^j \in (L2 - L1)$, x can be classified by the value of $\arg max_{j \in (L2-L1)} P_{E_n}^j$. The algorithm ends.
5. If $\arg max_{j \in L2} P_{E_n}^j \in L1$, modify the outputs of E_n by setting $P_{E_n}^j = 0, j \in (L2 - L1)$ and $P_{E_n}^j = \frac{P_{E_n}^j}{\sum_{j \in L1} P_{E_n}^j}, j \in L1$, then take the classifier combining rule M to combine classifiers $\{f_1^1, f_1^2, ..., f_1^m, E_n\}$ and get the combined classifier E. E outputs the posterior probability of x belonging to all classes: $P_E^j, j \in L1$.
6. Classify the test input x by the value of $\arg max_{j \in L1} P_E^j$.
7. The algorithm ends.

Fig. 1. Incremental learning algorithm by classifier combining

In order to test ILbyCC's performance on incremental learning when different incremental step takes different parameters. Optimal parameters in each incremental step were chosen among 25 pairs of (C, γ) by 10-cross-validation. 25 pairs of (C, γ) were generated around the values of (C, γ) in Table1 by a product factor of 2.

In order to ensure the reliability of the experimental results, the first three experiments were repeated 10 times and averaged results were presented. Only the last experiment was run one time for its large size. In order to evaluate the performance of ILbyCC , several exsiting algorithms were run for a comparison study. We adopted the algorithm of Syed [3] that was denoted as ILbySV for convenience. In addition, the basic incremental learning algorithm is *Batch-training*, i.e. when the i-th incremental batch comes, the classifiers trained before are all discarded and $S_1 \bigcup S_2 \bigcup ... \bigcup S_i$ is used to train a new classifier. Obviously, Batch-training should keep all training data gotten by far, and further,

catastrophic forgetting takes place when new data comes. In order to compare ILbyCC with Learn++, the paper directly quotes the experimental results of Learn++ [9]. For convenience, when all the training sessions of ILbyCC uses the same parameters, ILbyCC is denoted as ILbyCC1, when different session of ILbyCC use different parameters, ILbyCC is denoted as ILbyCC2.

3.2 Results and Analysis

Both Fig. 2 and Fig. 4 show that ILbyCC was able to preserve the knowledge learned before and acquire new information. Fig. 3 and Fig. 5 illustrate that ILbyCC can incrementally learn successfully, ILbyCC1 and ILbyCC2 have nearly the same generalization ability, and ILbyCC is slightly good then Learn++. Because all incremental batches are not always in the same distribution, the incremental learning performance of ILbySV fluctuates.

Fig.6 and Fig.8 show that the generalization performance of ILbyCC first decreases slightly when new classes are introduced and increases when training data with the same class labels are continuously added, indicating that ILbyCC can preserve the learned knowledge. From Fig. 7 and Fig.9, it seems that a large improvement on the performance is obtained after new classes that were not available earlier are introduced, but only minor improvements in the performance can be observed from the test accuracy curves when new classes are not introduced, indicating that ILbyCC can learn from new introduced classes.

In Fig. 10, it can be seen that the training time of ILbyCC is far smaller than the training time of Batch-training and ILbySV. The large speedup of ILbyCC can compensate the slight decrease of its generalization performance compared with Batch-training.

Why can ILbyCC work effectively? According to the theory of bias-variance [13], decomposing training data will introduce bias and makes the generalization ability of single classifier decrease, however, decomposing training data will increase the variances between all classifiers and increase the generalization ability

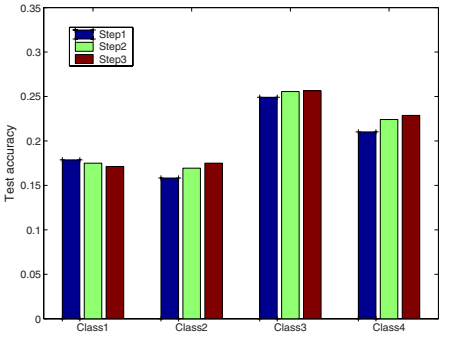


Fig. 2. The generalization performance of ILbyCC1 on each class in Vehicle Silhouette database

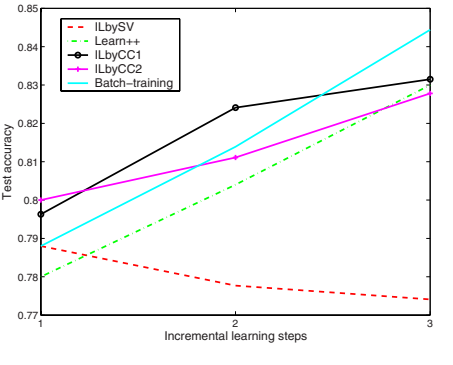


Fig. 3. Accuracy comparison of various incremental learning algorithms on Vehicle Silhouette database

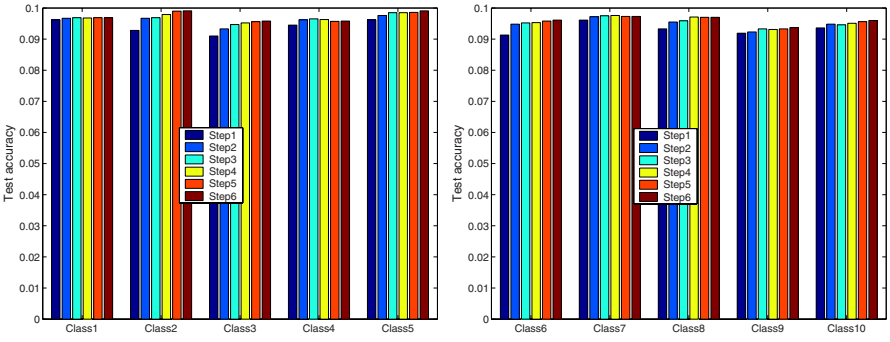


Fig. 4. The generalization performance of ILbyCC1 on each class of Optical digits database

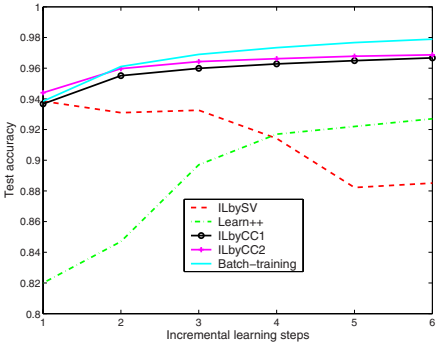


Fig. 5. Accuracy comparison of various incremental learning algorithms on Optical digits database

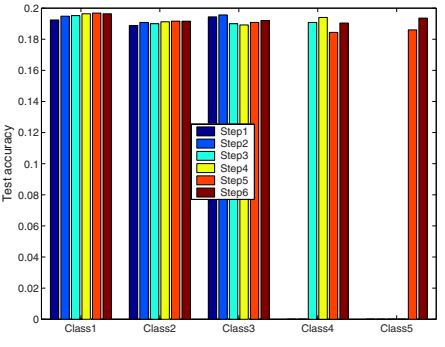


Fig. 6. The generalization performance of ILbyCC1 on each class of Concentric Circle database

of the combined classifier, which compensates the decrease of the generalization ability caused by decomposition. Therefore, ILbyCC has nearly the same test accuracy with Batch-training. In addition, the combining rule (2) can automatically invalidate the classifiers that is not much confident of its outputs, i.e., given $P_j(y = 1|x) \approx ... \approx P_j(y = k|x)$, the result of the equation (3) will not be influenced by the outputs of the j-th classifier. Therefore, Averaged Bayes can automatically select the classifiers that is confident of its outputs to combine.

Note that the performance of ILbyCC1 and ILbyCC2 in all the simulations are nearly the same, it is very interesting to observe that the time complexity for selecting optimal parameters is decreased by training data decomposition. It is not reasonable for incremental learning algorithm to wait for all training data collected to select optimal parameters. It is also not reasonable to apply the parameters, which is gotten from the first incremental batch, to the following incremental steps. Therefore, ILbyCC not only decreases the time complexity of parameter selection but also makes incremental learning possible.

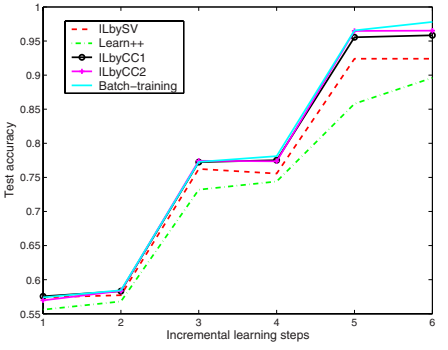


Fig. 7. Accuracy comparison of various incremental learning algorithms on Concentric Circle database

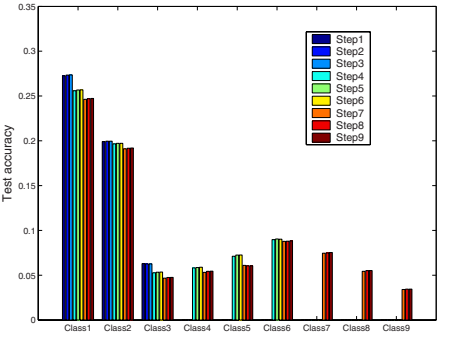


Fig. 8. The generalization performance of ILbyCC1 on each class in Yomiuri News Corpus database

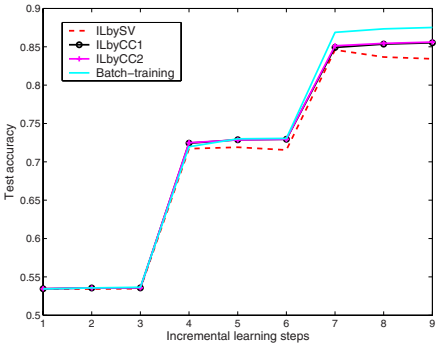


Fig. 9. Accuracy comparison of various incremental learning algorithms on Yomiuri News Corpus database

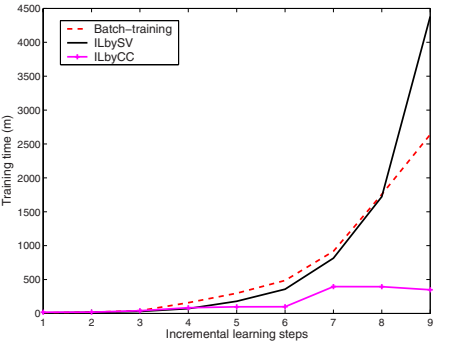


Fig. 10. Comparison of training time on Yomiuri News Corpus database

3.3 Discussions

Compared with Learn++, the proposed ILbyCC satisfies the criteria proposed by Polikar [9] and has comparable incremental learning ability, but ILbyCC can be implemented more simply. Learn++ is a kind of AdaBoost in essence, Learn++ should use more parameters and train more classifiers. Note that ILbyCC is a bagging-like algorithm, ILbyCC can be parallized for training speedup, while Learn++ can only be implemented in serial. In addition, ILbyCC needs no communication between classifiers, it can well protect the privacy of data. The work in this paper can prove the availability of the algorithm estimating the posterior probabilistic of SVMs. To our best knowledge, ILbyCC is the first application to apply posterior probabilistic SVMs to real problem.

4 Conclusions

In this paper, we have proposed a novel incremental learning algorithm ILbyCC that uses Averaged Bayes rule to combine classifiers. The experimental results indicate that ILbyCC can not only preserve the knowledge learned before but also can learn new knowledge from new added data and further new knowledge from new introduced classes. Three main advantages of ILbyCC over existing algorithms are simply implementing, small time complexity for parameter selection, and training time saving. In addition, the proposed algorithm is a general framework of incremental learning and any machine learning algorithm that can output posterior probabilistic can be integrated into ILbyCC.

References

1. Jantke, P.: Types of Incremental Learning. AAAI Symposium on Training Issues in Incremental Learning, March 23-25, Standford CA, 1993
2. Zhou, Z.H. and Chen, Z.Q.: Hybrid Decisions Tree. Knowledge-Based System, 15 (2002) 515-528
3. Syed, N.A., Huan, L., and Sung, K.K.: Handling Concept Drifts in Incremental Learning with Support Vector Machines. In: Proceedings of KDD-99, San Diego, CA,USA, 1999
4. Grossberg, S.: Nonlinear Neural Networks: Principles, Mechanisms and Architectures. Neural Networks, 1 (1988) 17-61
5. R*ü*ping, S.: Incremental Learning with Support Vector Machines. In: Proceedings of the IEEE International Conference on Data Mining, San Jose, CA (2001)
6. Lu, B.L., and Ito, M.: Task Decomposition and Module Combination Based on Class Relations: a Modular Neural Networks for Pattern Classification. IEEE Transaction on Neural Networks, 10 (1999) 1244-1256
7. Zhou, Z.H. and Chen S.F.: Neural Network Ensemble. Chinese J.Computers (in Chinese), 25 (2002) 1-8
8. Xu, L., Krzy*ż*ak, A., and Suen, C.Y.: Methods of Combining Multiple Classifiers and Their Application to Handwriting Recognition. IEEE Transaction on Systems, Man, and Cybernetics, 22 (1992) 418-434
9. Polikar, R., Udpa, L., Udpa, S.S., and Honavar, V.: Learn++: An Incremental Learning Algorithm for Supervised Neural Networks, IEEE Transaction on Systems, Man, and Cybernetics, 31 (2001) 497-508
10. Lu, B.L. and Ichikawa, M.: Emergent Online Learning in Min-max Modular Neural Networks. In: Proceedings of IJCNN'01 (2001) 2650-2655
11. Macek, J.: Incremental Learning of Ensemble Classifiers on ECG data. In: Proceedings of CBMS'05 (2005)
12. Wang, H.X., Fan, W., Yu, P.S., and Han, J.W.: Mining Concept-drifting Data Streams Using Ensemble Classifiers. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (2003)
13. Breiman, L.: Bagging Predictors. Machine Learning, 24 (1996) 123-140

Clustering Zebrafish Genes Based on Frequent-Itemsets and Frequency Levels

Daya C. Wimalasuriya,[1] Sridhar Ramachandran,[2] and Dejing Dou[1]

[1] Department of Computer and Information Science
University of Oregon, USA
{dayacw,dou}@cs.uoregon.edu
[2] Zebrafish Information Network
University of Oregon, USA
sramacha@uoregon.edu

Abstract. This paper presents a new clustering technique which is extended from the technique of clustering based on frequent-itemsets. Clustering based on frequent-itemsets has been used only in the domain of text documents and it does not consider frequency levels, which are the different levels of frequency of items in a data set. Our approach considers frequency levels together with frequent-itemsets. This new technique was applied in the domain of bio-informatics, specifically to obtain clusters of genes of zebrafish (Danio rerio) based on Expressed Sequence Tags (EST) that make up the genes. Since a particular EST is typically associated with only one gene, ESTs were first classified in to a set of classes based on their features. Then these EST classes were used in clustering genes. Further, an attempt was made to verify the quality of the clusters using gene ontology data. This paper presents the results of this application of clustering based on frequent-itemsets and frequency levels and discusses other domains in which it has potential uses.

1 Introduction

1.1 Clustering Based on Frequent-Itemsets

Clustering based on frequent-itemsets is recognized as a distinct technique and is often categorized under frequent-pattern based clustering methods [10]. There are many clustering techniques categorized under this theme and some of them are not based on frequent-itemsets; they are generally based on the frequent patterns observed in some of the dimensions in high-dimensional data. For instance, the pClustering method [12] which performs clustering by pattern similarity in microarray data analysis is generally identified as a frequent-pattern based clustering technique. While all these clustering techniques clearly have something in common in terms of discovering clusters based on frequent patterns, the patterns involved are quite different in different clustering techniques. In techniques such as pClustering these patterns refer to the patterns observed in the values of some dimensions for a set of objects whereas in techniques based on frequent-itemsets the patterns are the frequent-itemsets.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 912–920, 2007.

Clustering techniques based on frequent-itemsets have been hitherto applied almost exclusively in the domain of text documents. In a pioneering work in this area presented by Beil *et al.* [9], two algorithms for discovering clusters of documents based on frequent terms (a sequence of characters separated from other terms by delimiters) they contain have been developed. The two algorithms presented in this paper are based on the concept of assigning the objects that have a frequent-itemset to a cluster. One of the main advantages of this method is that it provides an inherent definition for the clusters, in terms of the frequent-itemsets they have. Such a meaning is generally not provided by other clustering techniques. However, the clusters defined in this manner can be overlapping and non-exhaustive (not covering all the objects) and the algorithms mentioned above are carefully designed to overcome these issues.

The clustering technique used in this work presents an improvement over the basic frequent-itemset based clustering technique because it takes in to consideration the frequency levels of items in a record or an object. Typically, the relative frequency level of an item within a record or an object is not considered in identifying frequent-itemsets; frequent-itemsets are identified based on the items that are observed in a number of objects or records higher than a predefined threshold. For instance, in a data set consisting of 1000 purchase records and with the use of a cut-off limit of 10% , frequent-itemsets are identified as the items that appear in at least 100 records. While this is sufficient in most situations, the frequency of items within a record is also important in some situations. For example, if each purchase record contains at least 100 units, the quantity or the number of times an item appears in each record is also important; in addition to identifying the itemset A,B,C as being frequent, identifying that 30 units of A, 10 units of B and 5 units of C are frequent provides more information. When using these frequent-itemsets for clustering, paying attention to the relative frequencies of items within records in this manner provides more insight in to common characteristics of the objects or records of the cluster. This served as a basis in developing the clustering technique used here.

1.2 Genes and Expressed Sequence Tags (EST)

A gene can be uniquely described by its sequence of nucleotides, which can be thousands of nucleotides long. There are several techniques that are used to identify genes and the use of Expressed Sequence Tags (ESTs) is one such technique. ESTs provide a snapshot of the DNA that is expressed in a given tissue of a eukaryotic organism at a given time. This is in accordance with the Central Dogma of Molecular Biology, which states that DNA first has to be converted in to RNA through the process of transcription and then the RNA has to be converted to proteins, which do the actual work of altering a cell's chemistry, through the process of translation. ESTs are typically short and restricted to about 300 - 500 nucleotides. Since they indicate the regions of DNA that have been transcribed to RNA, they can be used to identify genetic material that are active in a particular situation. Further, contiguous blocks of DNA can be assembled using ESTs and these blocks can be used to identify genes.

Several organizations keep records of the genes and ESTs discovered by researchers and make them publicly available through online databases. One such organization is GenBank [2], which provides access to different types of genetic data. UniGene [6], which is a part of GenBank, provides details on the genes that have been identified based on ESTs and other techniques. Further, it is possible to directly download the data that show how genes have been constructed using ESTs, for each species as a single file. One such file is available for zebrafish. In essence, it lists out the ESTs of each gene of zebrafish. The details of ESTs can be obtained from GenBank, which provides a list of files that contain the details of all the ESTs found in different species. Each record of a file provides comprehensive details on an EST including its nucleotide sequence and the tissue type and/or development stage it was observed.

The main objective of the research was to identify groups of genes based on the similarity of their EST makeup. Since the ESTs provide an idea of the active genetic material in a particular tissue type at a particular development stage of the organism, it is reasonable to assume that genes that have a highly similar EST make up are active in the same tissue type and/or the same development stage. This would probably indicate that such genes may have similar functions in particular tissues or are involved in a common biological process. There have been previous work on analyzing the EST makeup of genes. GEPIS [4] integrates EST and tissue source information to compute gene expression patterns in normal and tumor samples. EST miner [5] is another work in this area.

Any given EST is a part of one and only one gene since genes are identified based on a contiguous sequence of nucleotides. An EST constitutes a section of the nucleotide sequence of the DNA of an organism. As such the unique identification numbers of ESTs can not be used in clustering genes since a given identification number is found in only one gene. Hence the approach adopted was to classify the ESTs in to a group of classes according to the tissue type and the development stage they are found and then use these EST classes in clustering genes.

2 Related Work

Although the concept of discovering clusters based on frequent-itemsets is generic, it has clearly been associated with clustering of text documents. Beil *et al.* [9] use the term "term set" to highlight the fact that the items in text documents are terms and defines the technique as "frequent term-based clustering" instead of "frequent-itemset based clustering". This name has been used by others as well [10]. According to this technique, when a frequent-itemset is identified the set of records or objects that have the frequent-itemset in concern becomes a potential cluster. Different variations of this clustering technique such as Frequent-Term based Clustering (FTC) and Hierarchical Frequent Term-Based Clustering (HFTC) described in [9] identify the final set of clusters from these potential clusters in different ways. In its pure form, the potential clusters identified using frequent-itemsets are overlapping and non-exhaustive.

Some techniques such as FTC ensure that final clusters are not overlapping and that they cover all the objects (records) making them more consistent with the functionality of classical clustering techniques.

Our work demonstrates the use of a clustering technique based on this method in a very different domain. This shows that this clustering technique is a more generic technique. Further since our technique takes frequency levels into consideration in addition to frequent-itemsets, it might be capable of discovering better clusters. Previous work has shown that paying more attention to the structure of documents would result in better clustering techniques. For instance, Li and Chung [11] have implemented a text document clustering technique, which shows a better performance, by considering frequent word sequences in documents. In addition, this work can be related to other attempts to discover clusters using some kind of summarization of frequent-patterns, such as the research in [13].

3 Our Approach

To identify groups of genes based on the similarity of their EST makeup, we designed a new clustering algorithm based on frequent-itemsets and frequency levels. It is presented below:

Algorithm 1. Clustering based on frequent-itemsets and frequency levels

1: Determine a cut-off frequency level for the data set.
2: Identify items to be used in frequent-itemset mining from the records of the data set using the cut-off frequency level.
3: Determine a threshold to be used in frequent-itemset mining.
4: Identify frequent-itemsets using the apriori algorithm or the FP-growth technique.
5: Extract meaningful frequent-itemsets.
6: Identify the records that have the meaningful frequent-itemsets by scanning the data set for each meaningful frequent-itemset.
7: Present the records that share a meaningful frequent-itemset as a cluster and define the meaning of the cluster in terms of the frequent-itemset.

In steps 1 and 2 the concept of frequency levels is used to identify items in records, to be used in frequent-itemset mining later. This requires a cut-off frequency level as a parameter. For a particular item of an record, the number of items to be used in frequent-itemset mining is determined by performing integer division (or division followed by floor operation) on its frequency within the record using the cut-off frequency level. This can be expressed as follows.

Let the total count of items in the record be n.
Let count of item i in the record be m. $(m \leq n)$
Let cut-off frequency be c. $(c < 1)$
$\therefore$ number of items of i to be used in frequent-itemset mining $= \llcorner (m/n)/c \lrcorner$

A numbering scheme together with a special character is used to represent the fact that the items identified in this manner relate to the same item in the actual record. Assuming that the special character is `$` the following would represent 4 items to be used in frequent-itemset mining, all based on item `A`.

```
A$1, A$2, A$3, A$4
```

In steps 3 and 4, a standard frequent-itemset mining technique is employed on the items identified in the previous steps. This is based on discovering longer frequent-itemsets using shorter frequent-itemsets. Both apriori algorithm and FP-growth technique can be used here. The same method is used in Hierarchical Frequent Term-Based Clustering (HFTC) technique described in [9].

Step 5 extracts the meaningful frequent-itemsets from the set of all frequent-itemsets discovered in step 4. This is necessary because the number included in an item according to the numbering scheme discussed above has a meaning. For example, assuming that the cut-off frequency is 10%, `A$2` represents the second 10% step within the frequency of `A` in the record in concern. This makes some frequent-itemsets meaningless. For instance, if a group of genes have 30% of ESTs of class `A`, all the following are identified as frequent-itemsets, assuming that a cut-off percentage of 10% is used.

```
{A$1},{A$2},{A$3},{A$1, A$2},{A$2, A$3},{A$1, A$3},{A$1, A$2, A$3}
```

Clearly, the itemsets {`A$2, A$3`} and {`A$1, A$3`} do not make sense, since it is not meaningful to say that a group of genes share the second and third or first and third steps of 10% of the EST class in concern. Such meaningless itemsets can be excluded by considering only the itemsets that have `$1` item for each class and where all the numbers are in the consecutive order for each class.

In step 6, the entire data set is scanned again to identify the clusters to which each record belongs. Here, it is checked whether a record has the frequent-itemsets used in defining the clusters. This completes the clustering process and step 7 is concerned with presenting the results.

As mentioned earlier, the resulting clusters may be overlapping and may not be exhaustive. While classical clustering techniques ensure that the discovered clusters are not overlapping, in many real world situations overlapping clusters do exist and are useful. There have been some work on identifying overlapping clusters, particularly in the domain of bio-informatics as presented by Banerjee *et al.* [8]. The same can also be said about clusters which do not cover all the objects of the data set. Therefore, the clusters discovered are left as is.

4 Clustering of Zebrafish Genes Based on Their EST Makeup

4.1 Objective

We used our new clustering algorithm to identify clusters of genes based on the ESTs. First, it was expected to classify the ESTs in to a set of classes based on the tissue type and the development stage they are found. The possibility of

employing the standard clustering techniques was also examined in this work. In particular, partitioning methods and hierarchical methods were explored. One main problem with these methods with regards to this data set is not presenting a clear meaning for the clusters identified. In addition, dealing with a large number of attributes (101, which is the number of EST classes) can also be problematic with some implementations of these techniques.

It was also intended to test the quality of clusters discovered using gene ontology data. These data can be obtained from the Gene Ontology Project [3]. It develops three ontologies to describe gene products in terms of their associated biological processes, cellular components and molecular functions. Each description in one of these three categories is given a unique number known as a Gene Ontology ID (GO-ID) and these GO-ID numbers can be used to describe gene products without ambiguity. The similarity between two genes in terms of their involvement in shared biological processes, presence in cellular components and molecular functions can be obtained by counting the number of common GO-IDs between the two genes.

4.2 Implementation

There are three independent data sets, as follows, and each of them required a significant amount of data preprocessing.

1. The data regarding the ESTs of zebrafish genes obtained from UniGene (the build of 16th July was used)
2. The data regarding ESTs of all species obtained from GenBank (the build of 3rd August was used)
3. The gene ontology data of zebrafish genes available from the Gene Ontology Project (the build of 15th August was used)

Regarding the second data set it was necessary to separate the ESTs of zebrafish from the set of all records. It was also necessary to extract only the GenBank accession number, which uniquely identifies each EST, and the tissue type and the development stage of the EST, which were to be used in identifying classes of ESTs from the records on zebrafish ESTs.

After extracting the accession number, tissue type and development stage of zebrafish ESTs, all the different combinations of tissue types and development stages were identified and each combination was recognized as an EST class. Each EST class was also given a unique class identification number to be used in the subsequent steps. Altogether 101 such classes were identified. Some of such EST classes are shown in Table 1. Note that null values were allowed in one field (tissue type or development stage).

From the first data set, the genes and their ESTs were extracted. Then the IDs of ESTs were replaced by their respective EST classes. At the end of this step, the records contained the EST classes of each gene. Then another program was used to identify the EST classes that had a percentage higher than the cut-off percentage and to list the items to be used in the clustering process based on the concept of frequency levels. The cut-off percentage used was 10%. At the end of this step, the data was ready to be clustered using frequent-itemsets.

Table 1. Examples for EST classes

EST Class ID	*Tissue Type*	*Development Stage*
1	myocardium, endocardium, vessel	Adult
2	embryonic	6 - 48 hours post fertilization
28	olfactory epithelium	null

In terms of Gene Ontology data, the first step was to separate the Gene ID and their GO-IDs from the other data. Identifying the Gene IDs required an additional step because the IDs used by Gene Ontology data were those defined by Zebrafish Information Network [7]. It was also necessary to do some additional processing regarding GO-IDs since in some records, their meaning was changed by another field which added qualifiers such as "not" and "contributes to".

We used a publicly available implementation of the apriori algorithm [1] to discover frequent-itemsets. The minimum support level used for frequent-itemset mining was 5%. Clusters of genes were identified from the entire set of genes. In addition, the genes were divided in to groups based on the number of ESTs found in the genes, and clusters of genes were identified from the genes of each group. The rationale behind identify clusters in this manner is that genes having a similar number of ESTs might have some similarity in behavior. The ranges in the number of ESTs used in identifying groups of genes were based on similar ranges identified in UniGene. 10 such groups were identified, which were named G-0 to G-9. Genes of group G-0 are those listed as having no ESTs. These genes are mainly defined based on entire RNA sequences rather than on ESTs. Such genes do not play a major part here.

4.3 Results

Clusters of genes were identified from the entire set of genes as well as from each group of genes other than Group 0, which was excluded from the clustering process because genes of this group have no ESTs. Altogether 256 clusters were identified. An attempt was also made to measure their similarity using gene ontology data. A global similarity measure was calculated for all the genes with gene ontology data. Then a similarity measure of each cluster was compared with the global similarity measure. Table 2 summarizes the results.

We compared the similarity measure for genes of each cluster with the calculated global similarity measure, which is 13.6042%. The details of one cluster with a high similarity measure are presented below.

```
Cluster ID: G4-46 Group: G4 (5-8 ESTs) Common EST structure:
{Tissue Type = Embryo, Development Stage = 7 different stages}  - 10%
{Tissue Type = null, Development Stage = myoblast}              - 10%
Similarity Measure: 36.4416%
No of genes in the cluster: 345
No of genes with gene ontology data: 105
```

Table 2. Clusters identified

Group #	*# of clusters*	*# of interesting*[a] *clusters*	*Highest similarity measure*
All Genes	23	8	49.6712%
G-1	27	9	93.3756%
G-2	18	9	21.5147%
G-3	36	23	51.1322%
G-4	48	41	36.4416%
G-5	31	23	21.6830%
G-6	25	19	22.7529%
G-7	19	3	15.9517%
G-8	15	1	14.5355%
G-9	14	2	14.1515%

[a] These have a higher than normal similarity measure based on gene ontology data.

4.4 Discussion

The interestingness of the clusters of genes identified arises from the fact that they are active in the same tissue type and the same development stage. If the genes of a cluster that have gene ontology data exhibit a similarity in gene ontology data, it would probably indicate that these genes have a higher level of similarity. However, a higher similarity in gene ontology data can not be seen as directly leading to the conclusion that the genes of that cluster have similar behavior. It depends on several other factors also. The quality of the results is affected by the manner in which the data regarding ESTs are presented in GenBank. Currently, a consistent terminology is not used to describe tissue types and development stages and therefore two terms might in fact mean the same thing. A consistent terminology would lead to more accurate results.

This work demonstrates the use of the technique of clustering based on frequent-itemsets and frequency levels. Since it is a generic technique, this can be used in other domains as well. One obvious candidate for its application is the domain of text documents, where clustering based on frequent-itemsets have previously been used. The use of frequency levels together with frequent-itemsets would result in better clusters here. In addition, it can be applied in several other situations. One example would be to identify precincts or other geographical areas whose populations show a similarity based on factors such as ethnicity, religion or age.

5 Conclusion and Future Work

Our work shows that the technique of clustering based on frequent-itemsets and frequency levels is capable of identifying clusters in an effective manner. More research work is needed to ensure its applicability in different domains. It is important to show that this clustering technique can be generic rather than being restricted to a particular area. Such work would also be required to verify

that the overlapping and non-exhaustive nature of the discovered clusters do not seriously hamper their usefulness. It is also necessary to verify the usefulness of the gene clusters discovered based on their EST makeup and to extract more information from them. The work presented in GEPIS [4] and EST miner [5] can be investigated more thoroughly as a part of such an exercise.

References

1. An implementation of the apriori algorithm. `http://www.ug.cs.usyd.edu.au/~abright/`.
2. GenBank Database. `http://www.ncbi.nlm.nih.gov/Genbank/`.
3. Gene Ontology Project. `http://www.geneontology.org/index.shtml`.
4. GEPIS (Gene Expression Profiling in silico). `http://www.cgl.ucsf.edu/Research/genentech/gepis/gepis.html`.
5. Sorghum EST Clustering Analysis. `http://cggc.agtec.uga.edu/estMiner/estMiner.jsp`.
6. UniGene Database. `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene`.
7. ZFIN: The Zebrafish Information Network. `http://www.zfin.org`.
8. A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. J. Mooney. Model-based overlapping clustering. In *KDD*, pages 532–537, 2005.
9. F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In *KDD*, pages 436–442, 2002.
10. J. Han and M. Kamber. *Data Mining: Concepts and Techniques*, pages 440–444. Morgan Kaufmann Publishers, second edition, 2006.
11. Y. Li and S. M. Chung. Text document clustering based on frequent word sequences. In *CIKM*, pages 293–294, 2005.
12. H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *SIGMOD Conference*, pages 394–405, 2002.
13. X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: a profile-based approach. In *KDD*, pages 314–323, 2005.

A Practical Method for Approximate Subsequence Search in DNA Databases

Jung-Im Won[1], Sang-Kyoon Hong[2], Jee-Hee Yoon[2], Sanghyun Park[3], and Sang-Wook Kim[1]

[1] College of Information and Communications
Hanyang University, Korea
{jiwon,wook}@hanyang.ac.kr
[2] Division of Information Engineering and Telecommunications
Hallym University, Korea
{kyoons,jhyoon}@hallym.ac.kr
[3] Department of Computer Science
Yonsei University, Korea
sanghyun@cs.yonsei.ac.kr

Abstract. In this paper, we propose an accurate and efficient method for approximate subsequence search in large DNA databases. The proposed method basically adopts a *binary trie* as its primary structure and stores all the window subsequences extracted from a DNA sequence. For approximate subsequence search, it traverses the binary trie in a breadth-first fashion and retrieves all the matched subsequences from the traversed path within the trie by a dynamic programming technique. However, the proposed method stores only window subsequences of the pre-determined length, and thus suffers from large post-processing time in case of long query sequences. To overcome this problem, we divide a query sequence into shorter pieces, perform searching for those subsequences, and then merge their results.

Keywords: DNA database, approximate subsequence search, suffix tree.

1 Introduction

Since the size of DNA databases is increasing considerably in these days, methods of fast indexing and query processing are essential for efficient DNA subsequence search. The *suffix tree* [4] has been known to be a good index structure for DNA subsequence search. Recently, there have been many research efforts on efficient construction and query processing with suffix trees [5] [10] [4]. The suffix tree still has the following drawbacks due to its structural features [3] [4] [11]: (1) high storage overhead, (2) poor locality in disk accesses, and (3) difficulty in seamless integration with DBMS.

In this paper, we propose a novel index structure that supports DNA subsequence search efficiently and also resolves the drawbacks of the suffix tree

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 921–931, 2007.

mentioned above. The proposed index structure basically adopts a trie [4] as its primary conceptual structure and realizes the trie by pointerless binary bit-string representation [13]. It extracts subsequences of the pre-determined length from every possible position of a DNA sequence, and stores only those subsequences in the index. They are called *window subsequences* and their length is usually much smaller than the average length of all the suffixes within a DNA sequence. This method is devised based on the observation that the length of longest common prefixes among suffixes in a DNA sequence is fairly small.

The DNA subsequence search with the proposed method uses the dynamic programming technique [4] and finds all the similar subsequences that exist on the paths of a binary trie. By traversing the trie index in a breadth-first fashion, it accesses each related page within the index only once. However, the proposed method stores only the window subsequences of the pre-determined length, and thus suffers from large post-processing time in case of long query sequences. To overcome this problem, we divide such a long query sequence into shorter ones, and then perform subsequence search for each of them. This alleviates the problem of performance degradation even with long query sequences.

2 Related Work

The performance of DNA subsequence search can be improved by exploiting indexing mechanisms. The methods proposed in references [1] [2] [12] employ the inverted index, which has been frequently applied in the area of information retrieval. They extract *words*, fixed length intervals overlapped with one another, from every sequence, and build a posting list of $<sequence\ number, offset>$ for each word. The method proposed in reference [6] maps every subsequence into a point in multidimensional space by the wavelet transform, and then constructs a multidimensional index on those points. By using the index, it processes range queries and nearest neighbor queries. This method enjoys nice search performance owing to a relatively small size of the index.

The suffix tree [4] is an index in a form of a persistent tree, and has been widely used in DNA subsequence search. Previously, it is not easy to construct a disk-resident suffix tree whose size is larger than that of main memory. Recently, reference [5] proposed a method for suffix tree construction by using the concept of partitioned suffix trees. Also, reference [10] proposed a top-down disk-based approach for efficient construction of suffix trees. Reference [7] proposed an approach for similar subsequence search that returns the results in the similarity based order by using dynamic programming and the A*-algorithm. However, the performance of approximate subsequence search with the suffix trees deteriorates as the length of a query sequence or a tolerance increases. A query partitioning method was proposed to solve this problem [8]. It partitions a given long sequence into shorter ones, and performs subsequence search for each of them with a smaller tolerance. Then, it merges the results thus obtained from all the subsequence searches.

3 Indexing Method

The suffix tree, which is a compressed digital trie built on all the suffixes of given sequences, has been known to be a good index structure for DNA subsequence search [4]. The suffix tree can compress input data sequences substantially when they have a lot of common prefixes. A DNA sequence can be considered as a string from the alphabet $\sum = \{A, C, G, T\}$. Since the size of the alphabet is very small (which is 4), it is likely that there exist a considerable number of common prefixes in the suffixes of input sequences. However, *longest common prefixes*(LCP) in the suffixes extracted from DNA sequences are commonly very short.

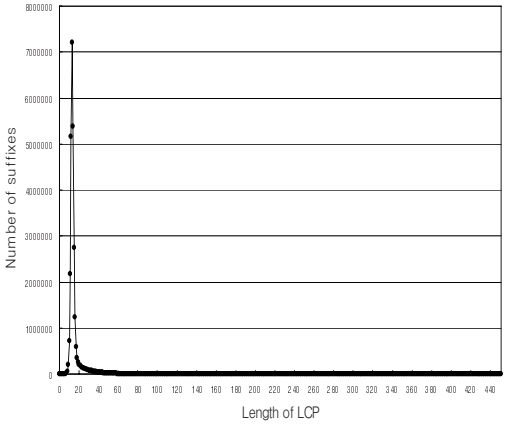


Fig. 1. Distribution of LCPs lengths

Symbol	Binary Code
$	000
A	001
C	010
G	011
N	100
T	101
S	110
Y	111

Fig. 2. Binary codes for symbols in the alphabet

Subsequence	Binary Representation
ACGA	001010011001
CGAC	010011001010
GACT	011001010101
ACT$	001010101000
CT$$	010101000000
T$$$	101000000000

Fig. 3. Binary representation of all the window subsequences from $S =$ 'ACGACT'

Figure 1 shows distribution of lengths of LCPs in suffixes extracted from a DNA sequence. We have used a 28.6Mbp DNA sequence in human chromosome 21 for the analysis. We have observed that the average and maximum lengths of LCPs in suffixes are 15 and 451, respectively, and that the number of suffixes that share LCPs whose length is 13 is largest(about 7.2 millions). Also, most suffixes share LCPs of a length 11 to 15, and 82.6% of LCPs have a length 1 to 15.

Based on this observation, we build an index not on all the suffixes extracted from a DNA sequence but only on their prefixes with a pre-determined length. That is, we place a sliding window of the length $|W|$ at every possible position of a DNA sequence, extract the subsequences covered by all the windows from a DNA sequence, and then insert them into the trie. We call these subsequences *window subsequences*. From our LCP analysis, we set the length $|W|$ as 15. We extract $|S|$ window subsequences from a DNA sequence S. The indexing with such window subsequences contributes to decrease the index size significantly and also makes the search of a leaf node simplified.

To represent all the symbols in the alphabet, we use the minimum number of bits instead of using one byte, thereby achieving high compression ratio. Figure 2 shows binary codes to represent all the symbols in DNA sequences. Here, N, S, and Y denote wild-card characters [12] and '$' denotes a special character used for padding to make all the window subsequences have a length of $|W|$. Given sequence S = 'ACGACT', we extract window subsequences whose length is 4 from S, allocate 3 bits for each symbol as shown in Figure 2, and represent each window subsequence into a corresponding binary bit-string as shown in Figure 3.

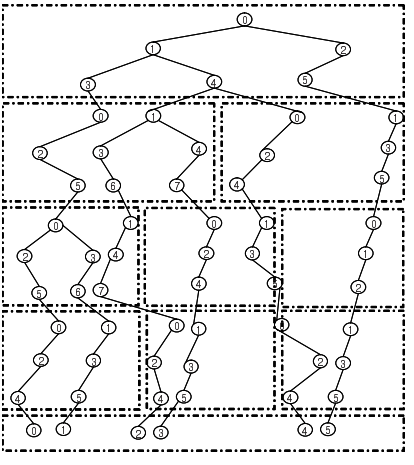


Fig. 4. Binary trie constructed from the window subsequences in Figure 3

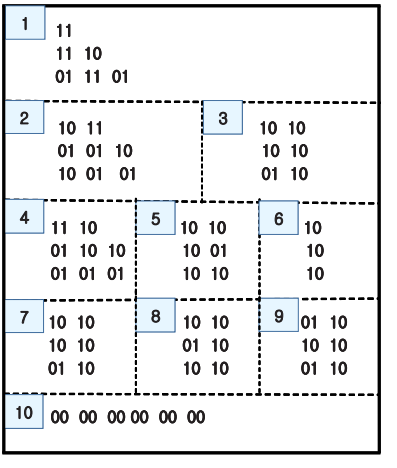


Fig. 5. Internal representation of the binary trie in Figure 4

In earlier work [13], we proposed a disk based index structure for efficient DNA sequence matching, exploiting the basic concept of pointerless binary tries. Pointerless binary tries require an alphabet to have only two symbols of 0 and 1. This makes every node have at most two outgoing edges. In this representation, the symbols on edges do not need to be stored explicitly if the following rules are enforced: (1) the outgoing edge labeled with 0 is assumed to connect to the left child node, and (2) the outgoing edge labeled with 1 is assumed to connect to the right child node. Our index structure basically adopts the binary trie as its primary conceptual structure. It consists of three parts: a binary trie, a page table, and a leaf table. The binary trie is an index structure storing all the window subsequences extracted from a DNA sequence. The page table stores the link information for pages within the binary trie. The leaf table stores the starting offsets of window subsequences within a DNA sequence. Figure 4 shows a binary trie constructed from the window subsequences of Figure 3. Here, node numbers in the trie are determined by the order of nodes being written into a disk page. Figure 5 shows its internal representation. The node structure is represented by a two-bit number and then is written into an appropriate page. In Figures 4 and 5, the rectangles of dotted lines represent pages stored in disk.

4 Query Processing Method

To discover all the subsequences similar to a query sequence Q, most similarity search algorithms [5] [7] [8] [4] based on a suffix tree traverse the tree in a depth-first fashion and, during the traversal, they build a dynamic programming (DP) table [4] using Q as its Y-axis and the sequence on the path from the root to the node being visited currently as its X-axis. We could apply such similarity search algorithms to binary tries. However, the proposed trie index contains only a two-bit number of each node. As a result, pointers from parents to their children, node levels, and subsequences on the paths from the root can not be extracted directly from the proposed trie index. Therefore, we need to uncover this implicit information whenever reading a new page during the traversal of the proposed binary trie. In addition, our binary trie is a disk-based index structure where the nodes on the same level are stored consecutively within a disk page. As a consequence, when we traverse all paths of the binary trie, we may access the same node multiple times within a single page and/or the same disk page more than once.

To solve the problem of accessing the same nodes and/or same disk pages multiple times, we propose to traverse the binary trie in a breadth-first order. That is, by visiting the nodes of the binary trie in a breadth-first fashion, our proposed Search-Trie() shown in Algorithm 1 effectively finds all the subsequences whose edit distances to a query sequence Q are not larger than a distance tolerance T.

Let us explain briefly how Search-Trie() operates. The algorithm employs two queues, Qpagenumber for examining data pages sequentially and Qnode for visiting the trie nodes of a current page one by one. The whole algorithm consists of two 'while' loops, an outer loop for data pages and an inner loop for trie nodes of a current page. For each child node CN_i of a current node $current\_Node$, we execute the following steps (Lines 7-19). First of all, we assign TRUE to variable $moreVisit$ which indicates whether or not we need to traverse the index further downwards (Line 8). Function AppendBitString() creates CN_i_Path, the path from the root to node CN_i, by extending the path from the root to node $current\_Node$ into node CN_i (Line 9). If the length of CN_i_Path becomes a multiple of 3, we compose a new symbol by aggregating the last 3 bits of CP_i_Path and then call function AddColumn() (Line 11). Function AddColumn() adds a column for the new symbol to the DP table constructed so far (i.e., $current\_DPT$), which results in a new DP table DPT_CN_i.

Let $dist$ be the value at the last row of the last column of DP table DPT_CN_i (Line 12). If $dist$ is not larger than distance tolerance T, all the subsequences containing the sequence on path CN_i_Path as their prefixes should be included in an answer set. Therefore, in such a case, we call function FindAnswers() where all leaf nodes under node CN_i are retrieved with their sequence and offset information. After that, we assign FALSE to variable $moreVisit$ in order to indicate more extension of path CN_i_Path is unnecessary (Line 15).

On the contrary, if $dist$ is larger than distance tolerance T, we call function FurtherVisit() which determines whether or not we have to go down under CN_i. Lines 18 and 19 are executed only when variable $moreVisit$ is TRUE. If node

CN_i is a leaf, we cannot decide if CN_i is an actual answer and thus should perform the post-processing of CN_i by executing function FindCandidateAnswer() (Line 18). This step is necessary for processing query sequences longer than window subsequence W. If node CN_i is not a leaf, we push node CN_i onto Qnode by calling function CheckpageAndPush() and continue the execution of the algorithm. Note that, if the number of nodes already processed within a current page reaches the maximum number of nodes (i.e., $maxNode$) that can be stored within a single page, we locate the next data page by looking up page table P and then push it onto Qpagenumber and its root node onto Qnode.

Since the binary trie has been built from a set of window subsequences of a fixed length, function FindCandidateAnswer() has to be executed when query sequences are longer than the window subsequences. However, in most cases,

Algorithm 1. Query processing algorithm Search-Trie

Input : binary trie I, query sequence Q, tolerance T, page table P, leaf table L, maxNode M
Output: set of answers answerSet

```
1  push(Qpagenumber, Root_pageNumber);
2  push(Qnode[Root_pageNumber], RootNode);
3  while notEmpty(Qpagenumber) do
4      pageNumber = pop(Qpagenumber);
5      while notEmpty(Qnode[pageNumber]) do
6          current_Node = pop(Qnode[pageNumber]);
7          for each child node CN_i of the current_Node do
8              moreVisit = TRUE;
9              AppendBitString(CN_i_Path, current_Node, CN_i);
10             if BitCount(CN_i_Path) mod 3 == 0 then
11                 DPT_CN_i = AddColumn(current_DPT, CN_i_Path);
12                 Let dist be the last row value of the new added column;
13                 if dist <= T then
14                     answerSet = answerSet ⋃ FindAnswer(CN_i, L);
15                     moreVisit = FALSE;
16                 else
                       moreVisit = FurtherVisit(DPT_CN_i);
17             if moreVisit then
18                 if terminal_Node(CN_i) then
                       answerSet = answerSet ⋃ FindCandidateAnswer(CN_i, L);
19                 else
                       CheckpageAndPush(Qpagenumber, Qnode, CN_i, P, M);
```

the number of candidate answers grows quickly as $|Q| - |W|$ becomes larger. In this paper, we propose to use a partition-based query processing [8] which circumvents this situation by decomposing a long query sequence into multiple pieces and then treating each piece as a separate query.

The proposed partition-based query processing algorithm is shown in Algorithm 2. Function Search-Trie-By-SubQuery() partitions a query sequence Q into p subqueries of appropriate lengths (Line 1). The number of subqueries and the length of each subquery are determined by considering how the performance of function Search-Trie() changes with respect to the length of a query sequence. For each subquery SQ_i obtained in the previous step, we perform the similarity-based searching by calling function Search-Trie() of Algorithm 1 (Lines 2-3). Note that the distance tolerance of each subquery is adjusted to $\lfloor T/p \rfloor$. At last, we construct a final answer set after executing function postProcessing() with a set of candidate answers *candidateSet* (Line 4). When offset i is given as a candidate answer, the post-processing step retrieves the corresponding data subsequence $S[i - |Q| - T, ..., i + |Q| + T]$ and computes its distance to Q using dynamic programming.

Algorithm 2. Query processing algorithm Search-Trie-By-SubQuery

Input : binary trie I, query sequence Q, tolerance T, page table P, leaf table L, maxNode M
Output: set of answers answerSet

1 p = partitionQuery(Q, T);
2 **for** *each subquery* SQ_i **do**
3 candidateSet = candidateSet $\bigcup$ Search-Trie(I, SQ_i, $\lfloor T/p \rfloor$, P, L, M);
4 answerSet = postProcessing(*candidateSet*, Q, T);
5 return answerSet;

5 Performance Evaluation

In this section, we show the effectiveness of our approach via performance evaluation with extensive experiments. We compared the performances of the three approaches Search-Trie, Suffix, and SW: (1) Search-Trie represents our approach that employs the pointerless binary trie as an index structure. Note that the window size is 15 (i.e., $|W| = 15$). (2) Suffix is an existing approach based on the suffix tree. We implemented the suffix tree by utilizing the source code of the TDD (Top-Down Disk-Based) technique [10] downloaded from http://www.eecs.umich.edu/tdd. (3) SW is the Smith-Waterman algorithm [9] based on dynamic programming. As a data set, we used two *Homo sapiens chromosome sequences*, chromosome 21 (chr 21) of 28.6 Mbps and chromosome 19 (chr 19) of 56Mbps. The hardware platform is the Pentium IV 3.2GHz PC equipped with 1 Gbytes main-memory. The software platform is Redhat Linux 9 (Kernel Version 2.4.20).

Data Size	Suffix				Search-Trie			
	Tree	Leaf_ Table	Rmost_ Table	Total	Binary_ Trie	Leaf_ Table	Page_ Table	Total
28.6Mbp (Chr 21)	267.6M	46.4M	46.4M	360.4M	50.2M	114.4M	0.5M	165.1M
56Mbp (Chr 19)	539M	91M	91M	721M	71.9M	224.1M	0.7M	296.7M

Fig. 6. Index sizes of the two approaches with increasing data set sizes

In Experiment 1, we compared Search-Trie with Suffix in the respect of an index size. Figure 6 summarizes the size of each index component of the two approaches with changing data sizes. From the experimental result, we observe that the index size increases linearly in proportion to the data size in both approaches. However, in comparison with Suffix, the proposed Search-Trie saves about 40% storage space.

In Experiment 2, we compared Search-Trie and Suffix in the respect of the elapsed time for approximate subsequence search. The total elapsed time is the time spent in finding all the subsequences whose edit distances to a query sequence are not larger than tolerance T. We also examined the total number of hits returned by Search-Trie and Suffix.

Query Length |Q|	Data Size = 28.6Mbp			Data Size = 56Mbp		
	Total hits	Query Processing Time(sec)		Total hits	Query Processing Time(sec)	
		Search-Trie	Suffix		Search-Trie	Suffix
10	4010	0.71	4.91	10641	0.8	7.22
20	137	2.03	12.81	2619	2.43	21.91
30	235	5.23	18.25	1815	9.48	35.42
40	77	30.96	28.08	848	47.32	48.08
50	52	182.33	36.26	712	279.67	63.16
60	82	868.85	57.80	376	1362.64	N/A

Fig. 7. Query processing times of the two approaches with increasing data set sizes and query lengths

Tolerance |T|	Data Size = 28.6Mbp			Data Size = 56Mbp		
	Total hits	Query Processing Time(sec)		Total hits	Query Processing Time(sec)	
		Search-Trie	Suffix		Search-Trie	Suffix
1	33	0.33	5.32	175	0.52	9.18
2	70	1.55	12.65	675	2.55	22.74
3	235	5.14	18.08	1815	9.51	35.28
4	490	23.73	26.58	3960	50.84	48.70
5	936	130.59	34.05	7614	300.66	61.84
6	1659	604.75	46.96	13596	1404.52	N/A

Fig. 8. Query processing times of the two approaches with increasing data set sizes and tolerance values

Figure 7 shows the elapsed times of approximate subsequence search by Suffix and Search-Trie with various query sequence lengths. The distance tolerance T is set to 10% of the length of the query sequence. Search-Trie outperforms Suffix when query sequences are not so long. Search-Trie runs 4 to 9 times faster than Suffix when query sequences are shorter than 40. However, the performance of Search-trie deteriorates as query sequences get longer. This is because Search-Trie generates many candidate answers, which result in much time being spent for post-processing when $|Q| \ll |W|$. Next, Figure 8 shows the elapsed times of

approximate subsequence search by Suffix and Search-Trie with various tolerance values. The query sequence length is fixed to 30 in this experiment. Search-Trie outperforms Suffix when tolerance values are not so large. Compared with Suffix, Search-Trie is about 4 to 17 times faster when tolerance values are less than 4. However, the performance of Search-trie deteriorates as tolerance values get larger. This results from the fact that the number of candidate answers increases as tolerance values become larger.

On the other hand, Suffix shows better performance for long query sequences or large tolerance values. For large DNA sequences, however, Suffix becomes impractical when query sequences are too long or tolerance values exceed a certain threshold. This is because, in such cases, Suffix has to traverse a huge index structure in a depth-first order and therefore needs to access a large number of index pages repeatedly. The experimental result reveals that Suffix on the data sequence of 56 Mbps (i.e., chr 19) cannot handle the cases of either ($|Q| = 60$ and $|T| = 6$) or ($|Q| = 30$ and $|T| = 6$).

In Experiment 3, we compared the performance of Seach-Trie-By-SubQuery with that of Search-Suffix-By-SubQuery. Seach-Trie-By-SubQuery denotes our approximate subsequence search approach based on Search-Trie and a query partitioning method with optimal p values. Search-Suffix-By-SubQuery denotes another approximate subsequence search approach based on Suffix, instead of Search-Trie, with the same query partitioning method. We also included the traditional algorithm SW in this experiment. When we apply the query partitioning method to Suffix or Search-Trie, we determine optimal p values by considering both the performance of index searching and the overhead of post-processing. To select optimal p values, we utilized results of Experiment 2.

Figure 9 shows the total elapsed times of the three approximate subsequence search approaches. The data sequences used in this experiment is chr 19 of 56Mbps. Also, the tolerance value T is set to 10% of the length of the query sequence. The query sequences are partitioned into subquery sequences of length

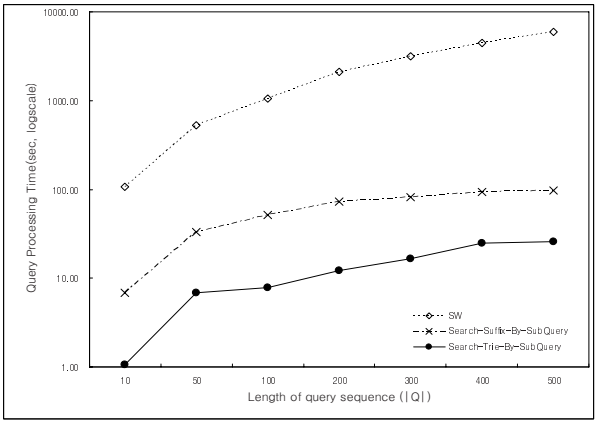


Fig. 9. Query processing times of the three approximate subsequence search approaches

25 in Seach-Trie-By-SubQuery. Also, the query sequences are properly partitioned into subquery sequences of length 20 or 40 in Seach-Suffix-By-SubQuery. According to our experimental results, we see that our method performs better than the other two methods and returns the answers very quickly even with large DNA data sequences. Search-Trie-By-SubQuery shows good performance regardless of the length of query sequences, and achieves 3 to 9 times speedup compared to Search-Suffix-By-SubQuery and 75 to 200 times speedup compared to SW.

6 Conclusions

In this paper, we have proposed an index structure and a query processing algorithm for approximate DNA subsequence search. The DNA subsequence search with the proposed index uses the dynamic programming technique, and finds all the similar subsequences stored on the paths of a binary trie. By traversing the trie index in a breadth-first fashion, it accesses just the pertinent pages within the index only once. In cases of a long sequence, it divides a query sequence into a set of shorter subsequences and retrieves actually similar subsequences by performing subsequence search for every shorter subsequence.

Acknowledgments. This work was supported by Korea Research Foundation Grant funded by Korea Government (MOEHRD, Basic Research Promotion Fund) (KRF-2005-206-D00015), by the Brain Korea 21 Project in 2007, by Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea Government(MOST) (No. R01-2006-000-11106-0), by the MIC of Korea under the ITRC support program supervised by the IITA(IITA-2005-C1090-0502-0009).

References

1. A. Califano and I. Rigoutso, "FLASH: A Fast Look-up Algorithm for String Homology", *Proc. Intelligent System Conference for Morecular Biology*, pp. 56-64, 1993.
2. C. Fondrat and P. Dessen, "A Rapid Access Motif database(RAMdb) with a search algorithm for the retrieval patterns in nucleic acids or proteun databanks", *Computer Applications in the Biosciences*, Vol. 11, No.3, pp. 273-279, 1995.
3. R. Giegerich, S. Kurtz, and J. Stoye, "Efficient Implementation of Lazy Suffix Trees", *Softw. Pract. Exp.*, Vol 33, pp. 1035-1049, 2003.
4. D.Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997.
5. E. Hunt, M. P. Atkinson and R. W. Irving, "Database indexing for large DNA and protein sequence collections", *VLDB Journal*, Vol. 11, No. 3, pp. 256-271, 2002.
6. T. Kahveci and A. K. Singh, "An Efficient Index Structure for String Databases", *Proc. VLDB Conference*, pp. 351-360, 2001.
7. C. Meek, J. M. Patel, and S. Kasetty, "OASIS: An Online and Accurate Technique for Local-Alignment Searches on Biological sequences", *Proc. VLDB Conference*, pp. 920-921, 2003.

8. G. Navarro and R. Baeza-Yates, "A Hybrid Indexing Method for Approximate String Matching", *Journal of Discrete ALgorithms*, Vol. 1, No. 1, pp.205-239, 2000.
9. T. Smith and M. Waterman, "Identification of Common Molecular Subsequences", *Journal of Molecular Biology* 147, pp. 195-197, 1981.
10. S. Tata, R. Hankins, and J. Patel, "Practical Suffix Tree Construction", *Proc. VLDB Conference*, pp. 36-47, 2004.
11. H. Wang et al., "BLAST++: A Tool for BLASTing Queries in Batches", *Proc. Asia-Pacific Bioinformatics Conference*, pp. 71-79, 2003.
12. H. E. Williams and J. Zobel, "Indexing and Retrieval for Genomic Databases", *IEEE TKDE*, Vol. 14, No. 1. pp. 63-78, 2002.
13. J. I. Won, J. H. Yoon, S. H. Park and S. W. Kim, "A Novel Indexing Method for Efficient Sequence Matching in Large DNA Database Environment", *Proc. PAKDD Conference*, pp. 203-215, 2005.

An Information Retrieval Model Based on Semantics

Chen Wu[1,2] and Quan Zhang[2]

[1] Graduate School of Chinese Academy of Sciences,100039 Beijing, China
[2] Dept. of NLP&SR Institute of Acoustics, CAS, 100080 Beijing, China
{Wuchen, Zhq}@mail.ioa.ac.cn

Abstract. Models of document indexing and document retrieval are mainly based on statistical NLP method. Computation of meaning is mainly based on the semantics. The former makes it possible to construct a high performance IR system easily. However, the latter is one of the significant methods which can substantially make computer well understand the language. The goal of this paper is to find the conjoined point which can combine the advantages of both schemes, and thus to propose an IR approach. Consequently, a concept-based IR model is proposed. This model is composed of two kernel schemes: the first is a domain language model, which is derived from the traditional language model. Its basic idea is to compute the conditional probability $P(Q \mid D)$. The concept extracting approach, which is the second kernel scheme of the proposed model, originates from the traditional linguistics. It can help to well extract the meaning of a term. Thus, we can take the concept (the formalized meaning), instead of the lexical term, as the processing object in the proposed model, and consequently resolve the word sense ambiguity. Experiments on the TREC6 Chinese collection show that the proposed model outperforms the traditional TF-IDF methods, especially in the average precision and the overall search time.

Keywords: Concept-based Information retrieval, Domain language model, Word concept, Sentence category, Clustering method.

1 Introduction

Most of the former or existing IR methods are based on the occurrences of terms in a document (or TF) and do not attempt to resolve the meaning of the terms. As we all know, a word may have a lot of senses. A sense can also be represented by more than one word. So, we doubted whether using the formalized meaning of the word, instead of the word itself, as the processing object can improve the accuracy of the IR system. Meanwhile, some research teams have fully studied the semantic space [1-3] and employed a formalized symbolic system to represent the space [2-3]. This symbolic system provide an important foundation for us to represent the word sense. On the other hand, the meaning of a word is implicated by the context. Therefore, the main task for us is to propose an approach which can accurately draw the meaning of a word (namely concept) according to the context based on the symbolic system.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 932–939, 2007.

After the concept can be successfully extracted, a suitable model for IR is required. Statistical model becomes the preferred approach since we are still not able to measure the semantic relevance of a query against a document within a collection using linguistic or even semantic method. A number of researches have, recently, confirmed that the language model is an effective and promising approach for IR [7, 8]. Therefore, we attempt to construct a suitable statistical model and apply varying degrees of NLU(Nature Language Understanding) scheme to the basic retrieval model. Consequently, we propose a domain language model. The main idea of a domain language model is derived from the Aspect Model [10].

2 Related Work

A formalized symbolic system which is used to express the meaning has been designed and constructed according to the needs of the NLP engineering recently [2-4]. It can be classified into two parts: the basal member sub-system and the sentence category sub-system.

The basal member sub-system is designed for describing the meaning of the terms. There are 108 concept trees within this sub-system. Each tree describes a category of the concepts. Each node of a tree, indexed by an exclusive character strings, represents a concept (namely, a meaning). The connotation of the child node represents a narrower sense than his father node, but more concrete. An algorithm for semantic relativity calculation based on the basal member sub-system is also addressed [9]. It can be used to measure the semantic relativity between two concepts.

Sentence category sub-system is designed for describing the meaning of sentences. The fundamental of this sub-system is using the finite expressions to express the meanings of the infinite sentences. These finite expressions are designed in advance. Huang concluded 57 types of primitive sentence category expressions and 57*56 compound ones [2, 3]. As a result, the meaning of sentences can be formalized.

The details of the symbolic system follow the definition described by MIAO [4].

Language modeling has been applied successfully in information retrieval [5, 6]. Given a document d and a query q, the basic principle of this approach is to compute the maximum conditional probability $P(D \mid Q)$ as follows.

$$\arg_D \max P(D \mid Q) = \arg_D \max P(Q \mid D)P(D)$$

If $Q = q_1 \wedge q_2 \wedge ... \wedge q_n$ and they are independent (This is a basic assumption). $P(Q \mid D) = \prod_{m=1}^{n} P(q_i \mid D)$. In order to assign a non-zero probability to the unseen words and to improve the maximum likelihood estimation, smoothing is involved. Most of the language models use the collection model as the reference model to interpolate or smooth the document model.

3 Word Concept Extracting Approach

The extracting approach is based on the semantic and linguistic relationships among the sentence category expressions, the semantic chunks and the words.

Three knowledge bases are involved in the approach. They are **TCK** (Term Concept Knowledge base), **SRK** (Scheduler Rule Knowledge base) and **SREK** (Semantic Relativity Knowledge base). These knowledge bases stored all the useful knowledge refined in advance by an assistant system [9]. They are the foundations of functions, $f()$ and $g()$.

TCK is the key knowledge base for the extracting approach. It is like a special vocabulary. It stores the terms and all their corresponding concept candidates. It also stores all the semantic and linguistic features of a term. These features will be the restrictions of selecting a concept candidate of a term in a certain context. The affiliation information between the terms and the sentence category expressions is also stored in TCK. It provides the information about which sentence category expression should be hypothesized and which character can be used to validate the hypothesis according to the given terms the computer reads. SRK determines which sentence category expression should be hypothesized first, which one second, according to the words in the sentence. SREK serves the concept relativity calculation. Relativity calculation can help to determine the term concepts within a local range. SREK was constructed based on the algorithm (called AlgZ)[7] for semantic relativity calculation. It is somewhat like the relation definition between two words in WordNet.

The details of the three knowledge bases are described in [9].

Processing strategy focuses mainly on the processing logic which tells computer how to get the sentence category expressions and the term concepts through some specific procedures. The whole processing strategy is an iterative procedure of $f()$ and $g()$. Due to the limitations of space, we only provide the block diagram here. The strategy is shown schematically in Fig. 1

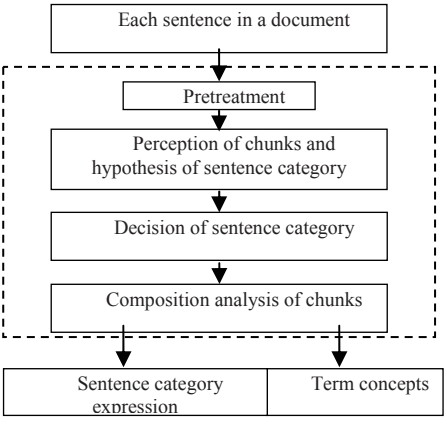


Fig. 1. The diagram of the processing strategy

The process can be divided into four sub-stages. The first sub-stage is called pretreatment. The assignments in this sub-stage depend on the language it processes. If Chinese is processed, the word segmentation will be performed. In the second sub-stage, the semantic chunks in the sentence and their corresponding sentence category

expressions will be hypothesized according to the terms in the sentence and their concepts recorded in CE of the TCK. The possible sentence category expressions are also stored in the SCE of the TCK. In order to hypothesize the chunks, the relativity calculation will be involved. It is used to derive the modifier of a headword or locate the juxtaposition of terms even without the use of coordinating or subordinating conjunctions. In the third sub-stage, the hypothesis made in the second sub-stage will be tested according to the "CC" in the TCK. The right sentence category expression will be proved and obtained. Composition analysis of chunks is the last sub-stage. Through this sub-stage, we can get all components of the chunks. Components are expressed by the exact term concepts. After processing, sentence category expression and the concept expression of each term in the sentence can be obtained.

4 Domain Language Model

The basic idea of the domain language model approach is to estimate the domain language model for a document and then to compute the likelihood that the query would have been generated from the estimated model. Therefore, the key issue is how to estimate the domain language model for a document based on the observation of a collection of documents. Consequently, the idea of Aspect Model [10] is introduced into the new model. The domain in domain language model is similar to the unobserved class variable in Aspect Model. As a result we can translate the domain language model into a joint probability model shown as following.

$$P(t \mid d) = \sum_{p} P(t \mid p) P(p \mid d) \tag{1}$$

$P(p \mid d)$ is the probability of d belonging to the domain p. $P(t \mid p)$ is the probability of domain p generating the concept t.

In order to obtain the unobserved domain variable p, clustering method is introduced into the domain language model. As a result, $P(t \mid p)$、$P(p \mid d)$ can be estimated based on the distribution of the domain-based clusters: $P(p \mid d)$ is estimated according to the appearance probability of the cluster p. $P(t \mid p)$ is estimated according to the frequency of the concept t in the cluster p.

In the proposed cluster generation method, K-Means is adopted for its ease of implementation. Profited from the concept trees defined in the basal member sub-system, the proposed method, different from the traditional method, assigned the data points more purposefully. According to the definition of the concept trees, 24 of the 108 concept trees have strong and discrete domain characters [3]. Therefore, we create 24 data points as domain seeds manually according to the concept trees. Each data point is a gathering of the characteristic concepts in the corresponding concept tree. As a result, initial domain-based clusters are assigned. After that, we use two stage K-means (which means the algorithm only alternates two times) to generate the clusters. When the clustering procedure is finished, we remove the seeds from the clusters.

The proposed method uses Kullback-Liebler distance algorithm to measure the correlation between a document and a cluster. The objective function follows.

$$KL(d,c) = \sum_{t_i \in d} \frac{n(t_i,d)}{|d|} \log \frac{n(t_i,d)/|d|}{n(t_i,c)/|c|} \tag{2}$$

where $KL(d,c)$ defines the degree of correlation between a document (or a collection of documents: cluster) d and a document (cluster) c; $n(t_i,x)$ is the document-concept weight, which is a measure of the number of occurrences of concept t_i in the document (cluster) x; $|x|$ is the number of concepts in the document (cluster) x.

The procedure for the clustering algorithm is simply described in Algorithm 1.

Algorithm 1 (Clustering algorithm)
Input: 24 clustering seeds and the document collection
Output: 24 clusters and the K-L distance between each document and each cluster
Procedure:
1. Translate the documents in the collection into their conceptual form according to the concept extracting approach described in Section 3, and then load them into the clustering candidate array $d_k \in D = \{d_1, d_2, \ldots, d_m\}$.
2. Assign 24 domain seeds (data points) to the 24 initial clusters. $p_k \in P = \{p_1, p_2, \ldots, p_{24}\}$.
3. **For** (i=1; i<=m; i++) **do**
4. **For** (j=1; j<= 24; j++) **do**
5. Calculate KL(d_i , p_j) defined as Eq. (2);
6. **End for**;
7. Assigned d_i to the cluster p_k whose KL(d_i , p_k) is minimum.
8. **End for**;
9. Repeat 3-8;
10. Calculate each KL(d_i , p_k) and save the values to a file, where $d_q \in D$ and $p_k \in P$

The proposed approach computes document query similarity in two stages.

In stage 1. All the documents will be indexed into the concepts. In stage 2, the query will be translated into their conceptual form.

To accomplish stage 1, we define the conditional probability $P(C \mid D)$ as the probability of using concept *C* as the domain for document *D*, which is given below.

$$P(C \mid D) = \prod_{t_j \in C} \frac{n(t_i, D) + \mu \sum_P P(t_i \mid p) P(p \mid D)}{|D| + \mu} \tag{3}$$

As can be seen from Eq. (3), the domain language model $\sum_P P(t_i \mid p) P(p \mid D)$ plays the role of collection estimates to compute the probability of a concept term.

$n(t_i, D)$ is a measure of the number of occurrences of concept t_i in document D. $|D|$ is the number of concepts in document *D*.

$P(p \mid D) = \frac{1 - kl(D,p)}{\sum_p 1 - kl(D,p)}$, where $kl(D,p)$ is the KL distance between document D and cluster p. The distance has been calculated in Algorithm 1 of section 4.2.2.

$P(w_{jn} \mid P_s)$ is a measure of the frequency of the key concept w_{jn} in cluster P_s. $P(w_{jn} \mid P_s) = \dfrac{n(w_{jn}, P_s)}{\mid P_s \mid}$.

In stage 2, the query should be translated into its conceptual form in order to match the conditional probability $P(C \mid D)$ in Eq. (3). The procedural steps for translation algorithms are simply described in Algorithm 2. This approach are based on the algorithm (called AlgZ) [10] for semantic relativity calculation.

Algorithm 2 (Query Translating)
Input: a set of query keys (query)
Output: a set of concepts and its reliability
Procedure:

1. Read the query keys into the candidate array $Q = \{q_1, q_2, ..., q_n\}$;
2. Select all the concept candidates c_i for q_i from TCK, and load them into a set of new arrays $c_i = \{c_{i1}, c_{i2}, ..., c_{ik}\}$, where, *1<=i<=n*, *k* is the number of the concept candidates of q_i;
3. **If** HasRelation(c_{ix}, c_{jy}) defined in AlgZ, where $c_{ix} \in c_i$, $i \neq j$ **then**
4. The relation value R(c_{ix}, c_{jy}) =1;
5. **Else** R(c_{ix}, c_{jy}) =0;
7. **End if**;
8. Get the optimized array $C = \{c_{1x}, c_{2y}, ..., c_{nz}\}$ which has the maximum sum of relation values between each other;
9. **If** more than one array has the same maximum sum (more than one *C* is returned) **then**
10. Compute $\prod_{j=1}^{n} P(c_{jk}, q_j)$ for each *C* as their reliability, $c_{jk} \in C$. Where $P(c_{jk}, q_j)$, obtained from TCK, is the probability of query key q_j translating to c_{jk}.
11. **For each** *C* **do**
12. Output C and its reliability;
13. **End for each**;
14. **Else**
15. Output C and set its reliability =1
16. **End if**.

From Algorithm 2, we can see that sometimes the output is not one set of concepts. In this situation, we compute the concept document similarity for each set of concepts, and then multiply this similarity scoring by the reliability value as the last scoring of the similarity.

5 Experimental Results

We have implemented a research prototype retrieval engine (called HNCIR) to test our approach. We now provide experimental results to illustrate the behavior of

HNCIR. The Chinese test collections are chosen from TREC6. It was 170 Mb as raw texts. There were 26 topics (CH 29-54) constructed.

The TF-IDF has shown its superior performance for document indexing [12]. Therefore, this scheme is used as a standard for comparison with HNCIR. Some modification was properly applied in order to enable it to implement the Chinese IR.

We also try to improve our probability estimates since this should yield better retrieval performance. We called this model HNCIR-X. This improvement of the estimate is to translate the queries into their conceptual form manually, and then input them as the new queries to the system. This can greatly help to find out the effect that the query translation brings us and the estimate precision of the query translation.

We measured the recall level precisions. The test results of these 3 approaches and the uninterpolated average precision over all relevant docs are given in Table 1.

Table 1. Recall Level Precision

Precision Recall	TF-IDF	HNCIR	HNCIR-X
0	0.8010	0.9217	0.9421
0.1	0.6021	0.8416	0.8431
0.2	0.5322	0.7624	0.7787
0.3	0.4642	0.6862	0.7099
0.4	0.3774	0.6515	0.6651
0.5	0.3213	0.5872	0.6100
0.6	0.3059	0.5661	0.5712
0.7	0.2292	0.4728	0.4973
0.8	0.1611	0.4021	0.4108
0.9	0.1041	0.2639	0.2895
1	0.0201	0.0541	0.0511
AvgPre	0.3321	0.5647	0.5769

The precision results show that the proposed system outperforms the TF-IDF method. HNCIR and HNCIR-X increased the precision of the traditional TF-IDF method by more than 20%. We can also see that the precision of HNCIR-X is similar to that of HNCIR. There are two possible conclusions can be made. One is that the sense ambiguities of the query keys are not serious. The second is that the AlgZ can well match the needs of the query translation. Considering all the test topics, we find the first reason effects more on producing such a result. It seems that the query translation does not contribute a lot to the improvement of the system performance in the case of the TREC6 test collection. Nevertheless, the AlgZ-based query translation has played an important role in CH37, CH41, CH42, Ch46 and CH47. It greatly helps the system to explicate the user's intentions through analyzing the semantic relationship between the query keys. Certainly, it costs some additional time to complete the estimation.

The cost of the computational time will be explicated in the following experiments. Table 3 shows the comparison between TF-IDF and HNCIR. It shows that the sub-total processing time for Chinese IR of HNCIR are more than 1.5 times than that of TF-IDF, but the retrieval time of the HNCIR are less longer. So we can conclude that on the time criterion, IR using HNCIR has advantages over IR using TF-IDF from the point of view of the information seekers, but has disadvantages from the

point of view of the document processing users (sometimes, the service providers). However, For a commercial system, the satisfaction of the searchers is the top goal of the service providers. Therefore, the overall time cost of HNCIR is not inferior to that of TF-IDF.

Table 2. Time for IR

	TF-IDF	HNCIR
Segmentation time (concept extracting time)	4 H 44 M	7 H 36 M
Indexing time	1 H 25 M	2H 08 M
Sub-total	6 H 09 M	9 H 44 M
Retrieval time	7 M	5 M

6 Conclusions

In this paper, we have proposed an information retrieval model. In this model, we tend to apply the NLU schemes to the SNLP (Statistical NLP) methods. The goal of addressing this issue is to study a new approach, which can take advantage of both the NLU and the SNLP, to better serve the IR. The experiments in which the proposed approach was compared with the traditional TF-IDF method highlighted the better performance of the proposed scheme, especially with regard to the average precision.

References

1. Schank R. Identification of conceptualizations underlying nature lauguage. In: Schank R, Colby K Eds. Computer Models of Thought and Language. San Francisco, CA: W H Freeman and Company. 1973
2. HUANG Zengyang. HNC (Hierarchical Network Concept) Theory. Beijing: Tsinghua University Press. 1998 (In Chinese)
3. HUANG Zengyang. Mathematics and physics symbol system of language in language concept space. Beijing: Ocean Press. 2004 (In Chinese)
4. Miao Chuanjiang. Guide of HNC (Hierarchical Network Concept) Theory. Beijing: Tsinghua University Press. 2005 (In Chinese)
5. J. Ponte and W. B. Croft (1998). A language modeling approach to information retrieval. *In Proceedings of SIGIR 1998*. pp. 275-281.
6. C. Zhai and J. Lafferty (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. *In Proceedings of SIGIR 2001*, pp. 334-342.
7. Linguistic Data Consortium. TDT pilot study corpus. Catalog no. LDC98T25, 1998.
8. ZHANG Yun-liang, ZHANG Quan. An Algorithm Based on HNC Theory for Semantic Relativity Calculation between Words. Application research of computers. 2006. (In Chinese with English abstract)
9. WEI Xiangfeng. The Software Platform for Expanded Sentence Category Analysis Based on the HNC Theory. Doctor's academic dissertation of IOA, CAS. Available: http://www.hncnlp.com/Abs/absEwxf.htm (In Chinese with English abstract)
10. Salton, G., & Buckley, C. Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 513–523. 1988

AttributeNets: An Incremental Learning Method for Interpretable Classification*

Hu Wu[1,2,**], Yongji Wang[1], and Xiaoyong Huai[1]

[1] Institute of Software, Chinese Academy of Sciences, Beijing 100080, China
[2] Graduate University of the Chinese Academy of Sciences, Beijing 100039, China
Phone: +86-10-62661660 ext 1009; Fax: +86-10-62661535,
wuhu@itechs.iscas.ac.cn

Abstract. Incremental learning is of more and more importance in real world data mining scenarios. Memory cost and adaptation cost are two major concerns of incremental learning algorithms. In this paper we provide a novel incremental learning method, AttributeNets, which is efficient both in memory utilization and updating cost of current hypothesis. AttributeNets is designed for addressing incremental classification problem. Instead of memorizing every detail of historical cases, the method only records statistical information of attribute values of learnt cases. For classification problem, AttributeNets could generate effective results interpretable to human beings.

1 Introduction

Incremental learning ability is vital to many real world machine learning problems [8]. The common characteristics of these problems are that either the training set is too large to learn in a batched fashion, or the training cases are available as a time sequence. We need machine learning methods updating their hypothesis only with the latest cases, i.e. in incremental fashion. Much work has been done to provide incremental learning ability for the classification problems.

While most powerful classification methods suffer from the problem that their results are hard to understand (e.g. neural networks, support vector machine), others give interpretable, but usually less effective results. Among the latter ones are decision tree, rule induction methods, several graph based methods and rough set based methods. Decision tree is a widely used structure for classification. Utgoff proposed three incremental decision tree induction algorithms: ID5 [5], ID5R [5], and ITI [6]; rule induction methods are also efficient solutions of classification tasks and have been extended to solve incremental learning problems [9]; Galois (Concept) lattices and several extensions are data structures based on Hasse graph [1, 3] and are widely used in incremental classification and association rule induction; Rough set based methods produce a decision table of a sequence of rules for classification [9].

* Support by National Nature Science Foundation of China(Grant Number 60372053).
** Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 940–947, 2007.

Recently, Enembreck proposed a data structure named Graph of Concepts (GC) [2] for incremental learning. GC is composed of several attribute layers each representing an attribute, and a classification layer representing the categories. The attribute layer is comprised of several attribute nodes mapping to the values of this attribute and the class layer is comprised of some classification nodes each mapping to a category. During the learning phase, it records all the cases by attaching the case sequence number to the corresponding node when the value of the attribute equals the node's value for each attribute layer and the classification node that the case belongs to. Then they used an entropy based method named ELA to utilize the information stored in GC for classification. ELA tags a label to the unlabeled case the same as the most similar case(s).

However, there are the following defects with these incremental methods:

a) Bad memory utilization: many algorithms need to record historical cases for updating. This limits the scalability of these methods (decision tree, rule induction, ELA, Galois lattice, rough set based methods)
b) Inefficiency of updating hypothesis (decision tree methods, rule induction, Galois lattice, rough set based methods)
c) Vulnerable to screwed data or noisy data (decision tree methods, Galois lattice, ELA)

To address these problems, we design a novel incremental learning algorithm which is based on the structure called AttributeNets. It outperforms most of incremental algorithms with our special concerns on the memory and adaptation computation costs, and the classification results are easy to understand.

The rest of this paper is structured as follows: in Section 2 we give the definition of AttributeNets; the learning algorithm based on AttributeNets is given in Section 3 while the classification algorithm is elaborated in Section 4; in Section 5, we give a case study to evaluate the performance of our method; finally, the conclusions and the future work are given in Section 6.

2 AttributeNets Structure

For each category, we construct an isomorphic structure named AttributeNet. With AttributeNets, we refer to the combination of these individual nets. Similar to GC, each AttributeNet is composed of several attribute layers comprising of attribute nodes (node for short). Likewise, each layer corresponds to a specific attribute of cases, and a node in the layer corresponds to a specific value of this attribute. However, there are two significant differences between GC and AttributeNet: first, AttributeNet does not have classification layer being that each AttributeNet simply refers to only one category; second, instead of attaching the case sequence number to each node, we only save the statistical information in AttributeNet. Each node keeps a counter (node degree) to record how many cases belong to this node; for any of two nodes, another counter (link degree) is kept recording how many cases belong to both nodes.

For explanation, we consider a simplified classification problem. There are three categories, and each case has 4 attributes that have the value of either 0 or 1. For each category, an AttributeNet is constructed, i.e. there are three isomorphic AttributeNets. One of them is illustrated in Fig. 1.

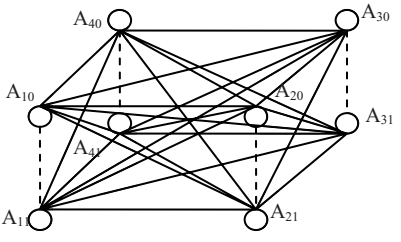


Fig. 1. A 4-layer AttributeNet

Table 1. Node value of the AttributeNet in Fig. 1

Layer	Node	Node Value
Layer1	A_{10}	0
	A_{11}	1
Layer2	A_{20}	0
	A_{21}	1
Layer3	A_{30}	0
	A_{31}	1
Layer4	A_{40}	0
	A_{41}	1

***Definition* 1** (Node)**.** Node is the basic unit in AttributeNet. A node represents a specific value (*node value*) of an attribute and keeps a counter (*node degree*) counting the number of cases that has this value for the specific attribute. In Fig. 1, $A_{ij}(1 \le i \le 4, 0 \le j \le 1)$ are all nodes. We say that a node A_{ij} is activated by a case if the *i*th attribute of the case has the node value of A_{ij}.

***Definition* 2** (Layer)**.** Each layer represents a specific attribute of the cases. So a layer is composed of several nodes representing the corresponding values of this attribute. In Fig. 1, $A_i(1 \le i \le 4)$ are layers, each layer is composed of two nodes: A_{i0} and A_{i1}.

***Definition* 3** (Node Link)**.** There are links between any two nodes of different layers. If a case belongs to both nodeand A_{ij} node A_{gf}, the *link degree* between these two nodes increases by 1. The initial link degree of any two nodes is 0. Note: the link degree between any two nodes of the same layer is always 0.

***Definition* 4** (AttributeNet and AttributeNets)**.** An AttributeNet is composed of several layers with each of which represents a specific attribute of cases. Each AttributeNet represents only one category in the classification problem. With AttributeNets, we refer to the combination of these nets.

3 AttributeNets Learning Algorithm

The learning process of AttributeNets is straightforward and efficient in time complexity which makes our method suitable for online learning.

AttributeNets memorizes statistic information of attribute values and relationships between any two of values of different attributes, with consideration of cases of only the net's own category.

Algorithm 1: (AttributeNets learning algorithm)

Input: AttributeNets ($Attr_i, 1 \le i \le \|Categories\|$) to be updated, new training case ($Case$)

Output: Updated AttributeNets

Step1: $i = categoryOf(Case)$

Step2: For $1 \le j \le \|Layers\|$

$node\_degree[j][k]++$ ($node\_degree[j][k]$ is the degree of the node$_{jk}$ which is one node of layer j of $Attr_i$ and is activated by $Case$)

Step3: For $1 \le j \le \|Layers\|$

For $1 \le u \le \|Layers\|$

$link\_degree[j][k][u][v]++$ ($link\_degree[j][k][u][v]$ is the degree of the node link between the activated nodes, i.e. node$_{jk}$ of layer j and node$_{uv}$ of layer u of $Attr_i$)

Step4: End □

When a training case of category i comes, AttributeNet$_i$ is activated while other nets other than category i are simply ignored by this case. With AttributeNet$_i$, for each attribute of the case, i.e. each layer of AttributeNet$_i$, we increase the degree of node if this attribute has the value identical to the node value. For any two nodes of different layers, we increase the link degree between these two nodes by 1 if both nodes are activated by the case.

Take the classification problem mentioned in Section 2 for example, in Table 2, there are 4 training cases of category 1, after training, the node degree and link degree of AttributeNet$_1$ are illustrated in Table 3, while values of AttributeNet$_i$ of category other than 1 are not changed by these cases.

Table 2. The training cases

No.	@1	@2	@3	@4
1	0	1	0	1
2	1	1	0	1
3	0	1	1	0
4	1	1	1	0

Table 3. Degree of nodes and links between nodes after training

	A_{10}	A_{11}	A_{20}	A_{21}	A_{30}	A_{31}	A_{40}	A_{41}
A_{10}	2	0	0	2	1	1	1	1
A_{11}	0	2	0	2	1	1	1	1
A_{20}	0	0	0	0	0	0	0	0
A_{21}	2	2	0	4	2	2	2	2
A_{30}	1	1	0	2	2	0	0	2
A_{31}	1	1	0	2	0	2	2	0
A_{40}	1	1	0	2	0	2	2	0
A_{41}	1	1	0	2	2	0	0	2

The AttributeNets is learnt case by case and the learning result is independent of the order in which cases are learnt. When new case comes, we only need to increase the node degree of the nodes and the link degree of node links it activates. The time and memory cost of learning process are $O(n^2)$, where n is the number of nodes of AttriuteNets.

4 AttributeNets Classification Algorithm

The learning process and the classification process could be intertwined in AttributeNets method. This ability is favorable in online learning scenario. In this section, a classification algorithm is given based on AttributeNets.

Algorithm 2: (AttributeNets classification algorithm)
Input: AttributeNets ($Attr_i, 1 \le i \le \|Categories\|$) been learnt, new case (*Case*) with its category unknown
Output: Category *c* of *Case*
Step1: For $1 \le i \le \|Categories\|$

$r_i = 1$

Step2: For $1 \le i \le \|Categories\|$

For $1 \le j \le \|Layers\|$

$r_i = r_i \times \sqrt{node\_degree[i][j] + \Delta}$ ($node\_degree[i][j]$ is the value of the activated node by *Case* in layer j of $Attr_i$, Δ is a small number preventing r_i to be 0)

Step3: For $1 \le i \le \|Categories\|$

For $1 \le j \le \|Layers\|$

For $1 \le k \le \|Layers\|$

$r_i = r_i \times (link\_degree[i][j][k] + \Delta)$

($link\_degree[i][j][k]$ is the value of the node link between the activated nodes of layer j and layer k of $Attr_i$, Δ is a small adjustment preventing r_i to be 0)

Step4: Return *i* which $Maximize(r_i)$ □

The time complexity and space complexity of algorithm 2 are both $O(m \times n^2)$, where m is the number of categories, n is the number of nodes in each AttriuteNet.

Moreover if the node degree of active nodes and the link degree of active links between two nodes of AttributeNets are investigated, through comparing these values from different nets, not only we could find out which category the case belongs to, but also could we find out which value is vital for the classification decision.

The classification result is interpretable to human because there exists an injection between layers of AttributeNets and attributes of cases.

5 Performance Evaluations

The performance of AttributeNets is a significant improvement of its counterparts. In this section we give the comparison results of AttributeNets and the related algorithms on the MONK-3 [10] classification benchmark set for the performance verification.

5.1 Performance and Robustness Evaluations of AttributeNets

MONK-3 problem is a widely used benchmark data set for classification algorithms evaluation. There are two categories denoted by 0 and 1, and each case has six attributes. The valid values of each attribute are listed in Table 4. The case which satisfies $(@1 = 3 \wedge @4 = 1) \vee (@5 \neq 4 \wedge @2 \neq 3)$ belongs to category 1; otherwise it belongs to category 0.

Table 4. Possible values of the attributes in MONK-3

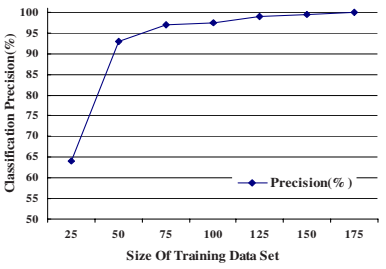
@ attribute1	{1,2, 3}	@ attribute2	{1,2, 3}	@ attribute3	{1,2}
@ attribute4	{1,2, 3}	@ attribute5	{1,2,3,4}	@ attribute6	{1,2}

For each category, an AttributeNet is constructed. Therefore, there are two nets representing category 0 and category 1, respectively. For training, 150 training cases are generated randomly, 5 percent of which are noisy cases, i.e. there are 8 mislabeled training cases. Then we generate randomly 100 test cases to be classified on three different platforms: AttributeNets, ELA, and ID5R [5]. The comparison results are shown in Table 5. AttributeNets outperforms other algorithms in both precision and time cost of learning and classification.

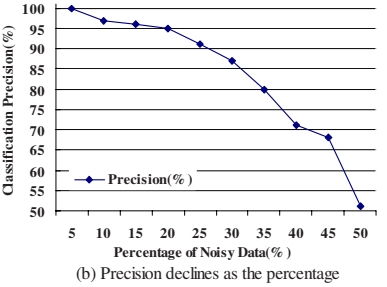
Table 5. Performance comparison of AttributeNets, ELA and Decision trees on MONK-3

	AttributeNets	ELA	Decision Tree(ID5R)
Precision (%)	99 ± 1	65 ± 10	92 ± 3
Learning Time(ms.)	16	15	157
Classification Time(ms.)	31	47	32

Also we carry out robustness tests on AttributeNets to see its performance in the cases of the noisy training data and the scarcity of training cases. The basic settings are the same as the above. First we increase the number of training cases from 25 to 175 in order to investigate the influence of training set size. Then noisy data of the percentage varying from 5 to 50 are mixed in the training set. The classification results are shown in Fig. 2(a) and (b), respectively.



(a) Precision climbs as the size of training data set grows



(b) Precision declines as the percentage of noisy data grows

Fig. 2. Robustness test of AttributeNets with varying size of training set and noisy data

We conclude that AttributeNets is robust with noisy data (as the percentage of noisy data runs up to 30%, the precision is still as high as 87%) and it works quite well with only a small size of training set available.

5.2 Performance Discussion

As Utgoff in [6] pointed out, there were 12 design principles that should be considered when designing an incremental learning classification system. We summarize them as the following:

1) The update cost of the method must be small
2) Input: the method should accept cases as input described by any mix of symbolic and numeric variables, sometimes continuous variables
3) Output: the method should be capable to handle multiple classes as well as two classes
4) Fault tolerance: the method should be strong enough to handle noisy data and inconsistent data
5) Capable of handling screwed data: the method should take the possibility that the data between categories are unbalanced into consideration
6) Capable of handling some problems with strong relationships among several attributes, like MONK-2 problem [10]

Our method satisfies principle 1, 3, 4, 5; partly satisfies principle 2 because we have not taken continuous attribute into account. The limitation of our method is that it only considers the relationships between any two attributes, therefore, if there are relationships between more than two attributes, like MONK-2, our method does not generate results as good as Neural Networks.

6 Conclusions and Future Work

Incremental learning algorithms provide new opportunities for industry whilst put forward new challenges to researchers: (1) how to memorize the knowledge been learnt for further updating without recording every case learnt before; (2) how to avoid (or

keep) order effects in which cases have been leant; (3) how to design fast updating algorithm; (4) how to make learning results interpretable to human. Trying to solve these problems, we have designed a new data structure (AttributeNets) and algorithms for incremental learning and classification. The advantages of our algorithm are in four folds:

1. It is in itself a multi-category classifier because of multi-nets structure
2. It is outstanding in memory utilization and adaptation speed which is of vital importance for incremental learning, specially online learning
3. The classification results are easy to understand
4. It is robust with the noisy data and the scarcity of training cases

Our future work includes: first, extensions could be made to enrich AttributeNets structure to improve classification precision; second, aside from the classification problems, AttributeNets could be naturally extended to induct association rules, which are also important data mining problems.

References

1. E. M. Nguifo, P. Njiwoua: IGLUE: A Lattice-based Constructive Induction System. In: Intelligent Data Analysis Journal, Vol. 5, No. 1, 2001, pp. 73-81.
2. F. Enembreck and J. P. Barths: ELA: A New Approach for Learning Agents. In: Journal of Autonomous Agents and Multi-Agent Systems, Vol. 3, No. 10, 2005, pp. 215-248.
3. Godin R.: Incremental Concept Formation Algorithm Based on Galois (Concept) Lattices. In: Computational Intelligence, Vol. 11, No. 2, 1995, pp. 246-267.
4. K. Hu, Y. Lu, C. Shi: Incremental discovering association rules: a concept lattice approach. In: Proceedings of the PAKDD-99, Beijing, 1999, pp. 109-113.
5. Utgoff P. E.: Incremental Induction of Decision Trees. In: Machine Learning, Vol. 4, 1989, pp. 161-186.
6. Utgoff P. E.: An Improved Algorithm for Incremental Induction of Decision Trees. In: Proceedings of the Eleventh International Conference of Machine Learning, 1994, pp. 318-325.
7. M. Maloof: Incremental Rule Learning with Partial Instance Memory for Changing Concepts. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN '03). Los Alamitos, CA, 2003, pp. 2764-2769.
8. S. Lange and G. Grieser: On the Power of Incremental Learning. In: Theory Computer Science, Vol. 288, No. 2, 2002, pp. 277-307.
9. Z. Zheng, G. Wang, Y. Wu: A Rough Set and Rule Tree Based Incremental Knowledge Acquisition Algorithm. In: LNAI2639, Springer-Verlag, 2003, pp. 122-129.
10. S. B. Thrun et al.: The MONK's Problems: A Performance Comparison of Different Learning Algorithms. Technical report. Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA, 1991.

Mining Personalization Interest and Navigation Patterns on Portal

Jing Wu[1], Pin Zhang[2], Zhang Xiong[3], and Hao Sheng[2]

[1] School of Computer Science, Beihang University
37th Xueyuan Road, Haidian District, Beijing, China, 100083
[1]wing.wujing@gmail.com, {jumpingzp, sh_buaa}@hotmail.com,
xiongz@buaa.edu.cn

Abstract. Personalization services pose new challenges to interest mining on Portal. Capturing the surfing behaviors of users implicitly and mining interest navigation patterns are the top demanding tasks. Based on the analysis of mapping the personalization interest behaviors on Portal, a novel Portal-independent mechanism of interest elicitation with privacy protection is proposed, which implements both the implicit extraction of diverse behaviors and their semantic analysis. Moreover, we present a hidden Markov model extension with personalization interest description of Portal to form interest navigation patterns for different users. Then experiments have been carried out in order to validate the proposed approaches.

Keywords: Portal, interest behavior, implicit interest elicitation, hidden Markov Model (HMM), navigation patterns.

1 Introduction

With the upcoming era of Web2.0, the resource integration and personalization technologies on Portal platform are becoming well-developed. Portal offers many powerful functions to customize desktops for users, so as to enable the user interests more various. Consequently, discovering and extracting the personalization interests from Portal are the essential tasks performed with understanding the preferences and access patterns. The implicit interest elicitation method can analyze trace data from log histories to find the interest features and relevant degrees, reducing the noise due to user participation. So it is significant in practice with the benefits to interest representation and well modeling, especially to user privacy protection. Various implicit schemes have been proposed in the related literature, e.g., based on intelligent agent [1], frequent user traversal paths discovery[1-4], site organization learning[5] and re-classification[6]. However, these traditional approaches with the restriction of the behaviors extraction cannot address the issues of personalization interests mining in Portal.

Furthermore, the existing web usage mining (WUM) techniques[1-4,6] lack the explanation on visit intentions and interest navigation trends. In general, navigation patterns can assist the designers of web site to understand the user access

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 948–955, 2007.

characteristics, adjust the structure reassembling and carry out the advertisements. As well, it is a critical issue for the Portal organizers to improve the personalization service quality by mining interest navigation patterns adaptively. Shi[7] presents a novel method to deal with interest navigation problems based on the hidden Markov model in order to discover users' interest navigation patterns, which seems some special association rules essentially and can be computed by an incremental algorithm. This approach is mainly processing uniform interest results of all users but lack of personalization descriptions, so it also leads to the need for its statement complement in Portal interest mining.

In this paper, we explore the mapping of the personalization interest behaviors on Portal. Then we briefly present a novel Portal-independent mechanism of interest elicitation supporting privacy protection, with solutions to implement implicit extraction of diverse access behaviors and corresponding semantic analysis. Emphasizing the association effect of interest weights and the prediction on interest intentions, we extend the hidden Markov model with personalization interest description of Portal forming interest navigation patterns for different users.

2 Mapping of the Personalization Interests on Portal

2.1 Basic Definitions

In this section, some definitions and assumptions are given as follows:

Definition 1. Each interest content on Portal points the classified concept object which the user interests in or accesses on the personalization desktop. Let $IC=\{\{Portlet_1,\cdots Portlet_l\},\{Link_1,\cdots Link_m\},\{Tab_1,\cdots Tab_n\}\}$ be the finite *Interest Content Set* (abbr. *IC*). Given $IC=\bigcup_{x=1}^{c} Cluster_x \left(1\le c\le l+m+n\right)$, where $Cluster_x$ represents one kind of interest content clustering results, while it can be expressed by $\sum=\{\sigma_1,\cdots,\sigma_x\}$, where σ_z denotes the feature of corresponding cluster.

Definition 2. Each interest behavior on Portal indicates certain potential typical visiting operation on *IC*. Let *IB={SetMode, SetRecomTime, Switch, Maximize, Minimize, Close, Click, Layout, Add, Delete, Edit, Quote, Comment}* be the finite *Interest Behavior Set* (abbr. *IB*). These interest behaviors may be classified into four categories shown in Table 1.

Definition 3. Given an *access transaction* (abbr. *at*) representing a user's interest behaviors on different interest contents occur during every session *T*, each *at* is defined as $at=\langle at.user,\cdots\{at.content_k,at.time_k,at.behavior_k\}\cdots\rangle\left(at.time_k-at.time_{k-1}\le T\right)$.Every user's *at*s can be gathered into the *Access Transaction Set* (abbr. AT_u) with temporal order as $AT_u=\{at_i\mid 1\le i\le |AT_u|\}$, where $|AT_u|$ denotes the total number of session *T* forming AT_u.

2.2 Analysis of Mapping Interest Behaviors

We put forward the detailed analysis of the operation semantic features of interest behaviors in Table 1. The listed mapping relationship offers a novel reference for eliciting the interest granularity and holding the priority. Besides, the noises caused by more details might be avoided considerably.

Table 1. Mapping of the personalization interest behaviors on Portal

Personalization interest behaviors on Portal		Requests Attribute (-Value)	Similar interest behaviors in Web site and corresponding interest degree		Effect factors	Interest degree
Custom configuration	SetMode	_mode	/	/	The state of the window	④or②
	SetRecomTime	_event	/	/	Frequency	④
Browse-click	Switch	_pageLabel	Forward / Back	④	Access transaction Set	④
	Maximize	_state-maximized	Open in a new window/ Drag stroll bar	⑤	The state of the window	⑤
	Minimize	_state-minimized	Exit	②		②
	Close	_state- closed				
	Click	_URL	Click hyperlink	④	Access transaction Set	④
Layout	Layout	_windowLabel	/	/	Layout sequence & Access transaction Set	⑤or②
	Add	_windowLabel	Add the bookmark	⑤		⑤
	Delete	_windowLabel	Delete the bookmark	①		①
Edit-comment	Edit	_mode-edit	Query	⑤	Access transaction Set	⑤
	Quote	_event	/	/		④
	Comment	_event	Feedback rate	④		④

Note: ①Strong negative; ②Negative; ③Weak positive; ④Positive; ⑤Strong positive

To represent these interest behaviors on Portal more practically, we introduce fuzzy logic [9,10] to incorporate the context of feature factors in Table 1.

Definition 4. Let $FS_{at}=Relation(AT_u, IC \cup IB)$ be a fuzzy relation set of access transaction on domain $AT_u \times (IC \cup IB)$. The membership weight $W_{at}^{i}(content_k) \in [0,1]$, on each item of AT_u, can be given by:

$$W_{at}^{i}(content_k)=\begin{cases}\left(\frac{2w_t}{W_T}-1\right)\times R/5 & 1/2 W_T \le w_t \le W_T, R<3\\ \frac{2w_t}{W_T}-1 & 1/2 W_T \le w_t \le W_T, 3 \le R \le 5\\ 1 & w_t > W_T, 3 \le R \le 5\\ 0 & otherwise\end{cases} \quad w_t=\frac{d(at_i.content_k)}{T_i} \quad W_T=\frac{\sum_{j=1}^{|AT_u|} d(at_j.content_k)}{\sum_{j=1}^{n} T_j} \quad R=\frac{\sum_{x\in at_i} r_x}{|at_i|} \tag{1}$$

With the entire accessing duration $d(at_i.content_k)$, w_t and W_T indicate a user's local and global interests respectively. Where $r_x \in [1,5]$ is the interest degree of each accessing interest behavior ranked as Table 1.

Definition 5. Let $S_u=\{at_i.sequence\}$ be the layout sequence set about interest contents (i.e., Portlets) on personalization desktop, where each *at.sequence* records every

updated appearance with left-right top-bottom order. Let FS_L=*Relation*(S_u, $IC \cup IB$)be a fuzzy relation set of layout sequence on domain $S_u \times (IC \cup IB)$. The membership weight $W_L(content_k)$, on each item of AT_u, can be defined as:

$$W_L(content_k) = \frac{|AT_u|}{\sum_{i=1, P_k \in S_u}^{|AT_u|} P_k^i} \tag{2}$$

Where P_k^i denotes the relative forward or backward transition offset in S_u.

Consequently, the particular interest weight of each user combining the effect of personalization interest behaviors on Portal can be denoted as follows:

$$W(content_k) = \sum_{i=1}^{|AT_u|} \sum_{content_k \in IC} W_{at}^i(content_k) W_L(content_k) \tag{3}$$

3 Implicit Interest Elicitation Mechanism

We observe that custom configuration and layout behaviors could hardly be logged by Portal server essentially, even no more entries within the captured records about browse-click and edit-comment yet. Based on the preceding statement, a novel Portal-independent mechanism of interest elicitation is proposed, which implements both the implicit extraction of diverse access behaviors and their semantic analysis.

The following strategy is taken in the approaches according to the mentioned four categories of interest behaviors: (i)**Custom configuration**: We use open API of Portal platform to gather the target User Profile (UP). The valid data we observe are those predefined parameter pairs (i.e., constructed in Attribute-Value format) in the UP, which have represented the semantic information and without additional semantic analysis. To cope with the general-purpose goal, we develop a new adapter used to effect operative compatibility among different particular Portals' interfaces[8]. (ii)**Browse-click**: We discuss based on the different possible operation objects. For the Portlets integrated with WSRP, to analyze the interaction requests at WSRP producer. For the normal web application requests, to capture the key requests using traditional means in WUM. And for the particular Portal applications, to call the special API to identify the behaviors related with the responses to these application events. Here, the captured results may be preprocessed into AT_u. (iii)**Layout**: We use the similar method with **(i)** to capture user's layout behaviors. Note that the captured results are the appropriate layout structures of different personalization desktops, which may contribute to form the S_u. (iv)**Edit-comment**: The method is similar as that mentioned in (ii).

We summarize the feature relationship of interest behavior semantics in Table 1 and build a mapping list in XML format. There involves two steps: Filtering the redundancy requests, and matching the valid parameter features with the Attribute-Value. The elicitation entry is defined as <*user,timestamp,object,behavior,desktop*>.

During the implicit interest elicitation process, there requires supplementing legal and technical mechanism for access control as well as a balance on the relationship between personalization and privacy. Based on the idea in transportation safety administration[11], we adopt recently released National Information Exchange Model

(NIEM) to establish the accountable representation logic. We use the Notation 3[12] to convert the transactional data into serializing RDF, which contains the representation of interest-extended rule. This is a short statute for the processing of matching the request parameters, which supports the validity determination and centralized storage of private data in privacy protection. By such means, we can provide the legal transparent accountable interests in terms of the interest behavior entries implied with the rule.

4 Mining the Interest Navigation Patterns

The navigation relationship among the interests can describe the user's next possible interest trends, and especially benefit the recommendation schema improvement in order of precedence whenever the prediction or resources presentation is proceeding. We attempt to seek the obtained AT_us to represent the navigation patterns on particular interest contents. Therefore, the presented hidden Markov model extension differs from [7] mainly in two aspects. First, the mapping of personalization interests on Portal is complemented. Second, the formula process is according to every particular user's personalization description instead of the entire user group. The HMM extension with the personalization interest of Portal is proposed as follows:

(i) Let the interest content node on personal desktop denote as the state node $q_i \in IC$ in HMM. Given the virtual initial state q_1, there is the relative interest concept mapping relationship as $q_i \mapsto \sigma_z^i \in \Sigma$. And there must exist the transfer probability distribution $P(q_i \to q_j)$ between any two nodes of AT_u:

$$P(q_i \to q_j) = P(q_j \mid q_i) = \frac{P(q_i q_j)}{P(q_i)} = \frac{\sum_{k=1}^{|AT_u|} \sum_{q_i \in IC} W_{at}^k(q_i) W_L(q_i) + \sum_{k=1}^{|AT_u|} \sum_{q_j \in IC} W_{at}^k(q_j) W_L(q_j)}{\sum_{k=1}^{|AT_u|} \sum_{q_i \in IC} W_{at}^k(q_i) W_L(q_i) \times |AT_u|} \tag{4}$$

(ii) For each q_j and its σ_z^j, there exists the observing probability distribution $P(\sigma_z^j \mid q_j)$. Given $Q_i = \{q_1', \cdots q_f' \mid 1 \le f \le |IC|, q' \in IC\}$ as a finite node set that user has accessed in at_i, denote $Q_{i,j}$ and Q_{i,j,σ_z} respectively as:

$$Q_{i,j} = \begin{cases} \{q_{j+y}' \mid q_j' = q_j, l = 0, \cdots (f-y)\} & q_j \in Q_i \\ Null & q_j \notin Q_i \end{cases}, \; Q_{i,j,\sigma_z} = \{q'' \mid q'' \in Q_{i,j}, q'' \mapsto \sigma_z^j\} \tag{5}$$

Denote $P(\sigma_z^j \mid q_j)$ as the ratio of the total number of posterior accessed nodes on σ_z^j and the total number of posterior accessed nodes on all concepts:

$$P(\sigma_z^j \mid q_j) = \frac{\sum_{i=1}^{|AT_u|} |Q_{i,j,\sigma_z}|}{\sum_{\sigma_z' \in \Sigma} |Q_{i,j,\sigma_z'}|} \tag{6}$$

(iii) The HMM with personalization interest description of Portal is to represent a state sequence with the maximal observing probability on certain personalization interest content σ_z^k, which can be defined as:

$$P_{\max}(\sigma_z^k) = \arg\max \prod_{q_k \in IC} P(q_k \to q_{k+1}) P(\sigma_z^k \mid q_k) \tag{7}$$

We consider not only the effect of different personalization interest behaviors on Portal, but also the prediction on interest intention in order to achieve the objectivity and integrality. Hence, the interest navigation patterns of every user can be inferred easily from the discovered navigation relationship. With the increasing of the user visiting, it is also necessary to set a threshold during the interest navigation patterns discovery [7] to ensure the accuracy and availability.

5 Experiments

We built a certain prototype Portal with BEA Weblogic Platform 8.1 in which the experiments might be performed completely. The dataset consists of 50 random users, 20 typical items forming *IC*, and 65,472 valid interest elicitation entries in 4,400 sessions obtained by our elicitation approach successfully.

5.1 Interest Elicitation Results and Analysis

The first experiment focuses on observing the effectiveness and evaluating the performance of implicit interest elicitation mechanism. After constructing the AT_us and S_us, we could compute the particular interest degree of each user with equation (3). Under this scenario, we observe the distribution of $W(content_k)$ varying with the number of sessions in AT_u and the user as a kind of measure method. Fig.1(a) shows one random user's interest distributions as the range of AT_u increases. It is clear that the interest degrees lead to user's personalization preference on *IC* when $|AT_u|$=50, and till $|AT_u|$=200, we could barely to detect the significant difference in flat region. Subsequently, we hold $|AT_u|$=50 and observe the interest distributions varying with different users. The results are illustrated in Fig.1(b), which indicate the inherent personalization interest of each user expected as our previous discussion.

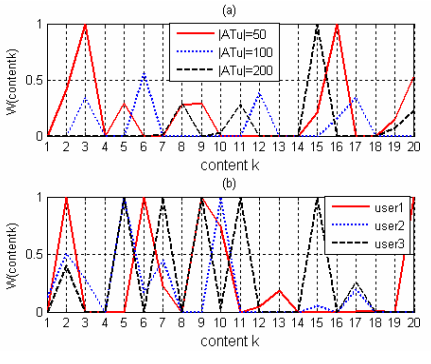


Fig. 1. Distribution of interest degree

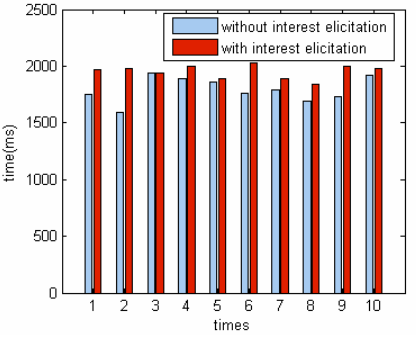


Fig. 2. Comparison in the elicitation performance

In this experiment, we also analyze the performance effect to the Portal server of our mechanism. We compare the cost (i.e., measured by time (ms)) on loading the Portal desktop into client browser whether the interest elicitation does in Fig.5. Although there is a little improvement measured within 12ms, our schema works with desired performance.

5.2 Mining Navigation Patterns Results and Analysis

The second experiment consists in discovering the interest navigation patterns of different users, which can be performed with two basic steps: categorizing the interest concept and calculating the observing probability. We still mine the representative entries in AT_u ($|AT_u|$=100) in Fig.1(a), setting three rough interest concept sets named $\sigma_1' = \{q_2, q_5, q_{13}, q_{15}\}$, $\sigma_2' = \{q_1, q_3, q_4, q_6, q_7, q_{10}, q_{11}, q_{12}, q_{14}, q_{16}, q_{19}, q_{20}\}$ and $\sigma_3' = \{q_8, q_9, q_{17}, q_{18}\}$ respectively. Under this scenario, the relative transfer probability $P(q_i \to q_j)$ is shown in Fig.3. The observing probability $P(\sigma_z^j \mid q_j)$ may be computed with equation (4-6) and contribute to fulfill our extended HMM. Fig.4(a-c) plot the user's entire possible interest navigation states on different interest concept sets measured by order. Hence, we could discover the interest navigation patterns with the maximal observing probability.

Pij =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0.0626	0.0171	0.0172	0.0296	0.0200	0.0200	0.0224	0.0185	0.0183	0.0175	0.0204	0.0481	0.0204	0.0806	0.0280	0.0205	0.0178	0.0182	0.0217
2	0.0119	0	0.0113	0.0114	0.0137	0.0119	0.0119	0.0124	0.0116	0.0116	0.0114	0.0120	0.0173	0.0120	0.0234	0.0134	0.0120	0.0115	0.0116	0.0122
3	0.0241	0.0843	0	0.0202	0.0377	0.0241	0.0241	0.0275	0.0220	0.0218	0.0207	0.0247	0.0639	0.0247	0.1097	0.0355	0.0249	0.0210	0.0216	0.0265
4	0.0239	0.0831	0.0198	0	0.0373	0.0239	0.0239	0.0273	0.0218	0.0216	0.0205	0.0244	0.0638	0.0244	0.1081	0.0351	0.0246	0.0208	0.0214	0.0263
5	0.0151	0.0368	0.0136	0.0137	0	0.0151	0.0151	0.0163	0.0140	0.0142	0.0138	0.0153	0.0294	0.0153	0.0460	0.0192	0.0154	0.0140	0.0142	0.0160
6	0.0200	0.0627	0.0171	0.0172	0.0297	0	0.0200	0.0225	0.0185	0.0184	0.0176	0.0204	0.0482	0.0204	0.0808	0.0281	0.0205	0.0178	0.0182	0.0217
7	0.0200	0.0627	0.0171	0.0172	0.0297	0.0200	0	0.0224	0.0185	0.0184	0.0176	0.0204	0.0482	0.0204	0.0807	0.0281	0.0205	0.0178	0.0182	0.0217
8	0.0181	0.0523	0.0157	0.0158	0.0258	0.0180	0.0180	0	0.0168	0.0167	0.0161	0.0184	0.0407	0.0184	0.0668	0.0245	0.0185	0.0163	0.0166	0.0194
9	0.0218	0.0719	0.0183	0.0185	0.0331	0.0217	0.0217	0.0246	0	0.0198	0.0189	0.0222	0.0549	0.0222	0.0931	0.0312	0.0224	0.0192	0.0196	0.0238
10	0.0220	0.0731	0.0185	0.0186	0.0335	0.0220	0.0220	0.0249	0.0202	0	0.0190	0.0225	0.0557	0.0225	0.0947	0.0316	0.0226	0.0193	0.0198	0.0240
11	0.0233	0.0797	0.0194	0.0195	0.0360	0.0232	0.0232	0.0265	0.0213	0.0210	0	0.0238	0.0606	0.0238	0.1036	0.0339	0.0239	0.0203	0.0208	0.0255
12	0.0196	0.0606	0.0168	0.0169	0.0289	0.0196	0.0196	0.0220	0.0182	0.0180	0.0173	0	0.0467	0.0200	0.0780	0.0274	0.0201	0.0175	0.0179	0.0213
13	0.0126	0.0238	0.0119	0.0119	0.0151	0.0126	0.0126	0.0133	0.0122	0.0122	0.0120	0.0127	0	0.0127	0.0285	0.0147	0.0128	0.0120	0.0121	0.0131
14	0.0196	0.0607	0.0168	0.0169	0.0289	0.0196	0.0196	0.0220	0.0182	0.0180	0.0173	0.0200	0.0467	0	0.0780	0.0274	0.0201	0.0175	0.0179	0.0213
15	0.0114	0.0174	0.0110	0.0110	0.0128	0.0114	0.0114	0.0118	0.0112	0.0112	0.0111	0.0115	0.0154	0.0115	0	0.0126	0.0115	0.0111	0.0112	0.0117
16	0.0155	0.0392	0.0139	0.0140	0.0289	0.0155	0.0155	0.0169	0.0147	0.0146	0.0142	0.0158	0.0311	0.0158	0.0492	0	0.0158	0.0143	0.0145	0.0165
17	0.0195	0.0600	0.0167	0.0168	0.0287	0.0195	0.0195	0.0218	0.0181	0.0179	0.0172	0.0199	0.0463	0.0199	0.0771	0.0271	0	0.0174	0.0178	0.0211
18	0.0228	0.0775	0.0191	0.0192	0.0352	0.0228	0.0228	0.0259	0.0209	0.0207	0.0197	0.0233	0.0590	0.0233	0.1006	0.0331	0.0235	0	0.0205	0.0250
19	0.0222	0.0743	0.0187	0.0188	0.0340	0.0222	0.0222	0.0252	0.0204	0.0202	0.0192	0.0227	0.0566	0.0227	0.0963	0.0320	0.0229	0.0195	0	0.0243
20	0.0185	0.0549	0.0160	0.0161	0.0268	0.0185	0.0185	0.0206	0.0173	0.0171	0.0164	0.0189	0.0426	0.0189	0.0703	0.0254	0.0190	0.0167	0.0170	0

Fig. 3. Distribution of transfer probability $P(q_i \to q_j)$

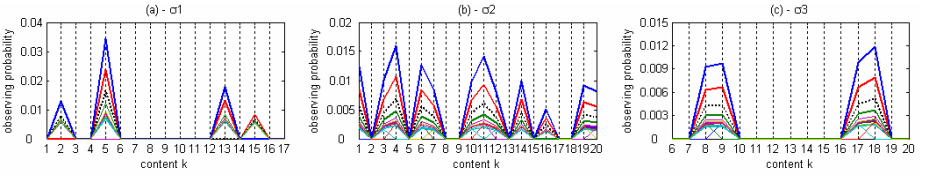


Fig. 4. Distribution of interest navigation patterns on rough interest concept sets

The results show that the HMM extension with personalization interest description of Portal has met the proposed interest elicitation mechanism sufficiently. Higher representation accuracy can be achieved, while helping us better to predict and understand the personalization interest navigation patterns and intentions of each user.

6 Conclusion

Mining personalization interest and navigation patterns on Portal is significant for the interest mining tasks in Portal area. In this paper, we summarize the mapping of the personalization interest behaviors on Portal and propose a novel Portal-independent mechanism of interest elicitation with privacy protection. Furthermore, we present an HMM extension with personalization interest description of Portal to discover the interest navigation patterns. The improvement on representation accuracy and mining capability for the complex personalization interests on Portal is a feature that clearly distinguishes our approaches from traditional ones. Since no attempt has previously been made to apply the mining results to better serve the personalization recommendation on Portal and perfect the consideration of privacy protection consideration, we attempt to conduct such a study to proceed as a design guideline in future research.

References

1. Massimiliano Albanese, Antonio Picariello, Carlo, et al. Web personalization based on static information and dynamic behavior. In: Proceedings of the ACM WIDM'04, USA, 2004: 80-87
2. Dong-Ho Kim, Il Im, Nabil Adam, et al. A clickstream–based collaborative filtering personalization model: Towards a better performance. In: Proceedings of the ACM WIDM'04, USA, 2004: 88-94
3. Luigi Lancieri, Nicolas Durand. Internet user behavior: compared study of the access traces and application to the discovery of communities. In: IEEE Transactions on System, Man and Cybernetics-Part A: Systems and Humans, 2006: 36(1)
4. Diamanto Oikonomopoulou, Maria Rigou1, Spiros Sirmakessis, et al. Full-Coverage Web prediction based on Web usage mining and site topology. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, 2004
5. Mao Chen, Andrea LaPaugh, Jaswinder Pal Singh. Categorizing information objects from user access patterns. In: Proceedings of the ACM CIKM'02, USA, 2002: 365-372
6. Juan Vel´asquez, Hiroshi Yasuda and Terumasa Aoki. Combining the Web content and usage mining to understand the visitor behavior in a web site. In: Proceedings of the 3rd IEEE International Conference on Data Mining, 2003
7. Shi Wang, Wen Gao, Li Jin-Tao, et al. Mining interest navigation patterns based on Hidden Markov model. Chinese Journal of Computers. 2001, 24(2): 152-157
8. Jing Wu, Zhang Xiong. A Portal-oriented personalized recommendation using meta-recommender engine. In: Proceedings of the International Conference on Artificial Intelligence, China, 2006: 570-576
9. Baoyao Zhou, Siu Cheung Hui, Alvis C. M. Fong. Discovering and visualizing Temporal-based Web access behavior. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence. 2005
10. John Canny. Collaborative filtering with privacy via factor analysis. In: Proceedings of the 25th ACM SIGIR, 2002
11. Daniel J. Weitzner, Harold Abelson, Tim Berners L. et al. Transparent accountable data mining: new strategies for privacy protection. In: Computer Science and Artificial Intelligence Laboratory Technical Report. www.csail.mit.edu , 2006
12. W3C.org: http://www.w3.org/2000/10/swap/

Cross-Lingual Document Clustering

Ke Wu and Bao-Liang Lu*

Department of Computer Science and Engineering, Shanghai Jiao Tong University
800 Dong Chuan Road, Shanghai 200240, China
{wuke, bllu}@sjtu.edu.cn

Abstract. The ever-increasing numbers of Web-accessible documents are available in languages other than English. The management of these heterogeneous document collections has posed a challenge. This paper proposes a novel model, called a domain alignment translation model, to conduct cross-lingual document clustering. While most existing cross-lingual document clustering methods make use of an expensive machine translation system to fill the gap between two languages, our model aims to effectively handle the cross-lingual document clustering by learning a cross-lingual domain alignment model and a domain-specific term translation model in a collaborative way. Experimental results show our method, i.e. C-TLS, without any resources other than a bilingual dictionary can achieve comparable performance to the direct machine translation method via a machine translation system, e.g. Google language tool. Also, our method is more efficient.

1 Introduction

The development of the World Wide Web has created the ever-increasing numbers of Web-accessible documents in languages other than English. The automated organization of these heterogeneous document collections has posed a challenge. On the other hand, the literature about cross-lingual document clustering is sparse. Typically, machine translation system is introduced to fill the gap between different languages[2,3]. In this paper, we propose a novel model, called **domain alignment translation model**, to effectively cluster the multilingual documents. Our model is inspired by the observation that its translation of a word greatly depends on the domain information of the context. In addition, our method differs widely from existing methods in that instead of the process of term translation and then clustering, the domain alignment translation model conducts term translation and clustering simultaneously by learning a bilingual domain alignment model and a domain-specific term translation model. This occurs in a collaborative way with the help of a bilingual translation dictionary

* To whome correspondence should be addressed. This work was supported in part by the National Natural Science Foundation of China under the grants NSFC 60375022 and NSFC 60473040, and the Microsoft Laboratory for Intelligent Computing and Intelligent Systems of Shanghai Jiao Tong University.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 956–963, 2007.

after conducting monolingual document clustering on two document sets, respectively. Experimental results show that the method based on the proposed model can achieve a comparable performance with the direct machine translation method, and that in some cases, the method can even outperform the latter one greatly.

The rest of this paper is organized as follows. In Section 2, we present related work on cross-lingual document clustering. In Section 3, we describe the domain alignment translation model consisting of a cross-lingual domain alignment model and a domain-specific term translation model. A method based on the proposed model is described in detail in Section 4. Experimental results with the method on data collected from the Internet are shown in Section 5. Finally, we conclude in Section 6.

2 Related Work

The literature about cross-lingual document clustering is sparse. Evans et al. (2003, 2004) [2] [3] used simple document translation for multilingual clustering in their Columbia Newsblaster system. Although they developed a simple dictionary lookup glossing system for Japanese and Russian, the system performed less well than full translation. Mathieu et al. (2004)[1] proposed a cross-lingual similarity measure for the documents, using bilingual dictionaries, employing a Shared Nearest Neighbor approach by Ertoz et al. (2001)[6] to cluster cross-lingual documents and achieving promising results. However, their method was not compared with full-fledged translation and it was not practical since it took eight hours for 3,000 documents to cluster in the cluster discovery phrase. Furthermore, Evans and Mathieu noticed a common phenomenon that found documents from the same language tending to cluster more easily than from different languages. Compared with the above two methods, Chen and Lin(2000)[4] proposed a different cluster mapping approach for cross-lingual document clustering in their multilingual news summarizer but did not conduct experiments for the clustering performance, since their system is for multilingual news summarizer. In their cross-lingual clustering, they select words with high frequency occurrence in the target language as the translations of the words in the source language.

3 Domain Alignment Translation Model

3.1 Model Description

Before describing the model, the following notations are introduced.

- S denotes a set of source words to be translated. It can be further represented as $\{w_i^S\}, i = 1 \ldots M$, where w_i^S is the ith word in S.
- T denotes a set of translated words given S. It can be further represented as $\{w_i^T\}, i = 1 \ldots M$, where w_i^T is a translation of the ith word in S. w_{ij}^T denotes the jth candidate translation of the ith word in S.

- $GEN(S)$ is a set of candidate translations given S.
- C denotes some specific domain and ζ denotes domain sets. That is, C is an element of ζ.

We use the term **domain alignment translation model** to refer to a mechanism that determine the probability $P(T, C|S)$. We need to gather the heterogeneous documents, *e.g.* Chinese documents and English documents into different groups. Compared with homogeneous documents, *e.g.* only Chinese document or only English document, there exists a wide language gap among heterogeneous documents. Meanwhile, it is our observation that a strong relationship between a translation of a word and its domain exists. For example, there are varied translations in different domains in the case of , the translation of which is export in business domain , is exit in transportation domain and is speak in politics domain etc. Accordingly, it is reasonable to search for the translation of words and the specific domain simultaneously. According to Bayes's theorem, given a set of source words S, the best T and C is the one that carry out maximization as follows:

$$\begin{aligned}\{T^*, C^*\} &= \underset{T \in GEN(S), C \in \zeta}{\arg\max} P(T, C|S) \\ &= \underset{T \in GEN(S), C \in \zeta}{\arg\max} P(C|S)P(T|C, S) \end{aligned} \tag{1}$$

where $P(C|S)$ is called cross-lingual domain alignment model and $P(T, C|S)$is called domain-specific term translation model. If we postulate that given a specific domain C and a set of source words S, its translation of each word in S is generated conditionally independently. The second term in Equation (1) can be reformulated as $P(T|C, S) = \prod_i P(w_i^T|w_i^S, C)$. Equation (1) can then be rewritten as

$$\{T^*, C^*\} = \underset{T \in GEN(S), C \in \zeta}{\arg\max} P(C|S) \prod_i P(w_i^T|w_i^S, C) \tag{2}$$

3.2 Parameter Estimation

In the section, we describes how to estimate the probabilities $P(w_{ij}^T|w_i^S, C)$ and $P(C|S)$. If we had available parallel corpus from some specific domain C, estimating $P(w_{ij}^T|w_i^S, C)$ could be the same as estimating the translation model in IBM noisy channel model. However, it is usually non-trivial to explicitly define what is the domain we need. On the other hand, it is also hard to acquire large scale parallel corpus. Therefore, we try to obtain $P(w_{ij}^T|w_i^S, C)$ from the corpus in the target language. Applying the chain rule to $P(w_{ij}^T|w_i^S, C)$, we can deduce Equation (3):

$$P(w_{ij}^T|w_i^S, C) = \frac{P(w_{ij}^T, C|w_i^S)}{P(C|w_i^S)} \tag{3}$$

If we assume that the occurrence of its translation w_{ij}^T in domain C is independent of word w_i^S, Equation (3) can be approximated through $\frac{P(w_{ij}^T, C)}{P(C|w_i^S)}$. Then we can obtain the following formula:

$$P(w_{ij}^T|w_i^S, C) = \frac{P(w_{ij}^T|C)}{P(w_i^S|C)} \cdot P(w_i^S) \tag{4}$$

Also, according to total probability formula, $P(w_i^S|C) = \sum_j P(w_{ij}^T|C)$. Therefore, Equation (4) can be written as:

$$P(w_{ij}^T|w_i^S, C) = \frac{P(w_{ij}^T|C)}{\sum_j P(w_{ij}^T|C)} \cdot P(w_i^S) \tag{5}$$

The problem of estimating $P(w_{ij}^T|w_i^S, C)$ now can be solved via estimating $P(w_{ij}^T|C)$ and $P(w_i^S)$. The probability of some translation w_{ij}^T of a source word w_i^S in a specific domain, $P(w_{ij}^T)$, can be calculated by the relative frequency of translation w_{ij}^T in the domain, that is, $P(w_{ij}^T|C) = \frac{TF(w_{ij}, C)}{TF(w, C)}$, where $TF(w_{ij}, C)$ denotes the frequency of word w_{ij} in the domain C and $TF(w, C)$ denotes the frequency of all words in the given domain. As for $P(w_i^S)$, it is actually the unigram model and thus can use the MLE estimation, smoothed by some known techniques. However, it doesn't really involve the resulting decision for optimal C and T, since it is constant in the decision-making process.

4 An Algorithm Based on the Proposed Model

In section 3, we propose a domain alignment translation model. In this section, we propose a algorithm based on the model. Simply speaking, the algorithm comprises two steps: mono-lingual document clustering; two-level search, that is, to search for term translation and the corresponding cluster that maximize $P(T, C|S)$. In the monolingual document clustering phrase, we cluster the documents in a language at an appropriate cluster number. In the search phrase, we simultaneously search the aligned clusters and term translation.

The clustering algorithm based on naïve Bayes model has been shown to be effective for high dimensional text clustering. Also, the clustering model has the similar assumption as our proposed model, which each word is generated independently in the given domain. Hence, we choose the algorithm to conduct monolingual document clustering. One can be referred to [8] for details.

On the other hand, to obtain the optimal translations and domain of a set of source words, we have to try all possible combination of their translations and the domains. However, it is computationally prohibitive. Therefore, our best option is to use a greedy algorithm toward this end. In our proposed two-level search algorithm, we just choose the set of translations with most high probability given some domain to avoid try too many candidate translations, totally ignoring the other possible translation combinations. We refer to the two-level search algorithm based on clusters as C-TLS. The algorithm is summarized

Algorithm: C-TLS(D_1, D_2,K_1,K_2,Dic)
Input: D_1: document collection in language $L1$;
D_2: document collection in language $L2$;
K_1: the number of clusters to be partitioned for D_1
K_2: the number of clusters to be partitioned for D_2
Dic: the general-purpose bilingual dictionary from $L2$ to $L1$
Steps:
1. nbEM(D_1, K_1); nbEM(D_2,K_2); %% clustering algorithm based on NB model
2. Construct the corresponding centroid v_i for each cluster c_i of D_2;
3. **For** each cluster c_i for D_2
4. **For** each cluster c_j for D_1
5. search the translation of each word with most probability for the centroid v_i in c_j ;
6. Compute and record $P(T, C|v_i)$;
End For
7. Select $< c_i, c_j^* >$ as a mapping relation if $P(T, C|v_i)$ is the highest among the recorded scores.
End For
Output: a partition of the document data given by the cluster identity vector $C = \{c_1, c_2, \ldots, c_N\}, c_i \in \{1..K\}, N = |D_1| + |D_2|$

Fig. 1. Two-level search algorithm based on clusters

in Fig. 1. In this paper, we also investigate the extreme case of the algorithm, called TLS. That is, it occurs when K_2 equals to $|D_2|$ in C-TLS.

5 Experiments

5.1 Experimental Setup

The test data is collected via RSS reader[1] . The test data comprises Chinese Web pages and English Web pages from various Web sites. They consist of news during December 2005, consisting of 6,462 English Web pages and 6,011 Chinese Web pages. We should have collected data with seven topics. Unfortunately, when we translate all Chinese Web pages into English Web pages via translation tools provided by Google language tool, there are various errors for some Web pages via Google translation tool[2], so that we have to select five topics for experimentation. They include business, education, entertainment, science and sports. The category information is obtained by RSS reader. In addition, in the experiments, we use a general-purpose Chinese-English bilingual dictionary with about 292,000 entries.

In the paper, we use average purity and average entropy for our evaluation metrics. Average entropy is used to measure mean status of how the various classes of documents are distributed within each cluster.

[1] http://www.rssreader.com/
[2] http://www.google.com/language_tools

$$AverageEntropy = \frac{1}{k}\sum_{j=1}^{k} E_j \tag{6}$$

$$E_j = -\frac{1}{\log q}\sum_{i=1}^{q}\frac{n_i^j}{n_i} * \log(\frac{n_i^j}{n_i}) \tag{7}$$

where q is the number of classes in the document collection, k is the number of partitioned clusters, n_i is the number of documents from cluster i, and n_i^j is the number of documents from cluster i assigned to category j.

The second measure is average purity that measures the average extent to which each cluster contained documents from one primary class. The purity measure is defined as follows:

$$AveragePurity = \frac{1}{k}\sum_{j=1}^{k} P_j \tag{8}$$

where P_j is the fraction of the overall cluster size that the largest class of documents assigned to that cluster represents.

5.2 Experimental Results and Discussion

In our experiments, Our main experimental results are shown in Fig. 2. All results are shown as average ± 1 standard deviation over 5 runs. The term `Google`, `Google(I2C)` and `Google(C2C)` represent our three baselines. Specifically speaking, Google refers to the method employing nbEM algorithm to all preprocessed English web pages and translated Chinese web pages, while `Google(I2C)` denotes the method making a mapping from a translated Web page to clusters of native English Web pages through nbEM and `Google(C2C)` denotes the method relating clusters of the translated web page to clusters of the native English web pages. In addition, En2Ch indicates that English is source language and Chinese is target language, whereas Ch2En indicates the reverse case.

From Fig. 2, Fig. 3 and Table 1, we can summarize the results as follows:

- `C-TLS` have better performance than `TLS` and can achieve comparable performance to `Google(C2C)` while `Google(C2C)` has to spend much time and waste much storage space on the translated documents;

Table 1. Comparison of mean time of four methods spent over different numbers of clusters

	Methods	**Time**(sec.)				
		5	10	15	20	25
1	TSL(Ch2En)	114	182	255	678	1203
2	TSL(En2Ch)	100	154	206	268	310
3	C-TSL(Ch2En)	12	18	24	36	52
4	C-TSL(En2Ch)	13	21	29	43	61

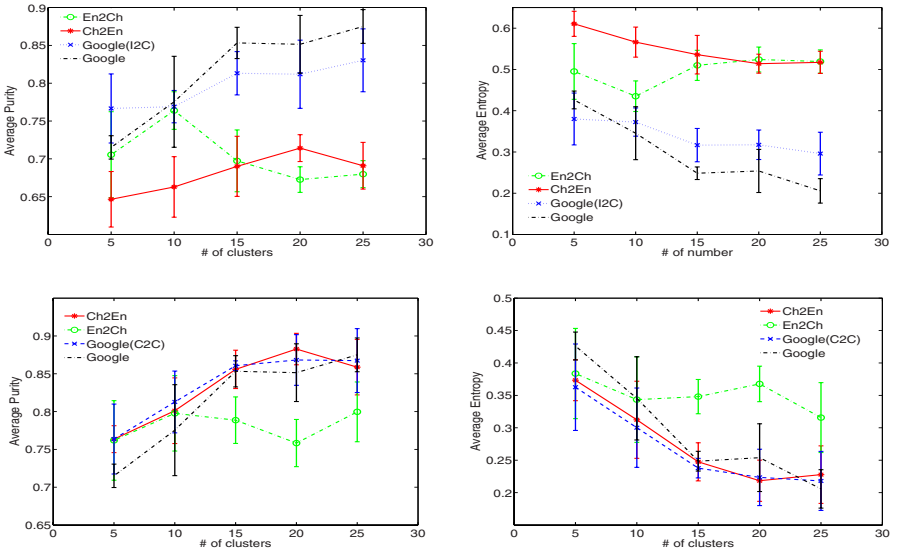


Fig. 2. Comparisons of different methods and baseline using direct machine translation. Results of `TLS`, `Google(I2C)` and `Google` are shown in the first row and results of `C-TLS`, `Google(C2C)` and `Google` are shown in the second row, where the number of clusters of English web pages is the same as one of Chinese web pages each run.

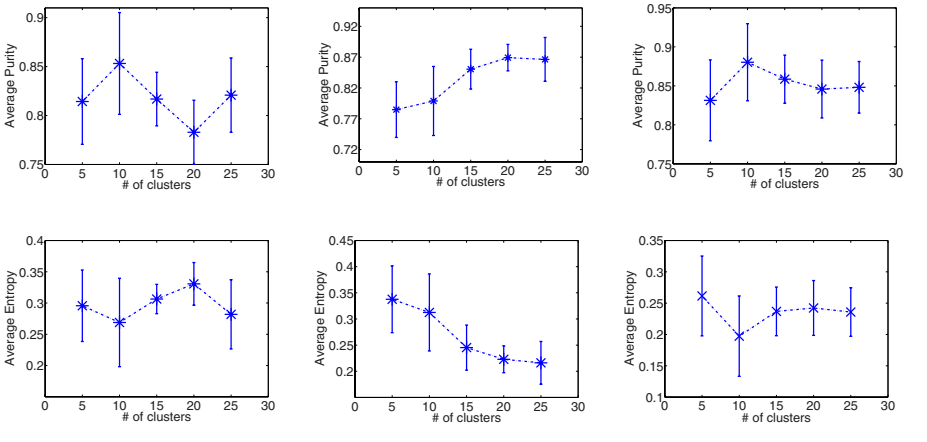


Fig. 3. Monolingual Clustering Results. Each column represents a set of results. Left-side column denotes Chinese web page clustering; middle-side column denotes English web page clustering; right-side column denotes the translated Chinese web page clustering.

- `C-TLS` achieves substantial and significant(p-value<0.05) improvements over `Google` method;
- Compared with `Google`, `Google(I2C)` and `Google(C2C)`, `TLS` and `C-TLS` is more efficient. It took about 8.3 hours for `Google`, `Google(I2C)` and

`Google(C2C)` to just translate Chinese web pages into English web pages and thus the time they spent on cross-lingual clustering is not listed in Table 1. In contrast, the longest runtime in Table 1 is about 20 minutes on Intel Pentium D 2.80GHz machine. This occurred when the number of clusters is 25 and `TLS(Ch2En)` method is used.

6 Conclusion

In this paper, we propose a novel domain alignment translation model to simultaneously conduct cross-lingual clustering and term translation. By learning a cross-lingual domain alignment model and a domain-specific term translation model in a collaborative way, we can cluster documents with a similar topic in different languages. Experimental results show our method without any resources other than a bilingual dictionary can achieve comparable performance to the direct machine translation method via Google translation tool. In our experiments, we only consider word, ignoring base phrase. We will incorporate translation of base phrase into our system in the future. On the other hand, the clustering in the source language and the clustering in the target language are related highly and thus we will explore how to reinforce their clustering quality interactively for future research.

References

1. Mathieu,B., Besançon,R., Fluhr C.: Multilingual document clusters discovery. In: RIAO'2004 proceedings, Université d'Avignon, France. (2004)
2. Evans,D., Klavans,J.L., McKeown,K.R. : Columbia Newsblaster: Multilingual News Summarization on the Web, In: Proc. HLT('04), Boston, MA. (2004)
3. Evans,D.K., Klavans,J.L.: A Platform for Multilingual News Summarization. Technical report, Columbia University Department of Computer Science. (2003)
4. Chen,H.H. and Lin,C.J. : A multilingual news summarizer. In: Proceedings of the 18th International Conference on Computational Linguistics. (2000) 159-165.
5. Hartigan,J.A. : Clustering Algorithms. John Wiley and Sons, Inc. (1975)
6. Ertoz,L., Steinbach,M., and Kumar,V. : Finding Topics in Collections of Documents: A Shared Nearest Neighbor Approach. In: Text Mine'01, Workshop on Text Mining (1st SIAM International Conference on Data Mining). (2001)
7. Carpuat,M. and Wu,D. : Word sense disambiguation vs. statistical machine translation. In: ACL 2005. (2005)
8. Meilă,M. , Heckerman,D. : An Experimental Comparison of Model-Based Clustering Methods. Machine Learning (2001) 42(1/2): 9-29

Grammar Guided Genetic Programming for Flexible Neural Trees Optimization

Peng Wu and Yuehui Chen

School of Information Science and Engineering
Jinan University, Jinan 250022, P.R.China
{ise_wup,yhchen}@ujn.edu.cn

Abstract. In our previous studies, Genetic Programming (GP), Probabilistic Incremental Program Evolution (PIPE) and Ant Programming (AP) have been used to optimal design of Flexible Neural Tree (FNT). In this paper Grammar Guided Genetic Programming (GGGP) was employed to optimize the architecture of FNT model. Based on the predefined instruction sets, a flexible neural tree model can be created and evolved. This framework allows input variables selection, over-layer connections and different activation functions for the various nodes involved. The free parameters embedded in the neural tree are optimized by particle swarm optimization algorithm. Empirical results on stock index prediction problems indicate that the proposed method is better than the neural network and genetic programming forecasting models.

1 Introduction

There has been growing interest in evolving architecture and parameters of a higher order Sigma-Pi neural network based on a sparse neural tree encoding [1]. Recently some approaches for evolving the neural tree model based on tree-structure-based evolutionary algorithm and random search algorithm have been proposed in [11] [12] [14].

Antonisse [15] used grammars firstly to constrain the generation of chromosome in his proposed system, which is called grammar-based GA. After then, some grammars-based GP systems was proposed. Stefanski [16] proposed the use of abstract syntax trees to set a declarative bias for GP. Robston [6] demonstrated how a formal grammar might be used to specify constraints for GP in the context of engineering design. Mizoguchi and Hemmi [7] suggested the use of production rules to generate hardware language descriptions during the evolutionary process.

Three typical grammar guided GP systems can be classified as: Whigham's CFG-GP system [8], Schultz's grammar-based expert systems and Wong's LOGENPRO system [10] [5]. Grammar Guided Genetic Programming (GGGP) [3] [4] is a typical tree-structure-based genetic programming system. GGGP using a grammar to constrain search space. The individual GP tree in GGGP must respect the grammar. This overcomes the closure problem in GP and provides a more formalized mechanism for typing (strongly-typed genetic programming).

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 964–971, 2007.

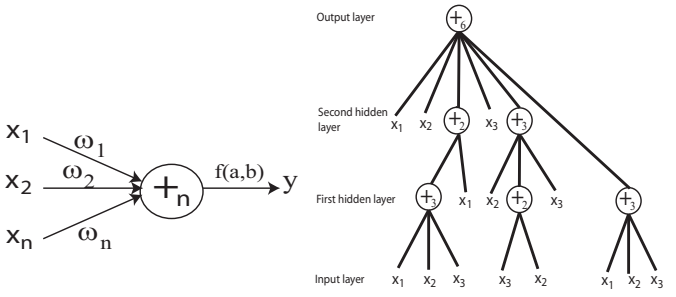


Fig. 1. A flexible neuron operator (left), and a typical representation of the FNT with function instruction set $F = \{+_2, +_3, +_4, +_5, +_6\}$, and terminal instruction set $T = \{x_1, x_2, x_3\}$ (right)

Actually, the grammar model can do more than just constrain the search space. In Whigham's work [9], in addition to the normal GGGP search, the grammar is slightly modified during the search. The updated grammar represents the accumulated knowledge found in the process of search.

In this paper, GGGP is firstly employed to optimize the Flexible Neural Tree (FNT). Based on a pre-defined instruction/operator sets, a flexible neural tree model can be created and evolved. FNT allows input variables selection, over-layer connections and different activation functions for different nodes. In our previous work, the hierarchical structure of FNT was evolved using PIPE with specific instructions [11] [12]. In this research, the hierarchical structure is evolved using the GGGP. The fine tuning of the parameters encoded in the structure is accomplished using Particle Swarm Optimization (PSO) [20]. The novelty of this paper is in the usage of GGGP for flexible neural tree optimization and for selecting the important inputs in the modeling of stock index.

The rest of paper is organized as follows. A simple introduction of Grammar Guided Genetic Programming is given in Section 2, and a hybrid-learning algorithm for evolving the FNT is also presented in this Section. Some simulation results for stock index prediction are given in Section 3. Finally, some conclusions are given in Section 4.

2 The Flexible Neural Tree Model

The function set F and terminal instruction set T used for generating a FNT model are described as $S = F \bigcup T = \{+_2, +_3, \ldots, +_N\} \bigcup \{x_1, \ldots, x_n\}$, where $+_i (i = 2, 3, \ldots, N)$ denote non-leaf nodes' instructions and taking i arguments. x_1,x_2,...,x_n are leaf nodes' instructions and taking no other arguments. The output of a non-leaf node is calculated as a flexible neuron model (see Fig.1). From this point of view, the instruction $+_i$ is also called a flexible neuron operator with i inputs. In the creation process of neural tree, if a nonterminal instruction, i.e., $+_i (i = 2, 3, 4, \ldots, N)$ is selected, i real values are randomly generated and used for representing the connection strength between the node

$+_i$ and its children. In addition, two adjustable parameters a_i and b_i are randomly created as flexible activation function parameters. For developing the forecasting model, the flexible activation function $f(a_i, b_i, x) = e^{-(\frac{x-a_i}{b_i})^2}$ is used. The total excitation of $+_n$ is $net_n = \sum_{j=1}^{n} w_j * x_j$, where $x_j (j = 1, 2, \ldots, n)$ are the inputs to node $+_n$. The output of the node $+_n$ is then calculated by $out_n = f(a_n, b_n, net_n) = e^{-(\frac{net_n - a_n}{b_n})^2}$. The overall output of flexible neural tree can be computed from left to right by depth-first method, recursively.

2.1 Tree Structure Optimization by GGGP

Grammar Guided Genetic Programming (GGGP) is one of the important extensions for GP [2]. The purpose of presented GGGP is mainly to overcome the closure problem [2], the generation and preservation of valid programs in GP system. For an object, some grammars are used to guide the generation of programs in GP, and a chosen declaration of bias can be set on the space of programs.

In this research, Context-free Grammar (CFG) [9] was chosen for FNT optimization. A CFG consists of 4 sets, $G = \{N, T, P, \Sigma\}$, Where N is a set of non-terminal symbols, T is a set of terminal symbols, P is set of production rules and Σ is set of start symbols, and $N \bigcap T = \emptyset, \Sigma \in N$. The production rules have the format $x \rightarrow y$, where $x \in N$, $y \in N \bigcup T$. The production rules specify how the non-terminal symbols should be written into one of their derivations until the expression contains terminal symbols only. For an example (Fig. 2), a CFG for generation one variable simply arithmetic expression can be described as follows,

$$
\begin{aligned}
&s \rightarrow exp \\
&exp \rightarrow exp\ op\ exp \\
&exp \rightarrow pre\ exp \\
&exp \rightarrow var \\
&pre \rightarrow sin|cos \\
&op \rightarrow +|- \\
&var \rightarrow x.
\end{aligned}
$$

Although the components of GGGP are the same as GP, there are still some distinct difference between GGGP and GP. In GGGP a tree-based program is generated according to the context-free grammar. In crossover, two internal nodes labeled with the same non-terminal symbol of the grammar are chosen at random, and the two sub-derivation trees underneath them are exchanged. In mutation, a new randomly generated sub-derivation tree rooted at the same non-terminal symbol replaces the sub-derivation tree of the selected node. The general evolutionary process in GGGP can be described as the same as GP. For detailed description of GGGP algorithm, please refer to [3] and [4].

2.2 Parameter Optimization with PSO

The Particle Swarm Optimization (PSO) conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA). A

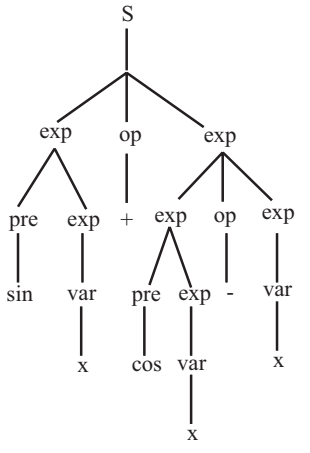


Fig. 2. Derivation tree of expression of $sin(x) + cos(x) - x$

population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector $\mathbf{x_i}$. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector $\mathbf{v_i}$. At each time step, a function f_i representing a quality measure is calculated by using $\mathbf{x_i}$ as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector $\mathbf{p_i}$. Furthermore, the best position among all the particles obtained so far in the population is kept track of as $\mathbf{p_g}$. In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle. At each time step t, by using the individual best position, $\mathbf{p_i}$, and the global best position, $\mathbf{p_g}(\mathbf{t})$, a new velocity for particle i is updated by

$$\mathbf{v_i}(\mathbf{t+1}) = \mathbf{v_i}(\mathbf{t}) + c_1\phi_1(\mathbf{p_i}(\mathbf{t}) - \mathbf{x_i}(\mathbf{t})) + c_2\phi_2(\mathbf{p_g}(\mathbf{t}) - \mathbf{x_i}(\mathbf{t})) \quad (1)$$

where c_1 and c_2 are positive constant and ϕ_1 and ϕ_2 are uniformly distributed random number in [0,1]. The term $\mathbf{v_i}$ is limited to the range of $\pm\mathbf{v_{max}}$. If the velocity violates this limit, it is set to its proper limit. Changing velocity this way enables the particle i to search around its individual best position, $\mathbf{p_i}$, and global best position, $\mathbf{p_g}$. Based on the updated velocities, each particle changes its position according to the following equation:

$$\mathbf{x_i}(\mathbf{t+1}) = \mathbf{x_i}(\mathbf{t}) + \mathbf{v_i}(\mathbf{t+1}). \quad (2)$$

For detailed description of PSO algorithm, please refer to [20].

2.3 The General Learning Algorithm

The general learning algorithm for GGGP-FNT model can be described as follow:

1) Initialization. Set the initial value of parameters used in GGGP and PSO algorithms. The initial population (flexible neural trees and the corresponding parameters) is generated randomly.

2) Structure optimization with GGGP algorithm, in which the fitness function is calculated by root mean square error (RMSE).
3) If a better structure is found then go to step 4), otherwise go to step 2).
4) Parameters optimization with PSO algorithm. In this stage, the structure of FNT is fixed and the best tree is taken from the end of run of the GGGP search, and the fitness function is also calculated by RMSE.
5) If the maximum number of iterations of GGGP algorithm is reached, or no better parameter vector is found for a significantly long time (100 steps) then go to step 6); otherwise go to step 4).
6) If satisfactory solution is found, then stop; otherwise go to step 2).

3 Experimental Studies

3.1 Stock Index Modeling

Prediction of stocks is generally believed to be a very difficult task - it behaves like a random walk process and time varying. The obvious complexity of the problem paves the way for the importance of intelligent prediction paradigms [17]. In this experiment, we analyze the seemingly chaotic behaviour of two well-known stock indices namely the Nasdaq-100 index of NasdaqSM [18] and the S&P CNX NIFTY stock index [19]. The Nasdaq-100 index reflects Nasdaq's largest companies across major industry groups, including computer hardware and software, telecommunications, retail/wholesale trade and biotechnology. The Nasdaq-100 index is a modified capitalization weighted index, designed to limit domination of the Index by a few large stocks while generally retaining the capitalization ranking of companies. Through an invest-ment in Nasdaq-100 index tracking stock, investors can participate in the collective performance of many of the Nasdaq stocks that are often in the news or have become household names. Similarly, S&P CNX NIFTY is a well-diversified 50 stock index accounting for 25 sectors of the economy. It is used for a variety of purposes such as benchmarking fund portfolios, index-based derivatives and index funds. The CNX Indices are computed using the market capitalization weighted method, wherein the level of the Index reflects the total market value of all the stocks in the index relative to a particular base period. The method also takes into account constituent changes in the index and importantly corporate actions such as stock splits, rights, and so on, without affecting the index value.

3.2 Experimental Setup and Results

In this experiment, we considered 7-year stock data for the Nasdaq-100 Index and 4-year for the NIFTY index. Our research investigates the performance of GGGP-FNT, GP and ANN for modeling the Nasdaq-100 and NIFTY stock market indices [13]. We used the same training and test data sets to evaluate the different models. The assessment of the prediction performance of the different paradigms were done by quantifying the prediction obtained on an independent

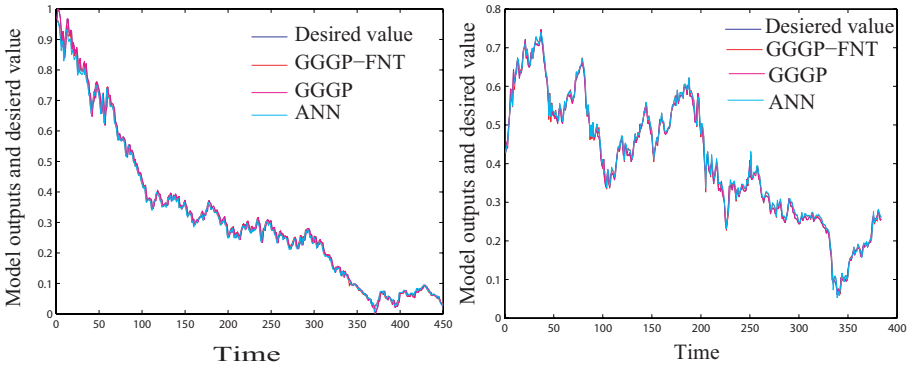


Fig. 3. Forecasting performances of the three models for the Nasdaq index (left) and NIFTY index (right)

data set. The Root Mean Squared Error (RMSE) is used to performance evaluation index.

The settings for GGGP-FNT are population size 100, cross rate 0.9, mute rate 0.1 and maximum depth 5. A FNT model was constructed using the training data and then the model was used on the test data set. The instruction sets used to create an optimal FNT forecaster are $S = \{+_2, +_3, x_1, x_2, x_3\}$ and $S = \{+_2, +_3, x_1, x_2, x_3, x_4, x_5\}$ for Nasdaq-100 and NIFTY stock index, respectively. Where $x_i (i = 1, 2, 3, 4, 5)$ denotes the 5 input variables of the forecasting model.

The grammars used for modeling the Nasdaq-100 index (left) and for modeling the NIFTY index (right) are shown as follow,

$$
\begin{array}{l}
s \rightarrow exp \\
exp \rightarrow exp\ op\ exp \\
exp \rightarrow thr\ exp\ exp\ exp \\
exp \rightarrow var \\
op \rightarrow +_2 \\
thr \rightarrow +_3 \\
var \rightarrow x_1 | x_2 | x_3
\end{array}
\qquad
\begin{array}{l}
s \rightarrow exp \\
exp \rightarrow exp\ op\ exp \\
exp \rightarrow thr\ exp\ exp\ exp \\
exp \rightarrow var \\
op \rightarrow +_2 \\
thr \rightarrow +_3 \\
var \rightarrow x_1 | x_2 | x_3 | x_4 | x_5
\end{array}
$$

For comparison purpose, a GGGP was also implemented to forecast the stock index. The settings for GGGP are population size 100, cross rate 0.9, mute rate 0.1, and maximum depth 15. The instruction sets $S = \{+, -, *, sin, cos, exp, x_1, x_2, x_3\}$ and $S = \{+, -, *, sin, cos, exp, x_1, x_2, x_3, x_4, x_5\}$ are used for modeling the Nasdaq-100 index and the NIFTY index, respectively. Training was terminated after 3000 epochs on each dataset.

Two ANN models with architecture $\{3 - 10 - 1\}$ and $\{5 - 10 - 1\}$ trained by PSO are also implemented for modeling the Nasdaq-100 index and the NIFTY index, respectively. Training was terminated after 3000 epochs on each dataset.

Table 1 summarizes the training and test results achieved for the two stock indices using the three different approaches. Figures 3 and 4 depict the test

Table 1. Comparison of RMSE results for three learning methods (training)

	GGGP-FNT	GGGP	ANN
Nasdaq-100	0.02582	0.02568	0.02573
NIFTY	0.01699	0.01658	0.01729

Table 2. Comparison of RMSE results for three learning methods (testing)

	GGGP-FNT	GGGP	ANN
Nasdaq-100	0.01725	0.01993	0.01789
NIFTY	0.01291	0.01366	0.01426

results for the one-day ahead prediction of the Nasdaq-100 index and the NIFTY index, respectively.

Comparing GGGP-FNT with GGGP and ANN, we found that GGGP-FNT has better generalization ability and high accuracy than GGGP and ANN forecasting models.

4 Conclusions

In this paper, a GGGP and PSO based learning algorithms are employed to optimal design of the FNT models. Simulation results on stock index forecasting problems show the feasibility and effectiveness of the proposed method. For the GGGP algorithm itself, the vital topic is a Context-free Grammar model (CFG). The gammer and its self-turning should be further discussed in our future work. It should be noted that other grammar model can also be used to guide the GP and used to design of FNT, and therefore it is valuable to give a further investigation.

Acknowledgment

This research was partially supported by the Natural Science Foundation of China under contract number 60573065 and the Key Subject Research Foundation of Shandong Province.

References

1. Zhang, B.T., et al.: Evolutionary induction of sparse neural trees. Evolutionary Computa-tion.5, (1997) 213-236
2. Koza,J. R.: Genetic Programming. On the Programming of Computers by Natural Selection, MIT Press, MA, (1992)
3. Nguyen X.H.: A Flexible Representation for Genetic Programming: Lessons from Natural Language Processing, Chapter 2, PhD Thesis, University of New South Wales, Australia, (2004)7-29

4. Shan, Y., McKay, R., Baxter, R., Abbass, H., Essam, D., and Nguyen., H.: Grammar model based program evolution. In Proceedings of The Congress on Evolutionary Computation, Portland, USA. IEEE.(2004)
5. Wong, M.L.and Leung, K.S.: An Adaptive Inductive Logic Programming System using Genetic Programming. Proceedings of the Fourth Annual Conference on Evolutionary Programming. MIT Press,(1995)737-752
6. Ross, B.J.: The Evolution of Stochastic Regular Motifs for Protein Sequences. New Gen-eration Computing, 20(2), (2002)187-213
7. Mizoguchi, J.,et al: Production Genetic Algorithms for Automated Hardware Design through Evolutionary Process. Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE Press, (1994)85-90
8. Whigham, P.A.: Grammatically-Based Genetic Programming, Proceedings of the Work-shop on Genetic Programming. From Theory to Real-World Applications, Morgan Kauf-mann, (1995)33-41
9. Whigham, P.A., Inductive bias and genetic programming. Proc. of First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, pages 461.466. UK:IEE, September 1995.
10. Wong, M.L. and Leung, K.S.: An Adaptive Inductive Logic Programming System using Genetic Programming. Proceedings of the Fourth Annual Conference on Evolutionary Programming, MIT Press, (1995)737-752
11. Chen, Y., Yang, Y. and Dong, J.: Nonlinear System Modeling via Optimal Design of Neural Trees. Int. J. of Neural Systems 14(2)(2004) 125 - 137
12. Chen, Y., Abraham, A., Yang, J. and Yang, B.: Hybrid Methods for Stock Index Modeling. Fuzzy Systems and Knowledge Discovery: Second International Conference (FSKD 2005), China, LNCS 3614, (2005)1067-1070
13. Chen, Y., Yang, B., Dong, J. and Abraham, A.: Time-Series Forecasting using Flexible Neural Tree Model. Information Science 174(3-4),(2005)219 - 235
14. Chen, Y., Yang, B., Dong, J.: Evolving Flexible Neural Networks using Ant Program-ming and PSO algorithm", International Symposium on Neural Networks (ISNN'04), LNCS 3173, (2004)211-216
15. Antonisse, H. J.: A Grammar-Based Genetic Algorithm. in G.J.E. Rawlins, editor, Foun-dations of Genetic Algorithms, Morgan Kaufmann, (1991)
16. Stefanski, P.A.:Genetic Programming Using Abstract Syntax Trees. Notes from the Ge-netic Programming Workshop (ICGA 93), (1993)
17. Abraham A., Nath B. and Mahanti P.K.: Hybrid Intelligent Systems for Stock Market Analysis. Computational Science, Springer-Verlag Germany, Vassil N Alexandrov et al. (Editors), (2001)337-345
18. Nasdaq Stock MarketSM: http://www.nasdaq.com
19. National Stock Exchange of India Limited: http://www.nse-india.com
20. Kennedy, J. and Eberhart, R.C.: Particls swarm optimization. Proc. IEEE int'l conf. on neural networks Vol. IV, (1995)1942-1948

A New Initialization Method for Clustering Categorical Data*

Shu Wu[1], Qingshan Jiang[1,**], and Joshua Zhexue Huang[2]

[1] School of Software, Xiamen University, Xiamen 361005, China
[2] E-Business Technology Institute, The University of Hong Kong, Hong Kong
james.wushu@gmail.com, qjiang@xmu.edu.cn, jhuang@eti.hku.hk

Abstract. Performance of partitional clustering algorithms which converges to numerous local minima highly depends on initial cluster centers. This paper presents an initialization method which can be implemented to partitional clustering algorithms for categorical data sets with minimizing the numerical objective function. Experimental results show that the new initialization method is more efficient and stabler than the traditional one and can be implemented to large data sets for its linear time complexity.

Keywords: Data mining; Cluster analysis; Partitional clustering; Categorical attribute; Initialization method.

1 Introduction

Partitioning a set of objects with multiple attributes into homogeneous clusters is one of the most fundamental operations in data mining. Object is divided into a series of sub-objects or clusters so that data points are more 'similar' to data points in the same cluster than data points in the other clusters. Many algorithms have been developed for clustering datasets. The *FCM* [1] algorithm proposed by Bezdek is broadly concerned, since it induces the concept of fuzzy set and can efficiently cluster large data sets. However, since the aim of *FCM* algorithm is to attain the minimum of its numerical objective function, it could only be applied to numerical data sets. Recently lots of researchers have carried on research on clustering method of categorical data sets, and have proposed many different algorithms such as fuzzy k -modes [2], ROCK [3] and COOLCAT [4]. Constructed in the framework of *FCM* , fuzzy k -modes algorithm is as effective as *FCM* algorithm in clustering large data sets.

In partitional clustering algorithms the procedure adopted for choosing initial cluster centers is extremely important as it has a direct influence on the formation of final clusters. Frequently, random centers may induce a cluster process to terminate in a local optimal result, while centers that well reflect data distribution probably serve to the global optimal clusters or comparatively good ones.

Several attempts, which primarily concentrated on clustering numerical data, have been reported to solve the initialization problem. Forgy adopted the random method in

* This research work is supported by No. 0000-X07204 and No. 0630-XK0011.
** Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 972–980, 2007.

1965 [5], Duda and Hart proposed a recursive method [6], Kaufman and Rousseeuw [7] introduced a method based on density, and Bradley and Fayyad [8] proposed a refining procedure. However, few researches are concerned with initializing categorical data. Ying Sun [9] introduced an initialization method which is based on the framework of refining. This method presents a study on applying Bradley's algorithm [8] to the k -modes, but its time cost is high and the parameters of this method are plenty.

To solve these problems, we propose a new initialization method which limits the process in a sub-sampled data set and uses a refining framework. It defines the point's density and the probability of a point to be a center in the sub-sample step. The new method can be implemented to all partitional clustering algorithms for categorical data set, though it is only applied in fuzzy k -modes [2] method in this paper. The paper is organized as follows. Section 2 explains our new initialization method. Section 3 presents and analyzes experimental result. Section 4 concludes the presentation.

2 The New Initialization Method

We propose a new initialization method and implement it in fuzzy k -modes [2].

2.1 Some Concepts About Categorical Data

Suppose $X = \{x_1, x_2, ...x_n\}$ is a categorical data set, where n is the number of objects, and $x_j (1 \le j \le n)$ is the set of categorical dimension as $[A_1, A_2, ..., A_p]$. The A_l is a value determined by $[a_l^{(1)}, a_l^{(2)}, ..., a_l^{(n_l)}] (1 \le l \le p)$ and n_l indicates the number of values in attribute A_l . x_j can be denoted through the form of $| x_{j1}, x_{j2}, ..., x_{jp} |$. $V = (v_1, v_2, ..., v_c)$ is the set of all cluster centers, and cluster center v_i is expressed by $[v_{i1}, v_{i2}, ..., v_{ip}]$, where c is the number of cluster. In fuzzy k -modes [2], the distance is defined as follow:

$$d(x_j, v_i) = \sum_{l=1}^{p} \delta(x_{jl}, v_{il}), \text{ where } \delta(x_{jl}, v_{il}) = \begin{cases} 1 & if\ v_{il} = x_{jl} \\ 0 & if\ v_{il} \neq x_{jl} \end{cases}.$$

2.2 Fundamental Initialization Steps

Firstly, we induce the definition of point's density for categorical objects. Take the numerical data as an example, Fig.1 illustrates the distribution of a numerical data set.

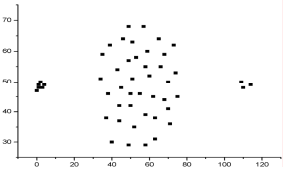


Fig. 1. Numerical data set with 3 classes

We extend the concept of density to categorical data sets on the assumption that categorical data sets also have this trait of distribution as numerical data sets.

Definition 1 [Point's density]. The density of a point is defined as follows:

$$Dens(x_i) = 1/\sum_{j=1}^{n} d(x_i, x_j) \tag{1}$$

The maximum value of a categorical data, if it can be expressed in a graph, is related with the densest point, which is most probability to be a cluster center.

When searching for a new center, if distances between the point and the already existing cluster centers are the only factors considered, it is possible that an outlier is taken as a new one. Meanwhile if only density is taken into consideration, it is most possible that many cluster centers locate in the surrounding of one center. They are unreliable initial points which could lead to bad partitions after the clustering process. In order to avoid these potential problems, we propose the probability of a point to be a cluster center, integrating the distance and the density measurements together.

Definition 2 [Probability to be a cluster center]. The probability of point x_i to be a cluster center is defined as follows:

$$Probability(x_i) = \alpha Dens(x_i) \times \beta \min(d(v_k, x_i)) \tag{2}$$

where $\alpha = 1/\underbrace{\max}_{n}\{Dens(x_r)\}$ and $\beta = 1/\max\{\underbrace{\min}_{c} d(v_k, x_j)\}$.

The fundamental steps for choosing v_i (the i th cluster center) are listed as follows:

Step 1. Calculate the densities of points we choose from a data set (Equation 1);
Step 2. Set the densest point as the first initial cluster center;
Step 3. Compute the probabilities of points and save it to set P (Equation 2);
Step 4. Choose the point with the maximum in P as v_i $(i \geq 2)$ and weed it out;
Step 5. If $i \geq c$, iteration terminate, go to step 6; or else, order $i = i + 1$, go to step 4;
Step 6. Output cluster center set $V (V = \bigcup_{i=1}^{c} v_i)$.

The time complexity of the above process is $O(n^2)$ which is higher than the clustering process. Therefore it is limited in clustering a small data set. In the next part, we induce a sub-sample method and refining framework, extending it to a large data set.

2.3 The New Initialization Method

The data set [10] described in Fig. 2 is a gauss data set with two attributes. After random sampling, a squared number of the whole data set was chosen and illustrated in Fig. 3. Its distribution approximately reflects that of the whole data set.

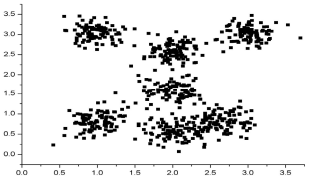


Fig. 2. Distribution of a data set with two dimensions

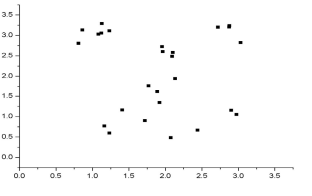


Fig. 3. Distribution after sub-sampled

Extending experiential principle to categorical data, the new method is defined below:

Step 1: Sub-sample Initialization

1.1 $CS = \phi$

1.2 For $i = 1,...,c$

1.2.1 Randomly choose the squared number of points sp_i as a sub-sample;

1.2.2 Choose cluster centers using the fundamental steps in section 2.2;

1.2.3 Calculate distance and belonging matrix, compute new centers cs_i ;

1.2.4 Save new cluster centers cs_i to CS ($CS = \bigcup_{i=1}^{c} cs_i$).

Step 2: Refinement

2.1 FCS = ϕ

2.2 For $i = 1,...,c$

2.2.1 Cluster on CS , using cs_i as initial points;

2.2.2 Save final cluster centers fcs_i attained from clustering process to FCS ;

Step 3: Evaluation

3.1 For $i = 1,...,c$

3.11 Compute sse_i value of fcs_i ; (sse_i will be explained below)

3.2 Choose the minimum sse_i , and set fcs_i as initial points.

The new initialization method contains three major steps. In the first step, small sub-samples $sp_i (i = 1...c)$ are randomly chosen from the whole data set. The number of sp_i is the squared number of the whole data set. We use fundamental initialization steps to choose initial points from sp_i . The set cs_i is the result of one time clustering.

In the second step, CS is treated as a data set, and is clustered c times using cs_i as initial points. The final centers $fcs_i (i = 1,...c)$ are saved in the set FCS .

In evaluation step, we introduce the index SSE [11], which means the sum of square-error between data points and their cluster center and is defined as $SSE = \sum_{i=1}^{n} d(x_i, v_{c(i)})$, where $v_{c(i)}$ means the cluster center that a sample point x_i belongs to. The minimum of SSE is selected, and the fcs_i , which is related with the minimum of SSE , are set as initial points. Let SSE be the union of sse_i , $SSE = \bigcup_{i=1}^{c} sse_i$.

Time complexity of the whole process is $O(c(c \mid c^2 \mid))$, where $c \mid c^2 \mid$ means the time complexity that clustering data sets with c^2 instances to get c clusters. Bezdek [11] advises that the number of cluster should be chosen between $c_{\min} = 2$ and $c_{\max} = \sqrt{n}$. Therefore the time complexity of whole process is less than the time complexity $O(c \mid n \mid)$ in clustering categorical data set.

3 Experimental Result and Discussion

Since there are no universally accepted methods for selecting initial cluster centers as reported by Meila and Heckerman [12], we compare the new method with the traditional random one. We have used many data sets to test the new method, while in this

paper some public data sets implemented for analysis are attained from the UCI [13]. Missing attribute values are treated as special ones and numerical attribute values are treated as categorical ones. Description of experimental data sets is listed in table 1.

Table 1. Description of experimental data sets

Data set	No. instance	No. attribute	No. cluster	Missing Value
credit	690	16	2	some
hepatitis	155	20	2	some
mushroom	8124	22	2	some
spect	267	23	2	none
voting-records	435	17	2	some
monks problems	432	8	2	none
lung-cancer	32	57	3	some
molecular	3190	62	3	none
car	1728	6	4	none
soybean	47	21	4	none
zoo	101	17	7	none

In our experiments, we implement our method in fuzzy k -modes [2] and set fuzzy coefficient $m = 2$ and terminative condition $\varepsilon = 10^{-5}$. All data sets are clustered with both random and new initialization method. The results in the table are the means of *SSE* in ten times computation. We use decreased percent to reflect the decrease. It is defined as: Decreased Percent = (Rand Init *SSE* - New Init *SSE*) / Rand Init *SSE* .

3.1 Capability Test and Analysis

The tables below show that the results obtained from the new algorithm are better than the random one with different numbers of cluster centers.

Table 2. Clustering with pre-requisite cluster number

Data set	No. Clusters	Rand. Init. *SSE*	New. Init. *SSE*	Decreased Percent
credit	2	6460	5448	15.67%
hepatitis	2	1599	1366	14.57%
mushroom	2	98716	68005	31.11%
spect	2	1791	1263	29.48%
voting-records	2	3814	1921	49.63%
monks	2	1803	1458	19.13%
lung-cancer	3	742	623	16.04%
molecular	3	141930	128414	9.52%
car	4	7827	5725	26.86%
soybean	4	428	285	33.41%
zoo	7	522	211	59.58%

We can observe from the table 2 that new method is prone to get a better result than the random one when clustering with a pre-requisite cluster number. The decreased percents of *SSE* are all above 10%, and the highest one nears 60%. The results of new method are evidently easier to approach the optimal distribution of clusters.

Table 3. Clustering with 8 cluster centers

Data set	Rand. Init. *SSE*	New. Init. *SSE*	Decreased Percent
credit	6492	4948	23.78%
hepatitis	1466	1158	21.01%
mushroom	76900	51461	33.08%
spect	1635	982.5	39.91%
voting-records	3674	1676	54.38%
monks problems	1567	981	37.40%
molecular biology	137890	124297	9.86%
car	7500	4857	35.24%
zoo	497	216	56.54%

Since soybean and lung-cancer data sets can no more meet our condition $c^2<n$, they are eliminated from the initialization process. The decreased percents in table 3 indicate that new method can attain a better result, and are closer to optimal distribution.

Table 4. Clustering with 16 cluster centers

Data set	Rand. Init. *SSE*	New. Init. *SSE*	Decreased Percent
credit	6236	4550	27.04%
mushroom	88884	49052	44.81%
spect	1701	929	45.39%
voting-records	3453	1323	61.69%
monks problems	1373	798	41.88%
molecular	137222	119287	13.07%
car	6820	4120	39.59%

Table 5. Clustering with 32 cluster centers

Data set	Rand. Init. *SSE*	New. Init. *SSE*	Decreased Percent
mushroom	70874	34017	52.00%
molecular	137477	126643	7.88%
car	6381	3635	43.03%

Since cluster number is set as 16 in Table 4, hepatitis and zoo data sets are eliminated. In Table 5, only three data sets satisfy the condition $c^2<n$. The comparisons of *SSE* values in two tables indicate that the new method is a more efficient one.

3.2 Stability Test and Analysis

In the stability test, we implement the two distinct initialization methods in all data sets we mentioned above with pre-requisite clusters number. They have been tested for ten times and the standard deviations (SD) of *SSE* are listed below.

Table 6. Standard deviation of *SSE* in ten times

Data set	Method	SD		Dataset	Method	SD
Credit	Random	270		Lung Cancer	Random	69
	New	*139*			New	*23*
Hepatitis	Random	134		Molecular	Random	1834
	New	*76*			New	*375*
Mushroom	Random	7023		Car	Random	360
	New	*3451*			New	*67*
Spect	Random	169		Soybean	Random	76
	New	*65*			New	*26*
Voting records	Random	148		Zoo	Random	44
	New	*126*			New	*37*

It can be observed from table 6 that the standard deviations of the new method are smaller than those of the random one. It reflects that the *SSE* values of the new method are stable and vary little in clustering different data sets.

3.3 Time Consumption Test and Analysis

Through the two experiments, we know that the new initialization method is efficient and stable. If the time cost of our method exponentially not linearly increases with the volume of data sets, the new method should be limited within the scope of small data sets. In order to demonstrate the trend of variety, connect-4 data set in UCI [13] is implemented in this test.

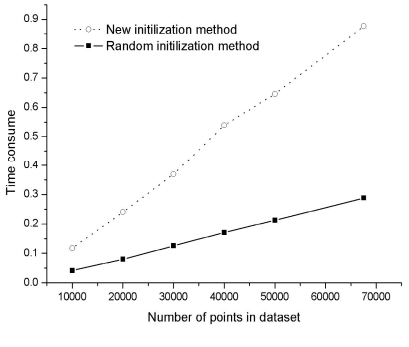


Fig. 4. Time consumption with 3 clusters

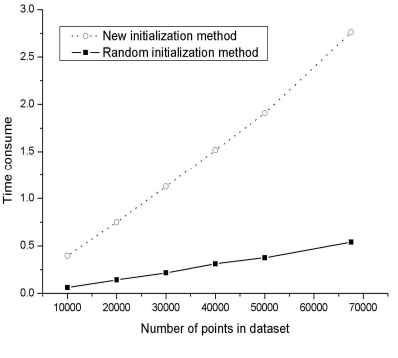


Fig. 5. Time consumption with 8 clusters

In Fig. 4, we set the cluster number 3 which was provided by the data sets donators. With the number of points increasing from 10,000 to 67,520, time consumptions of the random method and new one both are linearly increased. The time consumption of the random method in our method is also linear, since it includes *SSE* computing step besides random choice. The time cost of new method still linearly increases with the volume of data set when cluster numbers are 8 in Fig. 5. The experiment results show that the new initialization method can be implemented to cluster large date sets.

4 Conclusions

In this paper, we propose a new initialization method for categorical data clustering and implement it on fuzzy k -modes algorithm. Firstly, the new method reduces time cost though random sample process. Then we define point's density and point's probability to be a center in order to find the most potential centers. In order to avoid an unstable situation, the new method sub-sample many times. To attain more reliable initial centers, refinement method clusters in some choosing centers, thus no need to cluster the whole data set. The experiment result also indicates that new initialization method is efficient to attain more effective and stabler results than the random one. Increasing with the volume of data set, time consumption increases linearly.

There are some issues that need to be explored in order to enhance the performance of the new initialization method. We plan to propose a more effective way to compute the probability of point to be a center. We also plan to extend this method to textual data set, mixed data set and numerical data set, after changing the concept of density.

References

1. J.C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New Work, 1981.
2. Z. Huang, M.K. Ng, A fuzzy K -Modes algorithm for clustering categorical data, IEEE Trans. Fuzzy Syst. 7 (4):446-452, 1999.
3. Guha, S., Rastogi, R., & Shim, K. ROCK: A robust clustering algorithm for categorical attributes. In Proc. of 15th international conference on data engineering. pp. 512-521, 1999.
4. Barbara, D., Couto, J., & Li, Y. COOLCAT: An entropy-based algorithm for categorical clustering. In Proc. of the eleventh international conference on Information and knowledge management. pp. 582-589, 2002.
5. E. Forgy. Cluster analysis of multivariate data: efficiency VS. interpretability of classification. In WNAR meetings, Univ. of Calif. Riverside. number 768, 1965.
6. Duda, R.O. and Hart, P.E.. Pattern Classification and Sciene analysis. John Wiley and Sons, NY. 1973.
7. L. Kaufman and Rousseeuw. Finding Groups in Data: an Introduction to Cluster Analysis. Wiley, New York, 1990.
8. Bradley, P.S., and Fayyad, U.M. Refining initial points for k -means clustering. In 15th Internet. Conf. on Machine Learning. pp. 91-99, 1998.
9. Ying Sun, Qiuming Zhu, and Zhengxin Chen. An iterative initial-points refinement algorithm for categorical data clustering. Pattern Recognition letters 23:875-884, 2002.

10. Shu Wu, Qingshan Jiang. A novel fuzzy cluster validity index with new compositions. Proc. of the 6th World Congress on Control and Automation. pp. 5967-5971, 2006.
11. J.C. Bezdek. Pattern Recognition in Handbook of Fuzzy Computation. IOP Publishing Ltd., Boston, N.Y., 1998.
12. M. Meila and D. Heckerman. An experimental comparison of several clustering methods. Microsoft Research Technical Report MSR-TR-98-06, Redmond, WA., 1998.
13. C.L. Blake, C.J. Merz, UCI Repository of machine learning databases. http://www.ics.uci. edu/~mlearn/MLRepository.html, 1989.

L0-Constrained Regression for Data Mining

Zhili Wu and Chun-hung Li

Department of Computer Science
Hong Kong Baptist University
Kowloon Tong, Hong Kong
{vincent,chli}@comp.hkbu.edu.hk

Abstract. L2 and L1 constrained regression methods, such as ridge regression and Lasso, have been generally known for their fitting ability. Recently, L0-constrained classifications have been used for feature selection and classifier construction. This paper proposes an L0 constrained regression method, which aims to minimize both the epsilon-insensitive fitting errors and L0 constraints on regression coefficients. Our L0-constrained regression can be efficiently approximated by successive linearization algorithm, and shows the favorable properties of selecting a compact set of fitting coefficients and tolerating small fitting errors. To make our L0 constrained regression generally applicable, the extension to nonlinear regression is also addressed in this paper.

1 Introduction

In dealing with a usual regression task, we have a data matrix $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ in size $n \times d$, where n is the number of points (observations), and d is the dimension of each data vector. Correspondingly, we have a response vector $\mathbf{Y} = \{y_i\}_{i=1}^n$ in length n. A linear regression can be regarded as learning a coefficient vector $\mathbf{w}$ in length d and an offset constant b, such that

$$\mathbf{X}\mathbf{w} - b\mathbf{e} \approx \mathbf{Y}.$$

The ordinary least squares (OLS) learns the coefficient vector $\mathbf{w}$ by minimizing the residual squared loss

$$\min \quad ||\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y}||_2^2,$$

where $\mathbf{e}$ is a vector of ones.

However, OLS may not be accurate and robust enough. As a technique for improving OLS, ridge regression adds the L2 norm of the coefficient vector into the OLS objective function, hereby sets up an L2 constrained regression:

$$\min \quad ||\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y}||^2 + \lambda||\mathbf{w}||_2^2.$$

To obtain better prediction accuracy and interpretation [1], Lasso (least absolute shrinkage and selection operator) has been developed, which in terms of objective function substitutes the L2 norm of the coefficient vector in ridge regression for L1 norm:

$$\min \quad ||\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y}||^2 + \lambda||\mathbf{w}||_1^1,$$

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 981–988, 2007.

where the L1 norm of $\mathbf{w}$ can be calculated by summing up the absolute values of the entries of $\mathbf{w}$: $||\mathbf{w}||_1^1 = \sum_{i=1}^{d} |w_i|$. More generally, the Lp norm $||\mathbf{w}||_p^p, p > 0$, represented by $\sum_{i=1}^{d} |w_j|^p$, is mentioned in [2, 1, 3, 4], Among different norms, when $p \to 0$, the $||\mathbf{w}||_0^0$ is defined as the cardinality of nonzero elements of the coefficient vector $\mathbf{w}$ [4], and corresponds to selecting a subset of coefficients [1]. Generally speaking, the rationale of using a small p,e.g $p \leq 1$ is coefficient shrinkage and selection, but due to the computational difficulty, the cases of $p < 1$ for regression has seldom been implemented in practice.

Other than constraining coefficients, a new class of regression algorithms aim to incorporate alternative loss functions. For instance, support vector regression [5], instead of employing the squared loss in OLS, lasso and ridge regression, suggests the ε-insensitive loss ($\varepsilon \geq 0$) for each point $\mathbf{x}_i$

$$(|\mathbf{x}_i\mathbf{w} - b - y_i| - \varepsilon)_+ = \max\{0, |\mathbf{x}_i\mathbf{w} - b - y_i| - \varepsilon\}.$$

The ε-insensitive loss does not count any loss below ε, that is, whenever the absolute prediction loss for the $i-$th point x_i is smaller than ε, the loss is neglected and replaced by a zero value during total loss calculation.

This paper is motivated by seeking the set-up of $p \to 0$ in constraining regression coefficients, so as to select a good subset of coefficients and to obtain an easily interpretable regression model. Intrinsically the L0-constrained regression is companied with computational difficulties, but thanks to the study on L0-constrained classification [3, 6] and support vector regression, we can devise a type of regression algorithm to enforce L0-constraints upon coefficients and $\varepsilon-$insensitive loss for prediction, which can be efficiently approximated by successive linear programming.

This paper is organized as follows. In section 2, we explain our L0-constrained $\varepsilon-$insensitive regression and demonstrate how it relates to other works in literatures. And section 3 shows how to approximate the solution through Successive Linearization Algorithm. In section 4, the properties of L0-constrained regression is studied through experimenting on the prostate cancer data. Section 6 intends to discuss the extension of L0 constraints to nonlinear regression, together with simulation. Section 7 presents summaries and concludes the paper with future works.

2 L0-Constrained ε-Insensitive Regression

This section shows the formulation of our L0 constrained $\varepsilon-$insensitive regression. It considers the $\varepsilon-$insensitive loss seen in support vector regression, with the L0 norm of coefficients in linear models taken into account too. These two parts can be integrated into a single objective function, which is expected to be simultaneously minimized:

$$\min : \quad \mathbf{e}^T(|\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y}| - \varepsilon\mathbf{e})_+ + \lambda||\mathbf{w}||_0^0.$$

In [6] the L0 constraints $||\mathbf{w}||_0^0$ is approximated in the following way,

$$||\mathbf{w}||_0^0 \approx \mathbf{e}^T(\mathbf{e} - e^{-\alpha|\mathbf{w}|}), \alpha > 0,$$

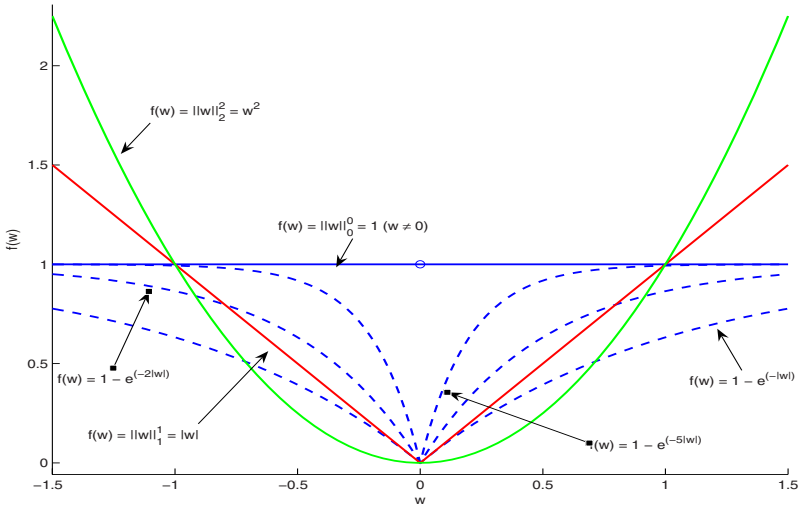


Fig. 1. Lp ($p = 0, 1, 2$) and the approximation to L0, with a single element in $\mathbf{w}$

where α is a positive tuning parameter and a larger α makes a closer approximation. And $|\mathbf{w}|$ converts the coefficients in $\mathbf{w}$ to their absolute values. Fig 1 shows how the $||\mathbf{w}||_0^0$ is approximated, and also how the L0 norm is different from L1 or L2 criteria. Through this approximation, and let $|\mathbf{w}| = \mathbf{v}, v_i \geq 0, \forall i$, the part of objective $||\mathbf{w}||_0^0$ is approximated by

$$\begin{aligned} ||\mathbf{w}||_0^0 &\approx \mathbf{e}^T(\mathbf{e} - e^{-\alpha\mathbf{v}}) \\ \text{subject to } \mathbf{w} &\leq \mathbf{v}, -\mathbf{w} \leq \mathbf{v} \\ \mathbf{v} &\geq 0 \end{aligned}$$

On the other hand, as a standard technique in support vector regression, the $\varepsilon-$ insensitive loss can be converted into the following form through introducing nonnegative vectors $\boldsymbol{\epsilon}$ and $\boldsymbol{\epsilon}^*$,.

$$\begin{aligned} \mathbf{e}^T(|\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y}| - \varepsilon\mathbf{e})_+ &\equiv (\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^*) \\ \text{subject to } \mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y} &\leq \varepsilon\mathbf{e} + \boldsymbol{\epsilon} \\ -(\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y}) &\leq \varepsilon\mathbf{e} + \boldsymbol{\epsilon}^* \\ \boldsymbol{\epsilon}, \boldsymbol{\epsilon}^* &\geq 0 \end{aligned}$$

Hereby the combination of these two parts has the following form

$$\begin{aligned} \min: \quad & \mathbf{e}^T(\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^*) + \lambda\mathbf{e}^T(\mathbf{e} - e^{-\alpha\mathbf{v}}) \\ \text{subject to } & \mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y} \leq \varepsilon\mathbf{e} + \boldsymbol{\epsilon} \\ & -(\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y}) \leq \varepsilon\mathbf{e} + \boldsymbol{\epsilon}^* \\ & \mathbf{w} \leq \mathbf{v}, -\mathbf{w} \leq \mathbf{v} \\ & \boldsymbol{\epsilon}, \boldsymbol{\epsilon}^*, \mathbf{v} \geq 0 \end{aligned}$$

3 Algorithm

In [6], L0 constrained linear support vector classification is proposed, and then successive linearization algorithm (SLA) is used to approximate the solution.

Our L0 constrained regression adopts ε-insensitive loss, rather than the hinge loss typical in classification, hereby leads to a different objective function. However, the converted objective form with linear constraints still makes the utilization of SLA possible. We summarize the algorithm for our L0 constrained regression using SLA as follows.

Algorithm: start with $\mathbf{v}^i, i = 0$ (e.g. via randomization), solve the following linear programming, hereby determine $(\mathbf{w}^{i+1}, b^{i+1}, \boldsymbol{\epsilon}^{i+1}, \boldsymbol{\epsilon}^{*(i+1)}, \mathbf{v}^{i+1})$:

$$\begin{aligned} &\min \mathbf{e}^T(\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^*) + \lambda\alpha(e^{-\alpha\mathbf{v}^i})^T(\mathbf{v} - \mathbf{v}^i) \\ \text{subject to} \quad &\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y} \le \varepsilon\mathbf{e} + \boldsymbol{\epsilon} \\ &-(\mathbf{X}\mathbf{w} - b\mathbf{e} - \mathbf{Y}) \le \varepsilon\mathbf{e} + \boldsymbol{\epsilon}^* \\ &\mathbf{w} \le \mathbf{v}, -\mathbf{w} \le \mathbf{v} \\ &\boldsymbol{\epsilon}, \boldsymbol{\epsilon}^*, \mathbf{v} \ge 0 \end{aligned}$$

After obtaining $\mathbf{v}^{i+1}$, keep solving the above linear programming till the maximum number of iterations is reached or the following stopping condition is satisfied:

$$|\mathbf{e}^T(\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^* - \boldsymbol{\epsilon}^i - \boldsymbol{\epsilon}^{*(i)}) + \lambda\alpha(e^{-\alpha\mathbf{v}^i})^T(\mathbf{v} - \mathbf{v}^i)| \le tol,$$

where tol is a very small constant.

4 Experiment on Prostate Cancer Example

We test the L0 constrained regression on the prostate data, which has been used in [1]. It has 97 samples, each is formed by eight features: log(cancer volume):*lcavol*, log (prostate weight):*lweight*, age, log(benign prostatic hyperplasia amount):*lbph*, seminal vesicle invasion:*svi*, log(capsular penetration): *lcp*, Gleanson score:*gleason* and percentage Gleason scores 4 or 5:*pgg45*. The aim is to regress them into log(prostate specific antigen) (lpsa). Following the preprocessing procedures in [1], each feature is normalized to be with zero mean and unit standard deviation, and the responses are also set to be with zero mean.

Fig 2 shows how the coefficients change along with the tuning parameter λ. As shown in the figure, when λ is larger than 2, the lcp, gleanson and pgg45 starts fading out of the regression model. Later the age and lbph features are dropped out too. Though our L0 regression has not been verified to possess exact piecewise solution paths as lasso does, the figure shows the shrinkage of coefficients as the tuning parameter λ increases. Useful features are likely retained for a long time, such as the lcavol, lweight and svi features. Particularly the lcavol feature presents a strong contribution to the regression model.

Table 1 lists several related regression approaches together for this prostate task. Both the least square and linear support vector regression build the regression model

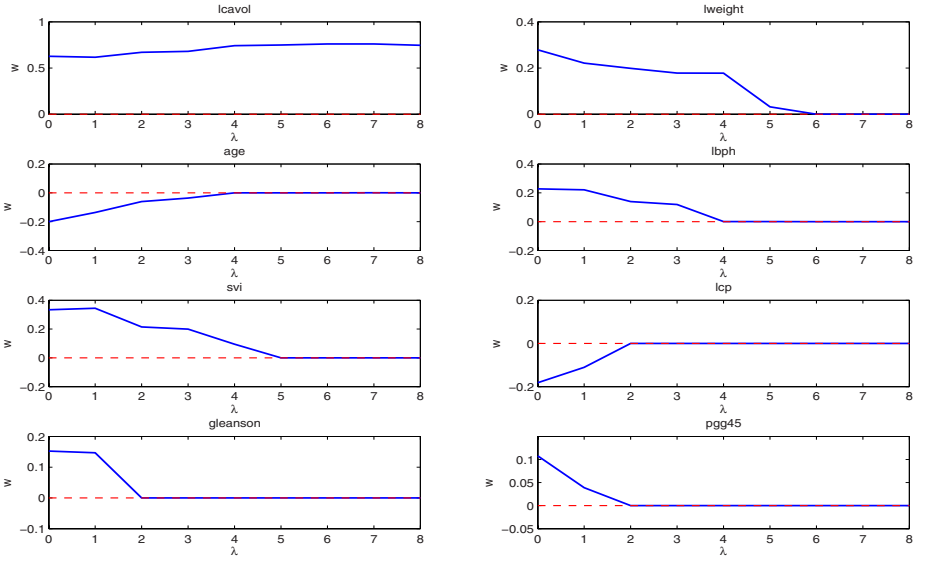


Fig. 2. L0 constrained regression on prostate data: each subplot shows the change of a regression coefficient when λ in L0 set-up changes from 0 to 8. $(tol = 1e-5, \alpha = 5, \varepsilon = 0.1)$.

upon all of the eight features. The L0 regression exerting a small constraint ($\lambda = 1$) on coefficients also leads to a model using all features. The least square regression undoubtedly minimizes the mean squared error, but support vector regression and L0 regression win with smaller absolute errors since by setup they intend to minimize absolute errors. As shown in the lower part of the table, Lasso following the parameter setting in [1] selects three features (lcavol, lweight,svi). Same features are selected by our L0 regression approach, with different coefficients and slightly better errors.

Table 1. Prostate cancer example

Method	mean-squared error	mean-absolute error	no. of non-zero coefficients
Least squares	0.4553	0.5176	8
Support vector regression	0.4671	0.5074	8
L0 regression ($\lambda = 1$)	0.4669	0.5068	8
Lasso ($s = 0.44$ [1])	0.5604	0.5847	3(lcavol, lweight,svi)
L0 regression ($\lambda = 5$)	0.5214	0.5659	3(lcavol, lweight,svi)

5 Extension to Nonlinear Regression

It has been believed that linear regression models might not suffice for tasks characterized with nonlinearity. Our L0 constrained regression mentioned above is developed for

linear models, by obtaining an explicit coefficient vector $\mathbf{w}$ for data points $\mathbf{X}$ in the input space. However, it can also be extended to the case of nonlinear regression. Inspired by kernel methods, here we present a type of L0 constrained nonlinear regression.

Assuming the (implicit) data representation now is $\mathbf{X} = \{\phi(\mathbf{x}_i)\}_{i=1}^n$, where $\phi(\cdot)$ comprises a nonlinear mapping. Explicitly we have a kernel matrix induced from the inner product of data matrix, that is, $\mathbf{K} = \mathbf{X}\mathbf{X}^T$. Upon building a regression model $\mathbf{X}\mathbf{w} - b\mathbf{e}$, we assume the coefficient vector $\mathbf{w}$ comes from a linear combination of data points, that is,

$$\mathbf{w} = \mathbf{X}^T\boldsymbol{\beta},$$

where $\boldsymbol{\beta}$ is vector in length n. Hereby the linear model is equivalent to $\mathbf{X}\mathbf{w} - b\mathbf{e} = \mathbf{K}\boldsymbol{\beta} - b\mathbf{e}$.

Hereby we can propose the following L0 constrained regression,

$$\min : \quad \mathbf{e}^T(|\mathbf{K}\boldsymbol{\beta} - b\mathbf{e} - \mathbf{Y}| - \varepsilon\mathbf{e})_+ + \lambda||\boldsymbol{\beta}||_0^0.$$

The constraints now affect $\boldsymbol{\beta}$ rather than $\mathbf{w}$. The switch of consideration is because $\mathbf{w}$ in feature space may be too long to be explicitly emulated, hereby controllably enforcing entries in $\mathbf{w}$ to be zero can hardly be done. Instead, restricting the L0 norm of $\boldsymbol{\beta}$ may lead to a small set of points that are then linearly combined into $\mathbf{w}$, hereby achieves the effect of introducing a compact model. This approach is also suggested in [7, 8], though they adopt L1 restriction on $\boldsymbol{\beta}$.

Similarly, this L0-constrained nonlinear regression can be further rewritten into the following form, which can also be approximated by SLA too.:

$$\begin{aligned} \min : \quad & \mathbf{e}^T(\boldsymbol{\epsilon} + \boldsymbol{\epsilon}^*) + \lambda\mathbf{e}^T(\mathbf{e} - e^{-\alpha\mathbf{v}}) \\ \text{subject to} \quad & \mathbf{K}\boldsymbol{\beta} - b\mathbf{e} - \mathbf{Y} \le \varepsilon\mathbf{e} + \boldsymbol{\epsilon} \\ & -(\mathbf{K}\boldsymbol{\beta} - b\mathbf{e} - \mathbf{Y}) \le \varepsilon\mathbf{e} + \boldsymbol{\epsilon}^* \\ & \boldsymbol{\beta} \le \mathbf{v} \\ & -\boldsymbol{\beta} \le \mathbf{v} \\ & \boldsymbol{\epsilon}, \boldsymbol{\epsilon}^*, \mathbf{v} \ge 0 \end{aligned}$$

To verify our new nonlinear L0-constrained regression, we apply it to a synthetical data set originated from [7]. The response is a nonlinear function of the one-dimensional data point x: $y_i = \cos(x)\cdot(\sin(5x)+\sin(4x))+1+\sigma\nu$, where ν is normally distributed random variable, with standard deviation $\sigma = 0.01$. The kernel used is Gaussian RBF, with $\gamma = 5$. Figure 3 shows the results by support vector regression and nonlinear L0 regression. Both build a nonlinear regression model by only utilizing a portion of data points as highlighted by square symbols. Our nonlinear L0 regression has the advantage of leading to a smaller number of non-zero β coefficients (They are called support vectors in SVR).

Table 2 tests this synthetic tasks under a series of parameter settings for 20 repeats, which further confirms the compactness of L0 regression models. In addition, it can be noticed that the linear kernel based ridge regression performs significantly worse than all of the three RBF kernel-based nonlinear regression methods. As the λ decreases (or equivalently increasing the C in SVR), the MSE drops gradually. When the SVR converges to a stable model of 6.6 support vectors on average, the L0 still keeps refining its

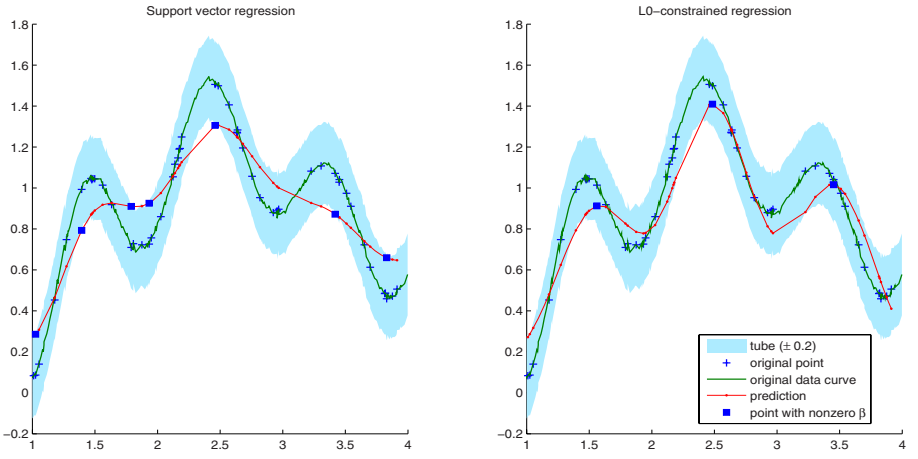


Fig. 3. SVR and L0 constrained regression on nonlinear synthetical data: $\lambda = 0.05$ (where C in SVR is equivalently set to $\frac{1}{\lambda} = 20$), and $\varepsilon = 0.2$. training:testing = $50 : 250$.

Table 2. Nonlinear L0 regression ($\varepsilon = 0.1$)

	Kernel Ridge Regression		L0 regression		SVR	
$\lambda(1/C)$	MSE (Linear Kernel)	MSE (RBF)	MSE	no. of nonzero β	MSE	no. of SVs
4.0	0.1137	0.0809	0.0595	1.3	**0.0247**	10.4
3.0	0.1106	0.0576	0.0332	2.6	**0.0236**	9.3
2.0	0.1073	0.0354	**0.0179**	3.8	0.0225	7.8
1.0	0.1043	0.0153	**0.0118**	4.5	0.0213	6.6
0.9	0.1040	0.0135	**0.0103**	4.9	0.0213	6.6
0.8	0.1038	0.0117	**0.0103**	4.9	0.0213	6.6
0.7	0.1036	0.0100	**0.0091**	4.9	0.0213	6.6
0.6	0.1034	0.0083	**0.0076**	5.3	0.0213	6.6

regression models. The RBF kernel ridge regression also keeps improving performance, but it unlike the SVR or L0 regression usually cannot result in a sparse model.

6 Summary and Future Work

This paper presents a new regression algorithm, which considers both the $\varepsilon-$insensitive loss and the number of regression coefficients. The integrated objective can be approximated by successive linearization algorithm. Experimental investigation shows it has the ability of selecting a small set of coefficients, so as to reduce the model size and often bring accuracy gain.

To make our L0 more generally applicable to nonlinear regression, we introduce the kernel trick into the L0 set-up, with deduction it also leads to a task to be handled by SLA. Experiments show that the nonlinear L0 regression usually gives a smaller

nonlinear model than support vector regression. When the parameter λ is small, it can even achieve more accurate models.

The linear programming is the main step of SLA. Although current computers can handle a moderately large size linear programming, and the number of succession steps in our SLA are often less than ten, more efforts should be put for quicker and more stable numerical methods.

References

[1] Tibshirani, R.: Regression shrinkage and selection via the lasso. In: Regression shrinkage and selection via the lasso. Technical report, University of Toronto, June 1994. (1994)
[2] Hastie, T., Mallows, C.: A statistical view of some chemometrics regression tools (with discussion). Technometrics **35**(2) (1993) 140–143
[3] Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: Proc. 15th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (1998) 82–90
[4] Weston, J., Elisseeff, A., Scholkopf, B., Tipping, M.: The use of zero-norm with linear models and kernel methods. In: The use of zero-norm with linear models and kernel methods. JMLR, to appear. (2002)
[5] Smola, A., Schoelkopf, B.: A tutorial on support vector regression. In: A tutorial on support vector regression, NeuroCOLT2 Technical Report NC2-TR-1998-030, 1998. (1998)
[6] Fung, G.: The disputed federalist papers: Svm feature selection via concave minimization. In: TAPIA '03: Proceedings of the 2003 conference on Diversity in computing, New York, NY, USA, ACM Press (2003) 42–46
[7] Smola, A., Scholkopf, B., Ratsch, G.: Linear programs for automatic accuracy control in regression. In: In Proceedings ICANN'99, Int. Conf. on Artificial Neural Networks, Berlin, 1999. Springer. (1999)
[8] Weston, J., Gammerman, A., Stitson, M., Vapnik, V., Vovk, V., Watkins, C.: Support vector density estimation. In: Advances in Kernel Methods — Support Vector Learning, pages 293–306, Cambridge, MA, 1999. MIT Press. (1999)

Application of Hybrid Pattern Recognition for Discriminating Paddy Seeds of Different Storage Periods Based on Vis/NIRS

Li Xiaoli, Cao Fang, and He Yong

College of Biosystems Engineering and Food Science, Zhejiang University, 310029, Hangzhou, China
yhe@zju.edu.cn

Abstract. Hybrid pattern recognition was put forward to discriminate paddy seeds of four different storage periods based on visible/near infrared reflectance spectroscopy (Vis/NIRS). The hybrid pattern recognition included extracting feature and building classifier. A total of 210 samples of paddy seeds, which belonged to four classes, were used for collecting Vis/NIR spectra (325-1075 nm) using a field spectroradiometer. The hybrid pattern recognition was integrated with wavelet transform (WT), principal component analysis (PCA) and artificial neural networks (ANN) models. WT was used to eliminate noises and extract characteristic information from spectral data. The characteristic information could be visualized in principal components (PCs) space, in which the structures correlative with the storage periods could be discovered. The first eight PCs, which accounted for 99.94% of the raw spectral data variance, were used as input of the ANN mode, and the model yielded high discrimination accuracy rates of 100%, 100%, 100% and 90% for four classes' samples respectively.

1 Introduction

Paddy sustains two-thirds of the world's population, so the paddy quality is very important. During the storage, a number of physicochemical and physiological changes occur, which is usually termed aging. These changes that include pasting properties, colour, flavour, and composition affect rice quality directly [1]. So a quick, accurate and nondestructive way is needed to classify the paddy seeds of different storage periods. Near infrared spectroscopy is the electro-magnetic wave from 780 to 2526 nanometers. The analytical capabilities of NIRS rely on the broad and repetitive absorption bands of carbon-hydrogen, oxygen-hydrogen, and nitrogen-hydrogen bands. It reflects the information of the structure, composition and state information of molecule. The overlapping of absorption bands makes direct interpretation of absorption spectra difficultly. So pattern recognition will play a more important role in the Vis/NIR technique development. Coffee varieties [2], melon genotypes [3], apple varieties [4], waxberry varieties [5], were classified using pattern recognition tools based on near infrared spectroscopy technique. However, few researches focused on discriminating the paddy seeds of different storage periods based on Vis/NIR spectroscopy technique.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 989–996, 2007.

NIR spectroscopy technique has been effectively combined with pattern recognition tools, such as multiple partial least squares (MPLS), principal component analysis [6][7] and discriminant analysis for classification, discrimination and authentication purposes. Linear discriminant analysis (LDA) [8] [9] failed with many variables, common solutions are to reduce the dimension of the predictor matrix by using data compressed arithmetic and then apply LDA. Wavelet transform is a very effective way to extract the useful information from mass spectral data [10] [11]. PCA can visualize the variability in a dataset, which can lead to the discovery of unknown structures [6][7]. In qualitative and quantitative analysis, artificial neural networks are more and more widely applied during the past several years [12]. Compared with SIMCA, PLS, DPLS, QDA and LDA et al method, the advantage of ANN is its anti-jamming, anti-noise and robust nonlinear transfer ability [12] [13]. But the shortcoming of ANN is difficult to be convergent when the input data are too mass. So the spectral data must be compressed as low-dimension data before ANN.

Inspired by this, we presented a novel pattern recognition tool for differentiating the paddy seeds of different storage periods by integrating wavelet transform, principal component analysis, and artificial neural network models. The wavelet transform was used to eliminate noises and extract features from the spectra, and the features were visualized in PCs space by principal component analysis, then the PCs which were closely correlative with the classes of these samples were used as the input of an ANN model for discriminating the classes of samples with different storage periods.

2 Materials and Methods

2.1 Materials

A total of 210 samples of paddy seeds were prepared for this experiment, they were obtained from Grain Supply Center of Hangzhou, Zhejiang province, China. These samples were harvested in four consecutive years from 2002 to 2005. The corresponding samples have been stored for one year (OYS), two year (TWYS), three year (THYS) and four year (FYS) (Table 1). All these samples were stored without any chemical or biological preservative treatment.

Table 1. Detail of the samples in the research

Storage	Variety	Producing area	Storage area	No.
One year	early-indica type rice	Jiangxi,China	Hangzhou,China	54
Two year	early-indica type rice	Jiangxi,China	Hangzhou,China	51
Three year	early-indica type rice	Jiangxi,China	Hangzhou,China	52
Four year	early-indica type rice	Jiangxi,China	Hangzhou,China	53

2.2 Vis/NIR Spectra Collection

A Vis/NIR spectroradiometer (Handheld FieldSpec) was used to collect spectra from 325 to 1075 nm at 3.5 nm bandwidth. Then all data were interpolated to 1-nm intervals. The uniform glass vessel (diameter: d=60mm, height: h=14mm) was used to

load the paddy seeds, and the vessel was filled with samples. The spectroradiometer was fixed at 120 mm above the surface of the sample with the field of view (FOV) of 25º. The light source of a Lowell pro-lam 14.5 V Bulb tungsten halogen that could be used both in the visible and near infrared region was placed 100 mm above the sample surface. The angle between the incident light and the spectroradiometer detector was about 45º.

A 100 mm^2 thick Teflon® disk was used as the optical reference standard for the system. A reflectance (R) was calculated by comparing spectral energy reflected from the sample with the standard reference. In order to reduce the operating error, for each sample, three reflection spectra were taken at three equidistant rotation positions of approximately 120º around the center of the sample. For each reflection spectrum the scan number was 10 at exactly the same position, a total scan for each example was 30. Due to the imperfection in the own system, a big scattering affected the accuracy of measurement could be observed at the beginning and the end of the spectral data, so the first 75 and the last 75 wavelengths data were excluded in all analysis, all the considerations were based on this range of wavelengths. The absorbance spectra of the four classes can be seen in fig. 1.

2.3 Pattern Recognition Tools

The wavelet transform (WT) enables the signal (spectrum) to be analyzed as a sum of functions (wavelets) with different spatial and frequency properties [11]. The generated waveforms are analyzed with wavelet multiresolution analysis to extract subband information from the spectral signal. Principal component analysis (PCA) is a very effective data reduction technique for spectral data. It summarizes data by forming new variables, which are linear composites of the original variables [18]. In the study, the spectral data was analyzed by principal component analysis (PCA) and defective information was eliminated. Artificial neural networks (ANN) are known as useful tools for pattern recognition, identification, and classification. A neural network model can determine the input-output relationships for a complicated system. And such a model can provide data approximation and signal-filtering functions beyond optimal linear techniques [14].

PCA was performed using the Unscrambler 9.5 software (CAMO). The matlab Wavelet Toolbox was used to perform the standard wavelet transform using the predefined wavelet filters. The matlab Neural Networks Toolbox was used to build the back-propagation neural network model.

3 Results and Discussion

Fig.1 shows the average absorbance spectra of samples of four different storage periods: (a) paddy seeds of one year storage (OYS) (b) paddy seeds of two year storage (TWYS) (c) paddy seeds of three year storage (THYS) and (d) paddy seeds of four year storage (FYS). Seemingly, there isn't a remarkable difference among these four classes in these spectral range. After comparing in detail, some differences can be detected from 600 nm to 700 nm, which makes it possible to discriminate the samples with difference aging. The differences may be caused by the different internal

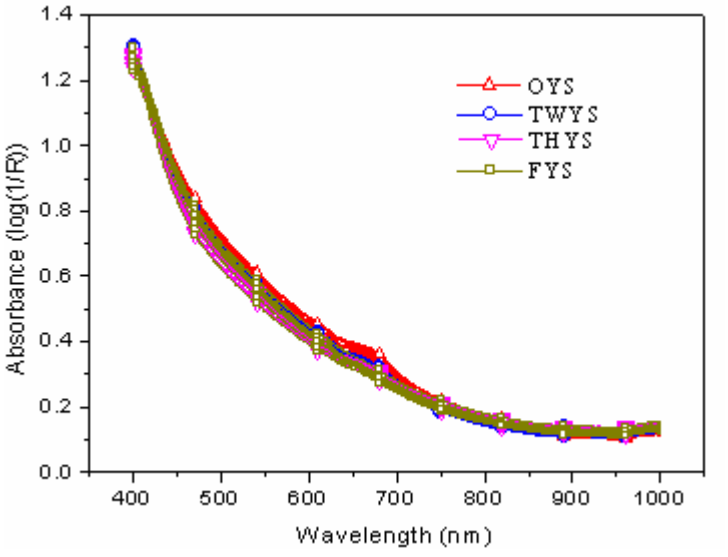


Fig. 1. Absorbance spectra of four classes' samples, OYS--paddy seeds of one year storage, TWYS--paddy seeds of two year storage, THYS--paddy seeds of three year storage, FYS--paddy seeds of four year storage

attribute of these samples, such as the starch and protein. The baseline drift in the spectra (shown in fig.1.) is mostly due to system noise, which can be eliminated by wavelet transform.

3.1 Noises Removal and Dimension Reduction by Wavelet Transform

In this research, the wavelet transform was used to eliminate noises and select features from the raw spectra. The WT was implemented using a dyadic filter tree. After trying, daubechies2 (db2) wavelet was selected as the suitable function to decompose the spectral signal. Then the spectral data, which have 210 rows and 601 columns, were decomposed at third level by db2 wavelet. To see the effect of WT, the signal was reconstructed with low-frequency coefficient (cA_3) and high-frequency coefficient (cD_1, cD_2, cD_3). The reconstructed signal based on wavelet coefficients at third level decomposition can be seen in fig.2. The first layer plot X is a raw spectra. The second layer plot cA_3' is the reconstructed signal based on low-frequency coefficient (cA_3). The third layer plot cD_3' is the reconstructed signal based on high-frequency coefficient (cD_3). The fourth layer plot cD2' is the reconstructed signal based on high-frequency coefficient (cD_2). The fifth layer plot cD_1' is the reconstructed signal based on high-frequency coefficient (cD_1). It can be seen that the signals cD_3', cD_2' and cD_1' contain mass high-frequency noise, especially at the beginning and end of this curve. The high-frequency coefficients contain mass noise and repeated information; it can barely give any help to classify the samples of different storage periods [15]. The signal cA3' reconstructed by low-frequency coefficient (cA_3) is very similar with the raw signal X. So the low-frequency coefficient (cA_3) (77-dimension) was used to replace the spectra.

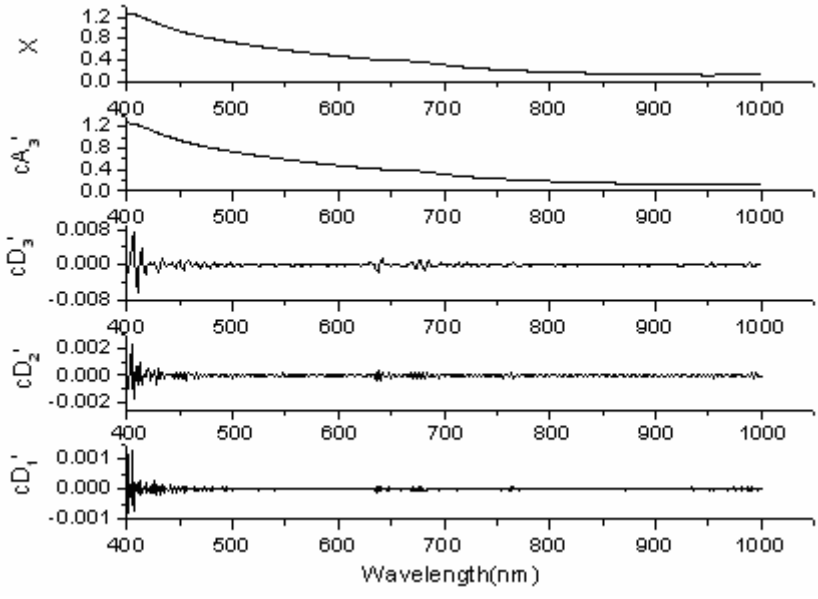


Fig. 2. Reconstruction signal based on wavelet coefficients at third level decomposition

3.2 Data Visualization by Principal Components Analysis

The principal component analysis aimed to mapping the wavelet coefficients in PCs space with the largest variability. PCA was performed on the 77-wavelet components of each sample, the dimensionality of the wavelet components was reduced from 77 to 20 by PCA, and hence 20 principal components could be achieved. If the scores of a principal component were organized according to the number of the sample, a new image could be created. The new image was then called 'PCA scores image'. The advantage of using principal components scores image was that it could display the clustering information of classes from multiple variables.

The scatter image of PC1 (variability, 84.4%) vs. PC2 (variability, 9.7%) vs. PC3 (variability, 4.2%) scores is shown in fig.3. The cumulative reliabilities of the first three principal components were 98.3%(seen in Table 2). In other words, the PC1,PC2 and PC3 accounted for 98.3% of the data variance. The image gives distributing information of four classes samples. In this scatter image, the four classes samples are closely clustered in strap shape respectively and the samples of four classes are composed as four well-defined groups. The differences among OYS samples, TWYS samples, THYS samples and FYS samples are pronounced. While, the boundaries of samples of four classes aren't clear in fig.1. It means that spectral diagnostic information can be showed clearer in PCs space than in raw spectral absorbance image. It can be concluded that the spectral diagnostic information can be extracted from raw spectra by WT and PCA. The extracted diagnostic information is strong correlation with storage period. But the TWYS and FYS samples are overlapped in the image. A more accuracy and clear separation need to be made. So, an artificial neural network algorithm was applied to classify the four classes with digital

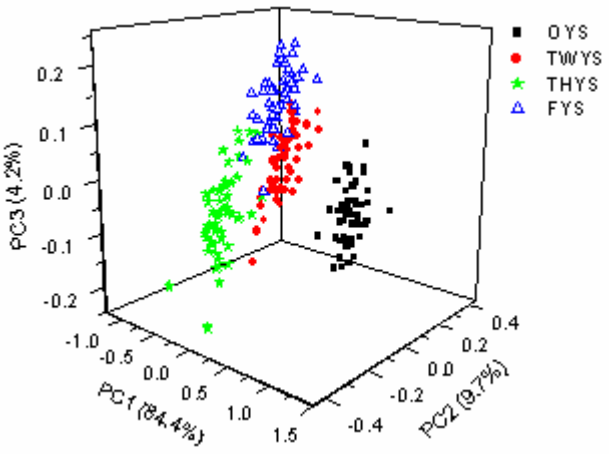


Fig. 3. Scatter plot of PC1vs PC2 vs PC3 scores of all samples

discriminative result. PCA shows that the cumulative reliabilities of the first 8 principal components are 99.9%. It means these components can explain 99.9% of the data variances, and the rest components don't give more useful information for detecting the classes.

3.3 Discriminating Samples by ANN

The design process of this BP neural network model consists of neural network topological architecture design, training data collection, neural network training and validation. A typical topological structure for a BP network consists of an input layer, at least one hidden layer and an output layer. For a BP network, a very important theorem is that a BP network with one hidden layer can approach any consecutive function in closed interval. The whole samples were separated randomly into two parts, the randomly selected 170 samples were used as calibration samples, and the remains 40 were used as prediction samples.

The first eight principal components from PCA were used as input vector. So the node of input layer was 8. As there were samples of four different classes, the output vectors of ANN were assumed to be binary system vectors of four bits. The node of output layer was 4. 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 0.001. The maximum time of training was set as 2000. The threshold value of error was 0.1. The number of neurons in hidden layers was optimized by "trial-and-error" method. It can be found that there was a smaller error and fewer training cycles when the node number of hidden layer was set as 8. A BP neural network model with three-layers was built. The optimal topology structure was 8-8-4 for three-layer neural network.

The discrimination results of calibration sample set and prediction sample set are summarized in table 2. The neural network yielded a very high discrimination accuracy, all of the OYS, TWYS and THYS samples were correctly classified for the

Table 2. Discrimination result of calibration sample set and prediction sample set of this model

Classes	Classification			Prediction		
	No.	FNo.	Ar	No.	FNo.	Ar
OYS	44	0	100%	10	0	100%
TWYS	41	0	100%	10	0	100%
THYS	42	0	100%	10	0	100%
FYS	43	0	100%	10	1	90%
Total	170	0	100%	40	1	97.5%

Note: OYS: paddy seeds of one year storage, TWYS: paddy seeds of two year storage, THYS: paddy seeds of three year storage, FYS: paddy seeds of four year storage, No.: number, FNo.: fault No., Ar: accuracy rate.

calibration and prediction sample sets respectively. The FYS paddy seeds were more difficult to classify in prediction. However, 90% of FYS samples were correctly classified in the prediction set. The total accuracy rate was 97.5% for all the four classes.

4 Conclusion

The hybrid pattern recognition obtained wonderful performance for discriminating paddy seeds of four different storage periods based on Vis/NIRS technique. The realization of hybrid pattern recognition needed two steps, the first step was noise removal and feature extraction, which was implemented by wavelet transform and principal component analysis. Subsequently, the characteristic spectral information was used as input of an ANN model. This model achieved a very good result for discriminating the four classes, which meant Vis/NIR spectroscopy could be used to classify paddy seeds of different storage periods non-destructively. In short, the hybrid pattern recognition has substantial potential for mining information from spectral data and discriminating different classes.

References

[1] Zhou, Z., Robards, K., Helliwell, S. and Blanchard, C. Ageing of Stored Rice: Changes in Chemical and Physical Attributes. Journal of Cereal Science 35 (2002) 65–78

[2] Esteban,I. D., Gonzalez, S.J.M. and 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

[3] Seregely, Z., Deak, T. and Bisztray, G. D. Distinguishing melon genotypes using NIR spectroscopy. Chemometrics and Intelligent Laboratory Systems 72 (2004) 195-203

[4] He, Y., Li, X. L. and 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 3809 (2005) 1053-1056

[5] He, Y. and Li, X. L. Discriminating varieties of waxberry using near infrared spectra. Journal of Infrared and Millimeter Waves 25(3) (2006) 192-194

[6] He, Y., Li, X. L. and Deng, X. F. Discrimination of varieties of Tea Using Near Infrared Spectroscopy by Principal Component Analysis and BP Model. Journal of Food Engineering 79 (2007) 1238–1242
[7] Pontes, M.J.C., Santos,S.R.B., Araujo, M.C.U., Almeida,L.F., Lima,R.A.C., Gaiao,E.N. and Souto,U.T.C.P. Classification of distilled alcoholic beverages and verification of adulteration by near infrared spectrometry. Food Research International 39 (2006) 182-189
[8] Osborne, B.G., Fearn, T. and Hindle,P.H. Practical NIR Spectroscopy. Longman, Harlow, UK, 1993
[9] Wu, B. Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. Bioinformatics 19 (2003) 1636-1643
[10] Vannucci, M., Sha, N.J. and Brown,P.J. NIR and mass spectra classification: Bayesian methods for wavelet-based feature selection. Chemometrics and Intelligent Laboratory System 77 (2005) 139-148
[11] Cocchi, M., Corbellini, M., Foca, G., Lucisano, M., Pagani, M.A., Tassi, L. and Alessandro U. Classification of bread wheat flours in different quality categories by a wavelet-based feature selection/classification algorithm on NIR spectra. Analytica Chimica Acta 544 (2005) 100-107
[12] Wang, D., Ran, M.S. and Dowell, F.E. Classification of damaged soybean seeds using near-infrared spectroscopy. Transaction of the ASAE 45 (2002) 1943-1948
[13] Dardenne, P. and Pierna, J.A.F. A NIR data set is the object of a chemometric contest at 'Chimiometrie 2004'. Chemometrics and Intelligent Laboratory System 80 (2006) 236-242
[14] Clifford, G. and Lau, Y. Neural Networks: Theoretical Foundations and Analysis. IEEE, New York, NY, USA, 1992.
[15] Chen, B., Huang, C.X. and Lu, D.L. Use of multi-resolution decomposition and principal components analysis in information abstraction from NIR spectrum. Journal of Jiangsu University (Natural Science Edition) 25 (2) (2004) 105-108.

Density-Based Data Clustering Algorithms for Lower Dimensions Using Space-Filling Curves*

Bin Xu and Danny Z. Chen

Department of Computer Science and Engineering
University of Notre Dame, Notre Dame, IN 46556, USA
{bxu,chen}@cse.nd.edu

Abstract. We present two new density-based algorithms for clustering data points in lower dimensions (dimensions ≤ 10). Both our algorithms compute density-based clusters and noises in $O(n)$ CPU time, space, and I/O cost, under some reasonable assumptions, where n is the number of input points. Besides packing the data structure into buckets and using block access techniques to reduce the I/O cost, our algorithms apply space-filling curve techniques to reduce the disk access operations. Our first algorithm (Algorithm A) focuses on handling not highly clustered input data, while the second algorithm (Algorithm B) focuses on highly clustered input data. We implemented our algorithms, evaluated the effects of various space-filling curves, identified the best space-filling curve for our approaches, and conducted extensive performance evaluation. The experiments show the high performance of our algorithms. We believe that our algorithms are of considerable practical value.

Keywords: Density-based clustering, secondary memory management, space-filling curves, multi-attribute hashing.

1 Introduction

Data clustering is a fundamental problem arising in many practical applications. In this paper, we consider the density-based data clustering problem on spatial data, defined as follows [7,3]. We are given a set S of n points in the d-D space $\mathbb{R}^d$ ($d \geq 2$ is a constant integer) and two parameters $\delta > 0$ and $\tau > 1$. For any point p in $\mathbb{R}^d$, we denote by $N_\delta(p)$ the sphere in $\mathbb{R}^d$ centered at p with radius δ (based on a given distance function such as an L_c metric). The sphere $N_\delta(p)$ is called the *δ-neighborhood* of p.

1. For a point $p \in S$, if there are at least τ points of S (including p) contained in the sphere $N_\delta(p)$, i.e., $|S \cap N_\delta(p)| \geq \tau$, then all points of $S \cap N_\delta(p)$ belong to the same cluster of S.
2. For two subsets C_1 and C_2 of S, if each of C_1 and C_2 belongs to a cluster and if $C_1 \cap C_2 \neq \emptyset$, then $C_1 \cup C_2$ belongs to the same cluster.
3. A *cluster* of S is a maximal set satisfying the two conditions above. All points of S that do not belong to any cluster are called *noise*.

* This research was supported in part by NSF Grants CCR-9988468 and CCF-0515203.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 997–1005, 2007.

The *density-based clustering (DBC) problem* is to identify all clusters of S and all noise of S. Let $S_\delta(p)$ denote the point set $S \cap N_\delta(p)$ for any point p in $\mathbb{R}^d$. A point $p \in S$ is called a *dense point* if $|S_\delta(p)| \geq \tau$; otherwise, $p \in S$ is a *sparse point*.

Considerable work on solving the DBC problem has been done. One of the most well known DBC algorithms is (G)DBSCAN [7]. (G)DBSCAN uses a spatial data structure, R^*-tree, to identify points within a distance δ from the dense points of the clusters. A heuristic algorithm for determining the parameters δ and τ was given in [7]. OPTICS [1] creates an augmented ordering of the database based on the local densities. DBCLASD [9] defines clusters based on the expected distribution of the distances to the nearest neighbors; hence no parameters are needed in the cluster search. FDC [10] defines clusters by an equivalence relationship on the objects in the database using a cell-based method. DENCLUE [5] models the overall point density analytically as the sum of the influence functions of the data points.

With the rapid increase in data volumes today, data sets are often too large to entirely fit in a computer's internal memory (i.e., the main memory), and instead must be stored in external storage devices (e.g., disks). In this paper, we consider using disk (or interchangeably, secondary memory or external memory, and they all mean the same thing) for data storage. A major performance bottleneck in this setting is the cost of input/output (I/O) communication between the external and internal memories, since such I/O operations are very time-consuming. One promising approach for dealing with this I/O difficulty is to design algorithms and data structures that bypass the virtual memory system and explicitly manage their own I/O. Vitter [8] considered this problem, and referred to such algorithms and data structures as *external memory algorithms and data structures* (see [8] for more details).

In this paper, we present the following results:

1. We developed two algorithms (called Algorithms A and B) that compute density-based clusters and noises in $O(n)$ CPU time, space, and I/O cost, under some reasonable assumptions, where n is the number of input points.
2. Besides packing data structures into buckets and using block access techniques to reduce the I/O cost as other commonly-used DBC algorithms, our algorithms apply space-filling curve techniques to further reduce the disk access operations. Both our algorithms are scalable to large data sets.
3. We evaluated the effects of various space-filling curves and identified the best space-filling curve for our algorithms based on both theoretical and experimental studies.
4. We implemented our algorithms and conducted extensive performance evaluation. We compared our two algorithms with DBSCAN [7], one of the current fastest DBC algorithms. Our experimental results showed that our algorithms are much more efficient when the dimension is no higher than 10. Since there are numerous important applications with lower dimension data sets, such as satellite images, molecular biology, and astronomy, our algorithms are expected to be of considerable practical value.

Like many other geometric algorithms, the CPU time and I/O bounds of Algorithms A and B have a constant factor of 3^d in the cluster search. Algorithms A and B are efficient for $d \leq 10$. Algorithm A focuses on handling not highly clustered input data, while Algorithm B focuses on highly clustered input data.

2 External Data Structures

We use hashing-based external data structures for both Algorithms A and B. There are three reasons for our choices of the external data structures: (1) They are cheap to construct; (2) they have low I/O cost for searching operations; (3) we can further reduce the I/O cost by applying space-filling curve techniques to these external data structures. Compared to using the general high dimension indexing methods based on space-filling curves to solve the density-based clustering, the external data structures and the cluster search algorithm in our algorithms are more tightly integrated to achieve high efficiency in both CPU time and I/O cost, and this is done at the cost of significant loss of generality of our external data structures.

2.1 Decomposing the Space

Given a set S of n points in $\mathbb{R}^d$ and parameters $\delta > 0$ and $\tau > 1$, we partition the space containing S into a set of cells, called the *basic cells*, so that for any point p of S in a basic cell c, we need to search only c's 3^d neighboring basic cells for the point set $S_\delta(p)$. The length of a basic cell in any dimension is 2δ.

To save disk space and reduce disk access operations, we need to store in one page the data points in many basic cells. Thus, we combine as many basic cells as possible into a larger cell, called *new cell*, such that the average number of points in each new cell is no bigger than θ, where θ is the capacity of one page. The decomposition of the space produces the smallest hypercube C that contains S, and the edge length of a new cell.

2.2 The Sequence Numbers of New Cells

We store all input points in each new cell as a collection. Clearly, we need to decide the order in which the new cells are to be stored. A space-filling curve helps create a mapping of the new cells from a d-D space to a single dimension (thus defining an order). It is desirable that the new cells that are close together in the d-D space be also close together in the mapped 1-D space [6]. Each new cell is mapped to a unique *new cell sequence number*, and we store input points based on this new cell sequence number.

We consider six space-filling curves [4,6]: Row-wise, snake row-wise, Z-ordering, Z-ordering with Gray code, Gray ordering, and Hilbert curve. We need to determine which curve is most suitable for our DBC algorithms. Due to the page limit, we refer to [4,6] for the details of these six space-filling curves.

Both our algorithms efficiently search the clusters by gradually "expanding" the clusters instead of "jumping" everywhere. Thus, we need to use a space-filling

curve which helps "expand" the clusters efficiently. The Hilbert curve has the least "jumps" comparing to other space-filling curves [6], and is the best choice for our cluster search algorithms, as shown by our experimental results.

The mapping from each new cell to its new cell sequence number based on any of the six space-filling curves mentioned above takes $O(1)$ CPU time. Thus, the overhead of this mapping is very small comparing to the savings in I/O costs.

2.3 External Data Structure of Algorithm A

Suppose there are totally N new cells. The first N data pages as shown in Figure 1(b) correspond to the N new cells, in the ascending order of the new cell sequence numbers. We call these N pages the *starting pages* because if one page cannot hold all input points in one new cell, we will allocate one or more *extra pages* for this new cell, and these N pages are always the starting pages for finding the extra pages. There are K extra pages in Figure 1(b). As shown in Figure 1(a), each data page contains the information of the row-wise sequence number of the corresponding new cell (say, c), the number of points in c, the points in c and their cluster IDs, and the pointer to c's next extra page.

We omit the construction of the external data structure of Algorithm A due to the page limit, and summarize it in the following lemma.

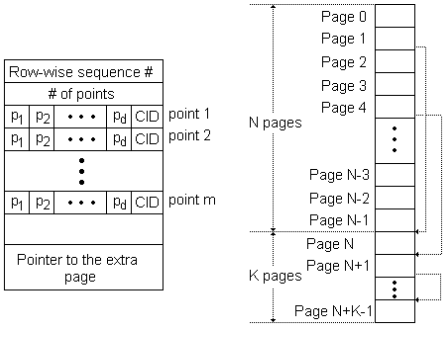


Fig. 1. Algorithm A: (a) The structure of one data page; (b) the data pages

Lemma 1. *Suppose we are given a set S of n points in the d-D space $\mathbb{R}^d$ (for any constant integer $d \geq 2$) and two parameters $\delta > 0$ and $\tau > 1$. By scanning the input data twice, we can create the data file for Algorithm A in $O(n)$ CPU time, space, and I/Os if there are $O(n)$ new cells and each new cell contains a constant number of points.*

2.4 External Data Structure of Algorithm B

The external data structure of Algorithm B has directory pages and data pages. The address of a new cell in the data pages can be located by consulting the directory.

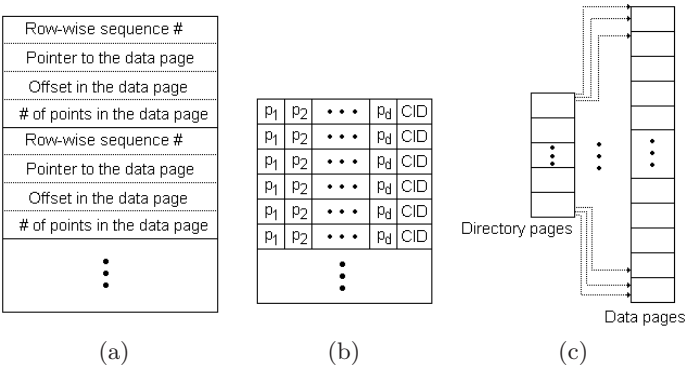


Fig. 2. Algorithm B: (a) A directory page, (b) a data page, and (c) directory pages and data pages

Figure 2(a) shows a directory page. Each new cell has an entry (called a *directory node* or *D-Node*) in directory. Each *D-Node* contains information of the row-wise sequence number of the corresponding new cell (say, c), the number of points in c, the pointer to the data page, and the offset in the data page. Figure 2(b) shows the structure of a data page. Figure 2(c) shows the external data structure of Algorithm B. New cells are saved consecutively along the ascending order of the new cell sequence numbers in the both directory pages and data pages. The space utilization is 100%, i.e., there are no useless directory and data pages.

We omit the construction of the external data structure of Algorithm B due to the page limit, and summarize it in the following lemma.

Lemma 2. *Suppose we are given a set S of n points in the d-D space $\mathbb{R}^d$ (for any constant integer $d \geq 2$) and two parameters $\delta > 0$ and $\tau > 1$. By scanning the input data three times, we can create the external data structure for Algorithm B in $O(n)$ CPU time, space, and I/Os if there are $O(n)$ new cells.*

3 Searching for Clusters

Below is the main procedure for cluster search in Algorithms A and B.

Input: External data structure F, and parameters δ and τ.
Output: External data structure F with cluster IDs assigned.

1. For each new cell c along the new cell number sequence, do the following:
   (a) Read c's 3^d neighboring new cells in $\mathbb{R}^d$, H, into the main memory.
   (b) Partition H into basic cells.
   (c) For each point $p \in c$, do the following:
      i. If p is a dense point then assign a cluster ID to all points in $N_\delta(p)$, and record the equivalent cluster IDs.
2. Eliminate the equivalent cluster IDs.

We search for clusters along the new cell sequence numbers in ascending order. In this way, we can reduce the swaps of data pages between the main memory and the disk, and operate on more consecutive pages. Given the new cell sequence number s of a new cell c, we can calculate the new cell sequence numbers of its neighboring new cells in $\mathbb{R}^d$ easily since we keep the row-wise sequence number of c in the external data structures of Algorithms A and B.

We use a "first-in-first-out" queue Q to record the new cells that are in the main memory, with the corresponding cell sequence numbers as keys. Q has a prescribed maximum length L. If we have more main memory, we can set L bigger to reduce the swap between the main memory and the secondary memory. When we read new cells into the main memory, we add new cells to Q. For those new cells that need to be read into the main memory, we first sort their new cell sequence numbers in ascending order, and then read them along this order. In this way, we can save much I/O cost by reading consecutive pages. When Q exceeds the length limit L, the new cells which came earlier will be removed, and these removed new cells will be written back to the secondary memory. We also use the ascending order of the new cell sequence to write back these new cells for reducing I/O cost.

We only need to search a point p's 3^d neighboring basic cells to find the point set $S_\delta(p)$. If p is a dense point, we need to assign all points in $N_\delta(p)$ to a same cluster. But, some of the points in $N_\delta(p)$ may have already been assigned different cluster IDs, while these points in fact should belong to the same cluster. We call different cluster IDs which are actually for the same cluster the *equivalent cluster IDs* [10]. We reduce the number of equivalent cluster IDs by doing the following. For a dense point p, if there is a point in $N_\delta(p)$ already having a cluster ID, we just use this ID (instead of creating a new ID) to label the points in $N_\delta(p)$. In this way, we can reduce a great deal of equivalent cluster IDs. We formulate the problem of eliminating equivalent cluster IDs as one of computing the connected components to completely eliminate any equivalent cluster IDs.

We have the following theorem as the summary of Algorithms A and B.

Theorem 1. *Given n points in $\mathbb{R}^d$ (for any constant $d \geq 2$) and parameters $\delta > 0$ and $\tau > 1$, Algorithms A and B can compute all density-based clusters and noises in $O(n)$ CPU time, space, and I/Os if there are $O(n)$ new cells and each new cell contains a constant number of data points.*

For an input data set, if we like to use different values of δ, Algorithms A and B need to reconstruct the external data structures. This overhead actually does not have a significant impact on the overall efficiency of Algorithms A and B since the time for constructing the external data structures is much less than the time for searching clusters, especially on large data sets.

4 Experimental Results

We conducted extensive experiments on our two DBC algorithms, using an Intel Pentium 4 (1.4 GHZ with 512M main memory, running MS Windows 2000).

For each data set, we apply the heuristic algorithm [7] to determine the parameters δ and τ. Note that in specific applications, we can adjust the values of δ and τ to find the types of clusters interesting to us, such as low density sparse clusters. The experiments have shown that our algorithms find the same clusters as DBSCAN when using the same values of δ and τ on input data sets. We used the library ANN [2] to generate the synthetic data sets in the following way: 15 "core" points were first chosen from the uniform distribution (on the interval $[0, 1]$) in the unit hypercube, and then many points based on a Gaussian distribution with standard deviation σ centered around each core point were generated in the unit hypercube. In the experiments below, we use two σ values to create two kinds of data sets: We use $\sigma = 0.05$ to create data sets that represent not highly clustered data, and $\sigma = 0.005$ to create data sets that represent highly clustered data. Due to the page limit, we omit the experiments with real data sets and big data sets, and those about relation between the execution time and dimension, etc.

For all the experiments below, the execution time of Algorithms A and B includes the time of both the external data structure construction and cluster search, since we need to reconstruct the external data structures when δ changes. The execution time of DBSCAN only includes the cluster search time.

4.1 Random Disk Access Operations and Space-Filling Curves

We use six different space-filling curves and record the total random disk access operations for cluster search. When several disk accesses are on consecutive data pages, we count them as one random disk access. This is because consecutive I/Os is much faster than the discrete I/Os. We use random disk accesses instead of disk accesses to better reflect the real I/O costs in cluster search.

In Figure 3, we use 2-D data sets of sizes from 10M to 40M with $\sigma = 0.005$. We can see that the random disk access operations in cluster search with the Hilbert space-filling curve are the smallest for all data sizes. When data size is 40M, other space-filling curves, Gray code, row-wise, snake row-wise, Z-ordering, and Z-ordering with Gray code, are 1.5, 1.1, 1.0, 0.7, and 0.6 times bigger than Hilbert, respectively.

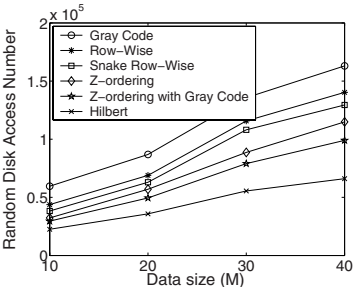


Fig. 3. Relation between the random disk access operation number of cluster search and data size in Algorithm B

Based on many experiments with data sets of different sizes, dimensions, and σ values, we conclude that the Hilbert curve is the best choice for Algorithms A and B on reducing the I/O costs. We use the Hilbert curve in all the experiments below.

4.2 Experiments with Different σ Values

Figure 4 shows the relation between the execution time and data size with different σ values. In Figure 4(a), the input data is not highly clustered ($\sigma = 0.05$). For data sizes of 1M, 2M, 3M, and 4M, DBSCAN uses 2.9, 4.5, 8.9, and 15.7 times as much time as Algorithm A, 1.4, 2.5, 4.9, and 9.3 times as much time as Algorithm B, respectively. For the data set of size 4M, DBSCAN needs 87,962 seconds (i.e., over 1 day) for the cluster search, while Algorithms A and B use 1.3 and 1.6 hours of execution time, respectively. Algorithms B uses around 1.8 times as much time as Algorithm A for the data sizes.

In Figure 4(b), the input data is highly clustered ($\sigma = 0.005$). For data sizes of 1M, 2M, 3M, and 4M, DBSCAN uses 1.7, 3.0, 5.9, and 11.1 times as much time as Algorithm A, 2.0, 3.6, 7.1, and 13.3 times as much time as Algorithm B, respectively. For the data set of size 4M, DBSCAN needs 89,543 seconds (i.e., over 1 day) for the cluster search, while Algorithms A and B use 2.2 and 1.8 hours of execution time, respectively. Algorithms A uses around 1.2 times as much time as Algorithm B for the data sizes.

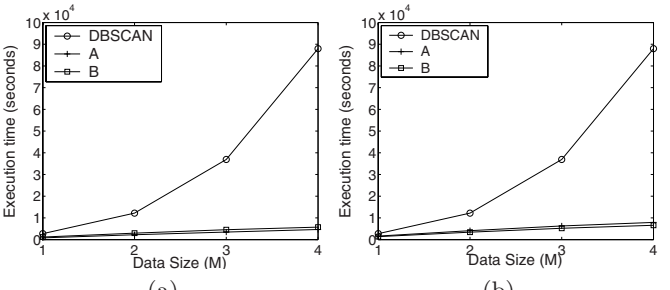


Fig. 4. Relation between the execution time and data size: (a) $\sigma = 0.05$; (b) $\sigma = 0.005$

References

1. M. Ankerst, M. Breunig, H.-P Kriegel, and J. Sander, OPTICS: Ordering points to identify clustering structure, *Proc. ACM SIGMOD Conf.*, 1999, 49-60.
2. S. Arya and D.M. Mount, ANN: A library for approximate nearest neighbor searching, *2nd CGC Workshop on Computational Geometry*, 1997. Also, see http://www.cs.umd.edu/∼mount/.
3. D.Z. Chen, M. Smid, and B. Xu, Geometric algorithms for density-based data clustering, *International Journal of Computational Geometry and Applications*, 15(3) (2005) 239-260.

4. C. Faloutsos, Gray codes for partial match and range queries, *IEEE Transactions on Software Engineering*, 14(10) (1988), 1381-1393.
5. A. Hinneburg and D.A. Keim, An efficient approach to clustering in large multimedia databases with noise, *Proc. 4th Int. Conf. Knowledge Discovery in Databases*, 1998, 58-65.
6. H.V. Jagadish, Linear clustering of objects with multiple attributes, *Proc. ACM SIGMOD Conf.*, 1990, 332-342.
7. J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, Density-based clustering in spatial databases: The algorithm GDBSCAN and its application, *Data Mining and Knowledge Discovery*, 2(2) (1998), 169-194.
8. J.S. Vitter, Online data structures in external memory, *Proc. Annual International Colloquium on Automata, Languages, and Programming, LNCS*, Vol. 1644, 1999, 119-133.
9. X. Xu, M. Ester, H.-P Kriegel, and J. Sander, A distribution-based clustering algorithm for mining large spatial datasets, *Proc. 14th Int. Conf. on Data Engineering ICDE*, 1998, 324-331.
10. B. Zhou, D.W. Cheung, and B. Kao, A fast algorithm for density-based clustering in large database, *Proc. 3rd Pacific-Asia Conf. on Methodologies for Knowledge Discovery and Data Mining*, 1999, 338-349.

Transformation-Based GMM with Improved Cluster Algorithm for Speaker Identification

Limin Xu, Zhenmin Tang, Keke He, and Bo Qian

School of Computer Science, Nanjing University of Science and Technology
meiwen_xu@yahoo.com.cn

Abstract. The embedded linear transformation is a popular technique which integrates both transformation and diagonal-covariance Gaussian mixture into a unified framework to improve the performance of speaker recognition. However, the mixture number of GMM must be given in model training. The cluster expectation-maximization (EM) algorithm is a well-known technique in which the mixture number is regarded as an estimated parameter. This paper presents a new model that integrates an improved cluster algorithm into the estimating process of GMM with the embedded transformation. In the approach, the transformation matrix, the mixture number and other traditional model parameters are simultaneously estimated according to a maximum likelihood criterion. The proposed method is demonstrated on a database of three data sessions for text independent speaker identification. The experiments show that this method outperforms the traditional GMM with cluster EM algorithm.

Keywords: Gaussian mixture model (GMM); Improved cluster algorithm; Linear transformation; Expectation-maximization (EM) algorithm.

1 Introduction

Gaussian mixture speaker model (GMM) statistically represents the underlying sounds or vocal tract configurations that characterize a person's voice, and it has been proven very effective for speaker recognition [1,2]. Usually, Gaussian mixture density functions use diagonal covariance matrices. The advantage is that the model is simple and easy for computation [3,4,5]. However this also reduces the likelihood of the data. In order to compensate the losing likelihood, many approaches have presented in recent years. Ljolje has demonstrated that explicitly modeling the correlation between feature elements can improve the performance of recognition [6]. The drawback of this method is that the orthonormal transformation is outside of the statistical framework and is not optimized together with GMM parameters. Kuo-Hwei [3] integrated the orthonormal transformation into the statistical structure. In this approach, the transformation matrix is regarded as a set of statistical parameters. Chih-chien [7] proposed a classification scheme that incorporates Karhunen-Loeve transform (KLT) [8] and GMM for text-independent speaker identification. Transformation based method is also applied in speaker adaptation algorithm [9].

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1006–1014, 2007.

However, the mixture number of GMM must be given beforehand in all these approaches. Charles A. B [10] advanced a cluster approach of parameter estimation for GMM with EM algorithm, in which the mixture number was regarded as an estimated parameter as the same as the other parameters such as the mean vectors and the covariance matrices. Thus the parameters estimated by the cluster approach are more accurately to depict the distribution of feature than the ordinary EM algorithm.

This paper presents a new approach to estimate the GMM parameters. We integrate an improved cluster algorithm into the estimating process of GMM with the embedded transformation. Fig. 1 illustrates the conceptual block diagram. The approach is referred as transformed cluster GMM (TC-GMM), and we refer to GMM with embedded transformation as TE-GMM and GMM with cluster EM algorithm as CE-GMM. In the new approach, the mixture number, together with other parameters of GMM i.e. the weights, the mean vectors and the diagonal covariance matrices would be estimated by the improved cluster EM algorithm. In the experiment, we investigate the performance of the TC-GMM and the other two methods.

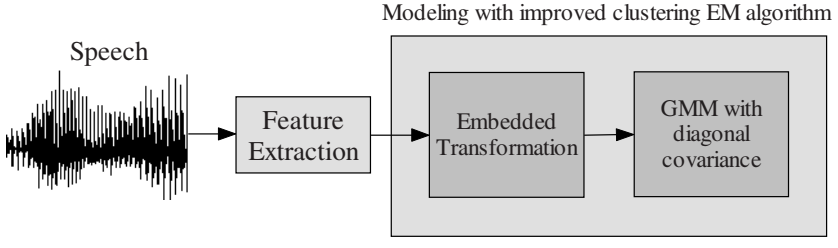


Fig. 1. Improved cluster EM algorithm and embedded transformation used in GMM estimation

2 Cluster and Transformation Technique for GMM

2.1 Transformation Embedded GMM with Diagonal Covariance Matrices

In the research of linear transformation with diagonal covariance matrices, two methods are usually used. In the early approach, the transformation matrix and diagonal-covariance Gaussian mixture parameters are modeled separately. In this paper, we apply another approach, that is, the transformation matrix and the diagonal-covariance Gaussian mixture are combined into a uniform statistical model [3].

In order to estimate the parameters of Gaussian mixtures, it is necessary to determine the number of mixtures. In the section, let us assume that this model has K mixtures and the number K is fixed. Specifically, let y be an M dimensional random vector to be modeled using a Gaussian mixture distribution. Then the parameters are required to completely specify the k^{th} mixture. π_k is the mixture weight. μ_k is the M dimensional mean vector for mixture k. R_k is the covariance matrix for mixture k. Then we use the notation π, μ and R to denote the parameter sets $\{\pi_k\}_{k=1}^K$, $\{\mu_k\}_{k=1}^K$, and $\{R_k\}_{k=1}^K$. So the complete set of parameters are then given by K and $\theta = (\pi, \mu, R)$.

Now let $Y = \{y_1, y_2, \cdots, y_N\}$ be N speaker feature vectors for speaker model training. Then the Gaussian mixture density function is given by

$$p(y_n|\theta) = \sum_{k=1}^{K} \frac{\pi_k}{(2\pi)^{M/2}|R_k|^{1/2}} \exp\left\{-\frac{1}{2}(y_n - \mu_k)^t R_k^{-1}(y_n - \mu_k)\right\}. \quad (1)$$

Where each covariance matrix R_k can be explicitly decomposed into eigenvalue matrix Λ_k and eigenvector matrix Ω, that is $R_k = \Omega\Lambda_k\Omega^T$ .The Ω is tied across all Gaussian components while each Λ_k is a component-specific matrix. The parameters of the statistical model are denoted as $\theta = \{\Omega, \pi_k, \mu_k, \Lambda_k | k = 1, \cdots, K\}$.

The ML estimation of the set θ is obtained by maximizing the likelihood function,

$$\theta_{ML} = \arg\max_{\theta} \prod_{n=1}^{N} p(y_n|\theta). \quad (2)$$

Because θ couldn't be solved directly, the EM algorithm is used. Starting from an initial model θ, the new model $\hat{\theta}$ is estimated by maximizing the auxiliary function.

$$Q(\theta, \hat{\theta}) = \sum_{n=1}^{N}\sum_{k=1}^{K} p(k|y_n, \theta) \cdot \log p(y_n, k|\hat{\theta}). \quad (3)$$

Where $p(k|y_n, \theta)$ is the posteriori probability and $p(y_n, k|\hat{\theta})$ is the priori probability.

In addition, two constraints can help to obtain the re-estimated formulas, that is $\sum_{k=1}^{K} \hat{\pi}_k = 1$ and $\hat{\Omega}\hat{\Omega}^t = I_M$ .Where I_M denotes a $M \times M$ identity matrix.

The re-estimated formulas of weights $\hat{\pi}_k$ and mean vectors $\hat{\mu}_k$ are easily derived, i.e.

$$\hat{\pi}_k = \frac{1}{N}\sum_{n=1}^{N} p(k|y_n, \theta), \quad k = 1, \ldots, K. \quad (4)$$

$$\hat{\mu}_k = \frac{\sum_{n=1}^{N} p(k|y_n, \theta) y_n}{\sum_{n=1}^{N} p(k|y_n, \theta)}, \quad k = 1, \ldots, K. \quad (5)$$

Since the derivation of the formulas for $\hat{\Omega}$ and $\hat{\Lambda}_k$'s is very complicated, we define $\hat{\Omega}$ by its column vectors as $\hat{\Omega} = [\hat{\xi}_1, \cdots, \hat{\xi}_M]$ and the diagonal matrices $\hat{\Lambda}_k$ by their diagonal elements, i.e.,

$$\hat{\Lambda}_k = diag[\hat{\rho}_{k,1}, \cdots, \hat{\rho}_{k,M}], \quad k = 1, \ldots, K. \quad (6)$$

Then the column vector $\hat{\varsigma}_m$ and diagonal elements $\hat{\rho}_{k,m}$ must be simultaneously solved from the following nonlinear equations:

$$\hat{\rho}_{k,m} = \hat{\xi}_m{}^t \hat{S}_k \hat{\xi}_m , \ k = 1, \dots , K, \ m = 1, \dots , M. \tag{7}$$

$$\hat{\xi}_m{}^t \left(\sum_{k=1}^{K} \hat{\pi}_k \frac{\hat{\rho}_{k,m} - \hat{\rho}_{k,j}}{\hat{\rho}_{k,m}\hat{\rho}_{k,j}} \hat{S}_k \right) \hat{\xi}_j = 0 \quad m, j = 1, \dots , M, \ m \neq j. \tag{8}$$

Where:

$$\hat{S}_k = \frac{\sum_{n=1}^{N} p(k|y_n, \theta)(y_n - \hat{\mu}_k)(y_n - \hat{\mu}_k)^t}{\sum_{n=1}^{N} p(k|y_n, \theta)} . \tag{9}$$

The nonlinear equations (7) and (8) can be solved by the FG algorithm. The details are shown in literature [3]. Thus the embedded GMM parameters $\theta = \{\Omega, \pi_k, \mu_k, \Lambda_k | k = 1, \cdots, K\}$ could be computed iteratively, while the diagonal covariance matrix is applied in the computation of Gaussian mixture density function.

2.2 Improved Cluster EM Algorithm with Diagonal Covariance Matrices

In section 2.1, the number of mixture K is fixed. Therefore, before modeling the speech feature distribution using transformation embedded GMM, the number K must be determined without any priori information. The number K would not be fitted with the actual feature distribution in most cases. In order to search the better parameters, the cluster EM algorithm [10] is applied. In the algorithm, the number of mixture K is also regard as a GMM parameter. The complete set of parameters are given by the number K and θ. Also the number K must be an integer greater than 0, $\hat{\Omega}\hat{\Omega}^t = I_M$ and $\sum_{k=1}^{K} \pi_k = 1$. The set of admissible θ for K mixtures model is denoted by $\Gamma^{(K)}$. The log of the probability of the entire training sequence is then given by

$$\log p(y|K, \theta) = \sum_{n=1}^{N} \log(\sum_{k=1}^{K} \pi_k p(y_n|k, \theta)) . \tag{10}$$

In the same way, the maximum likelihood (ML) estimate is used to estimate the parameter K and $\theta = \{\Omega, \pi_k, \mu_k, \Lambda_k | k = 1, \cdots, K\}$. It is given by

$$\hat{\theta}_{ML} = \arg\max_{\theta \in \Gamma^{(K)}} \log p(y|K, \theta) . \tag{11}$$

However, the ML estimate of K is not well defined because the likelihood may always be made better by choosing a large number of mixtures. The addition of a penalty term in the log likelihood of account for the over-fitting of high order models is generally adopted to estimate the model order. In this paper, the minimum description length (MDL) estimator [11] is used and the expression of minimization is

$$MDL(K,\theta) = -\sum_{n=1}^{N} \log(\sum_{k=1}^{K} \pi_k p(y_n|k,\theta)) + \frac{1}{2} L \log(NM) . \tag{12}$$

Where N is the length of the feature vectors, M is the dimension of a vector and L is

$$L = K(1 + M + \frac{(M+1)M}{2}) - 1 . \tag{13}$$

The new estimates of weights $\hat{\pi}_k$ and mean vectors $\hat{\mu}_k$ are computed by equations (4) and (5) with EM algorithm. Using functions (7) and (8) can estimate the $\hat{\Omega}$ and $\hat{\Lambda}_k$.

The four re-estimated formulas show how to update the parameter θ, they do not show how to change the model order K. That is, how to decrement the number of mixtures from K to K-1 is the remaining question. In the paper, we reduce the number of K by merging two mixtures to form a single mixture. With the idea, two pivotal problems must to be solved. One problem is determining which two mixtures should to be merged and the rule of determination. After the two mixtures are fixed, how to compute the values of the parameters of the new mixture is another problem. The two problems would be discussed in blending each other. Suppose two mixtures, l and m, may be effectively merge in a single mixture, then

$$\pi_{(l,m)} = \hat{\pi}_l + \hat{\pi}_m . \tag{14}$$

$$\mu_{(l,m)} = \frac{\hat{\pi}_l \hat{\mu}_l + \hat{\pi}_m \hat{\mu}_m}{\hat{\pi}_l + \hat{\pi}_m} \tag{15}$$

$$\Lambda_{(l,m)} = \frac{\hat{\pi}_l (\hat{\Lambda}_l + \hat{\Omega}^t (\hat{\mu}_l - \mu_{(l,m)})(\hat{\mu}_l - \mu_{(l,m)})^t \hat{\Omega}) + \hat{\pi}_m (\hat{\Lambda}_m + \hat{\Omega}^t (\hat{\mu}_m - \mu_{(l,m)})(\hat{\mu}_m - \mu_{(l,m)})^t \hat{\Omega})}{\hat{\pi}_l + \hat{\pi}_m} \tag{16}$$

Where $\pi_{(l,m)}$, $\mu_{(l,m)}$, $\Lambda_{(l,m)}$ denote the weight, mean vector and diagonal covariance matrix of the new mixture. Using (15) and (16), a distance function is defined as

$$d(l,m) = \frac{N\hat{\pi}_l}{2} \log\left(\frac{\left|\Lambda_{(l,m)}\right|}{\left|\hat{\Lambda}_l\right|} \right) + \frac{N\hat{\pi}_m}{2} \log\left(\frac{\left|\Lambda_{(l,m)}\right|}{\left|\hat{\Lambda}_m\right|} \right). \tag{17}$$

With the function (13), it is now possible to search over the set of all pairs, (l, m), to find the pair which minimizes $d(l, m)$, i.e.

$$(l^*, m^*) = \arg\min_{(l,m)} d(l,m) . \tag{18}$$

These two mixtures are then merged. The resulting parameter set $\theta^*_{(l,m)}$ is used as a initial condition for EM optimization with K-1 mixtures. Unfortunately, if the order of model is K_0 , cluster EM algorithm must do K_0 ordinary EM processes, which take a

long time. In this paper, the two step estimation algorithm is advanced. In the approach, we change the step length to n (n>1) so that it will improve the training efficiency for about several times. After determining the best value $K^*_{(n)}$, we then accurately re-search the parameters which minimize the value of MDL with the order of model ranging from $K^*_{(n)} - n + 1$ to $K^*_{(n)} + n - 1$ in which the step length is 1.

The final cluster EM algorithm is given in the following steps.

1. Initialize the order of model with a large number K_0.

2. Initialize $\theta = \{\Omega, \pi_k, \mu_k, \Lambda_k | k = 1, \cdots, K_0\}$ using k-means cluster algorithm.

3. Apply the EM algorithm to compute the parameters of the new estimate $\widehat{\theta}$. Specifically, using (4) and (5) to compute weight $\bar{\pi}_k$ and mean vectors $\bar{\mu}_k$ and Using (7) and (8) to re-estimate $\widehat{\Omega}$ and $\widehat{\Lambda}_k$.

4. Set $\theta = \widehat{\theta}$ and repeat Step 3 until convergence.

5. Record the final parameter $\theta^{(K)}$, and compute the value $MDL(K, \theta^{(K)})$.

6. If the number of mixtures is greater than n, apply equation (18) to reduce the number of mixtures with n step length, set $K \leftarrow K - n$, and go back to step 3.

7. Choose the value $K^*_{(n)}$ and parameters $\theta^{(K^*_{(n)})}$ which minimize the value of MDL.

8. Re-initialize the order of model with $K^*_{(n)} - n + 1$, then repeat step 2 to 3 until the last number of order is $K^*_{(n)} + n - 1$ in which the step length is 1.

9. Finally find the value K^* and parameters $\theta^{(K^*)}$ which minimize the value of $MDL(K, \theta^{(K)})$.

3 Experiments

3.1 Database and Feature

The database used in the speaker identification experiments was collected under the environment of the ordinary laboratory with 8 kHz sampling rate and finally 8-bit A-law coding quantization. It consist 30 speakers (16 males and 14 females). The whole database is recorded in Chinese Mandarin and every speaker has his (her) own dialect more or less. Each utterance contains about 3 minutes speech. There are 3 sessions for each target speaker with 3 different contents. Each speaker pronounces 10 sequences of 4 connected digits about 30 seconds in session 1 which we called digital session. In session 2, the speaker pronounces *The wind and the sun* of Aesop's fable in Chinese edition. This session is called fixed content session and about 1 minute. The 3rd session is named free speech session in which the speaker is asked to describe his environment or tell what he has done during the day or something else. Anyway, speakers were kindly suggested not to say the same thing from speaker to speaker and suggested to speak randomly and colloquially. This session is limited in 1 minute.

In the experiments, the speech data of session 2 is chose to train the models involved in the paper. Then we use session 1 and 3 to test the performance of these models. The speech data were processed into frames of 256 samples, with a frame advance of 128 samples. Each frame was represented by a 24 component feature vector consisting of 12 MFCCs plus their first order derivatives.

3.2 Results

First, we examine the process of the two step estimation algorithm described in session 2.2. Suppose the first step length n is 5. The training data of two random speakers is examined. Fig. 2 shows the first estimation process of the two speakers in which the step length is 5. The initial number of mixtures is 120. The minimum of MDL values of the two speakers occurs at K=25 and K=30 in the rough. Fig. 3 shows the second estimation process with the step length = 1 which is based on the first estimation shown in Fig. 2. The final minimum of MDL value occurs at K=23 and K=31.

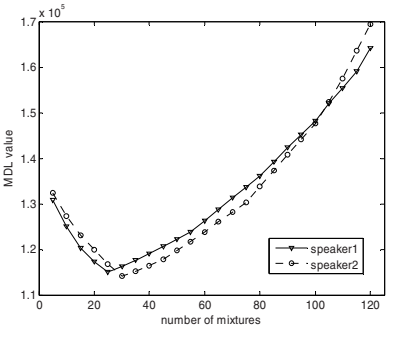


Fig. 2. First estimation process with step length = 5. Notice that the minimum of MDL values respectively occur at K=25 and K=30 with the two speakers in the rough

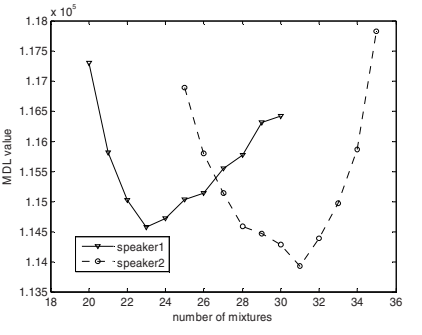


Fig. 3. Second estimation with step length = 1 on the basis of the process shown in Fig 2. Notice that the minimum of MDL value precisely occurs at K=23 and K=31

The second set of experiments compares the performance of the proposed TC-GMM with TE-GMM and CE-GMM. TC-GMM and CE-GMM have the same underlying structure and they both can achieve a best set of parameters including the mixture number K^*, but model the covariance matrices in different way. The performance is compared in terms of the average error rates and computational complexity. The results are shown in Table 1. Let the initial number of mixtures be K_0, Then the computational numbers in various mixture numbers of TC-GMM and CE-GMM are $\lfloor K_0 / n \rfloor + 1 + (2n - 2)$ and K_0, respectively. Where $\lfloor \cdot \rfloor$ is the greatest smaller integer function. The listed computational time is for processing one input utterance. The results show that TC-GMM is about 50% time saved than CE-GMM. TC-GMM has a better performance than CE-GMM and the error rates with TC-GMM decrease 2.1% compared with CE-GMM on average. The improvement of performance is because of applying embedded transformation matrices.

Table 1. The identification error rates for various testing data sets and average computational time of CE-GMM and TC-GMM

	Time (s)	Testing data sets (Error recognition rates (%))		
		Digital session	Free speech session	Average
CE-GMM	36.5	7.2	12.8	10.0
TC-GMM	18.4	6.1	9.8	7.9

TC-GMM and TE-GMM have the same way in modeling the covariance matrices, but the different structures are applied. Table 2 shows the performance of in terms of the error rates for various testing data sets. These error rates decrease as the mixture number increases when the number of mixture is below 32, while the error rates have a little change up and down when the number is greater than 32. The performance becomes saturation when the mixture number is around 32. The error rates directly reach the point of saturation by using the proposed model in two testing data sets. On the point, the error rates and computational complexity are balanced. Also, the error rates are smaller in the digital session than in the free speech session since the context in this session is random and completely independent with the training data set.

Table 2. The identification error rates of TE-GMM and TC-GMM for various testing data sets

Error recognition rates (%)			Mixture number K					
			4	8	16	32	64	128
Testing data sets	Digital session	TE-GMM	12.5	8.9	6.5	6.0	5.9	5.9
		TC-GMM			6.1			
	Free speech session	TE-GMM	16.2	12.0	10.5	9.9	9.8	9.7
		TC-GMM			9.8			

4 Conclusion

In this paper, TC-GMM is developed to integrate the improved cluster algorithm into the estimating process of Gaussian mixture models with the embedded transformation. The transformation matrix, the mixture number and other traditional model parameters are simultaneously estimated according to a maximum likelihood criterion. The experiments conducted on a speaker identification task show that this new method outperforms the traditional GMM with cluster EM algorithm. Moreover, compared with the transformation embedded GMM, the experiments show that TC-GMM can directly achieve the best point of saturation with the right mixture number in which the error rates and computational complexity are balanced.

References

1. S. Furui, in: C. Lee, F. Soong, K. Paliwal (Eds.), An Overview of Speaker Recognition Technology, Automatic Speech and Speaker Recognition, Kluwer Academic Press, 1996.
2. D.A. Reynolds, R.C. Rose, Robust Text-independent Speaker Identification Using Gaussian Mixture Speaker models, IEEE Trans. Speech Audio Process. 3 (1) 72-83 1995.

3. K.H. You, H.C. Wang, Joint Estimation of Feature Transformation Parameters and Gaussian mixture Model for Speaker identification, Speech Communication. 28 227-241 1999.
4. Q.Y. Hong, S. Kwong, A Discriminative Training Approach for Text-independent Speaker Recognition, Signal Processing. 85 1449-1463 2005.
5. Li, H., Haton, J.P., Gong, Y., On MMI Learning of Gaussian mixture for speaker models. Proceddings EUROSPEECH'95. 363-366 1995
6. Ljolje, A., The importance of cepstral parameter correlations in speech recognition. Computer Speech and Language. 8, 223-232. 1994.
7. C.-C.T. Chen, C.T. Chen, C.K. Hou, Speaker Identification Using Hybrid Karhunen-Loeve transform and Gaussian mixture model approach, Pattern Recognition. 37 1073-1075 2004.
8. Fukunaga, K., Introduction to Statistical Pattern Recognition, Academic Press. 1990.
9. C. Boulis, V. Diakoloukas, V. Digalakis, Maximum Likelihood Stochastic Transformation Adaptation for Medium and Small Data Sets, Computer Speech and Language. 15 257-285 2001.
10. C. A. Bouman, Cluster: An Unsupervised Algorithm for Modeling Gaussian Mixtures, http://www.ece.purdue.edu/~bouman, 2005.7.
11. J. Rissanen, A Universal Prior for Integers and Estimation by Minimum Description Length, Annals of Statistics. vol. 11, no. 2, pp. 417-431, 1983.

Using Social Annotations to Smooth the Language Model for IR

Shengliang Xu[1], Shenghua Bao[1], Yong Yu[1], and Yunbo Cao[2]

[1] APEX Data & Knowledge Management
Lab, Shanghai Jiao Tong University
{slxu,shhbao,yyu}@apex.sjtu.edu.cn
[2] Microsoft Research Asia
No.49 Zhichun Road, Beijing, P.R. China
{yunbo.cao}@microsoft.com

Abstract. In the paper, we present an exploration of using social annotations provided by the Web 2.0 sites (such as Del.icio.us) in helping web search. More specifically, we consider using the social annotations as an additional resource to strengthen existing smoothing methods for the language model for IR. The social annotations can benefit the smoothing of language model in two aspects: 1) the annotations themselves can serve as the summaries of the web pages given by the users; 2) the annotations can be seen as the links of the web pages sharing the same annotations. We propose three smoothing methods, addressing the two aspects and their combination, respectively. We call the new language model of using the proposed smoothing methods 'Language Annotation Model (LAM). Preliminary experimental results show that LAM significantly outperforms the traditional language models.

1 Introduction

Language modeling approaches to Information Retrieval (In the rest of this paper we take "Traditional Language Model" or the abbreviation "TLM" to represent the original language modeling approaches to Information Retrieval) have been approved to be very efficient [1, 2, 3]. The basic idea of this model is to estimate a language model for each document and then to rank documents by the likelihood of the query according to the estimated language model. One of the central problems of TLM is data sparseness. Many smoothing methods are studied trying to resolve this problem [2]. These classic smoothing methods are very efficient but one restriction: They all use only collection smoothing to compensate for data sparseness.

With the boosting of web 2.0 technologies, more and more web resources are annotated by the web users manually. Annotations are metadata of their owner webpage. With this extra data, is there some way to combine them into IR and improve IR efficiency? To answer this question, we first analyzed the characteristics of social annotations and then propose a novel model in three forms by integrating annotations in three manners: 1) the annotations are summaries of webpages in users' perspective; 2) an annotation is somewhat a cluster of the webpages sharing it; 3) their combination.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1015–1021, 2007.

We call the three Forms of LAM Language Annotation Model with Annotation Smoothing (LAM-AS), Language Annotation Model with Cluster Smoothing (LAM-CS), Language Annotation Model with both Annotation and Cluster Smoothing (LAM-ACS) respectively.

To evaluate our new models, we constructed a new test bed consisting of 1,736,268 web pages with 269,566 different tags from Del.icio.us[1]. 80 queries are collected and labeled by a group of CS students. The experimental results show that the three new models outperform both Vector Space Model (VSM) and TLM significantly.

In the rest of the paper, we first survey related work in Section 2, and then present three forms of the LAM in Section 3, later give experimental results in Section 4, finally make concluding remarks and give some future work in Section. 5.

2 Related Work

2.1 Language Model

For many years, the primary consumers of statistical language models were speech recognition systems [4]. In 1998, Ponte and Croft [1] proposed a smoothed version of the document unigram language model. Since then, there emerged a great amount of research work related to language model. Most of them tried to solve the following two problems: term dependency and data sparseness. Plenty of work has been done to model the proper dependencies between the query terms [5, 6]. This paper mainly focused on how to utilize annotation information to lighten the data sparseness problem and simply ignore the term dependency problem by assuming all terms were generated independently. In order to resolve the data sparseness problem, many smoothing methods were suggested to reevaluate the probabilities of generating the query terms that did not appear in the document. Song and Croft proposed the good-turing smoothing based on terms' power law distribution [3]. Zhai et al proposed the two-stage smoothing for language model [2]. In addition, cluster based smoothing methods were proposed and achieved significant improvement [7, 8].

2.2 Social Annotation Analysis

During the recent several years, many websites have been constructed to provide social annotation services (or collaborative tagging, folksonomy, social bookmarking).

Though much research work has been done on social annotation, little of them focus on integrating annotations to IR process. In [9], the authors give a very detail analysis of the social annotation data from Del.icio.us. In [10], the authors find the relationships among tags based on their co-occurrences with users or resources. However the above work didn't mention the integration of annotations and IR. [11] uses a probabilistic generative model to obtain the emergent semantics hidden behind the co-occurrences of web resources, tags and users and implements semantic search based on the emergent

[1] http://del.icio.us

semantics. In [12], the authors implement an annotation tool within enterprise environment and developed a method to improve search efficiency by utilizing annotations. Different from their work, this paper integrates annotations with TLM and proposes a new Language Annotation Model.

3 Language Annotation Model

The social annotations can benefit the smoothing of language model in two aspects: 1) the annotations themselves can serve as the summaries of the web pages given by the users; 2) the annotations can be seen as the links of the web pages sharing the same annotations. We propose three smoothing methods, addressing the two aspects and their combination, respectively.

In the first method, we consider using the combination of all the tags associated to a web page as a summary of the web page. More specifically, we concatenate all the tags to form a pseudo document which is used to smooth the language model from the original document. In the second method, we consider using the cluster of the documents linked to a given document by the annotations to smooth the language model from the given document. In the third method, we take into considerations the two aspects at the same time. The three methods lead to three new language models: Language Annotation Model with Annotation Smoothing (LAM-AS), Language Annotation Model with Cluster Smoothing (LAM-CS) and Language Annotation Model with both Annotation and Cluster Smoothing (LAM-ACS) respectively.

3.1 Language Annotation Model with Annotation Smoothing (LAM-AS)

In LAM-AS, we concatenate a web page's all annotations to create a pseudo document. The web page and the pseudo document are two data sources of one document.

$$P(q \mid d) = \prod_{i=1}^{m} \left\{ \begin{array}{l} (1-\gamma)\left[(1-\alpha)P(w_i|d) + \alpha P(w_i|C)\right] \\ +\gamma\left[(1-\lambda)P(w_i|d_a) + \lambda P(w_i|C_a)\right] \end{array} \right\} \quad (1)$$

Where $P(w_i|d)$, $P(w_i|C)$, $P(w_i|d_a)$ and $P(w_i|C_a)$ means the probability of generating the i^{th} query term from web page content, the whole web page collection, the pseudo document and the whole pseudo document collection respectively.

3.2 Language Annotation Model with Cluster Smoothing (LAM-CS)

In LAM-CS, we extend the idea of the cluster-based smoothing [7]. We consider two kinds of strategies in using social annotations for building clustering.

– The documents within a cluster are linked by a social annotation tag. The tag is actually extra information other than the documents themselves. Since a document can potentially have unlimited amount of annotations, one document can be smoothed by a number of other documents linking to it via the tags.

– The social annotation tags can be semantically related since a single semantic sense can be articulated in different words by different users. Thus, one document can also be smoothed by the documents whose social annotation tags are semantically close to the tag of the document.

For the first strategy, we calculate the arithmetic average of the smoothing scores of all the clusters containing the document. For the second strategy, we proposed to use "tag similarity" $sim(tag_a,tag_b)$ to quantify the semantic similarity between two tags tag_a and tag_b. Finally we get the cluster smoothing model:

$$P(q \mid d) = \prod_{i=1}^{m} \left\{ \begin{array}{l} (1-\alpha) P(w_i \mid d) \\ + \alpha \left[\begin{array}{l} \dfrac{(1-\beta)}{Count(tag)} \sum\limits_{j=1}^{Count(tag)} \left[sim(tag_j, w_i) \times P(w_i \mid TagClu_j) \right] \\ + \beta P(w_i \mid C) \end{array} \right] \end{array} \right\} \tag{2}$$

Where $P(w_i|TagClu_j)$ is the probability of generating the i^{th} query term from the j^{th} tag cluster. $Count(tag)$ is the amount of tags given to the document. $sim(tag_j,w_i)$ is the similarity from the j^{th} tag to the i^{th} query term.

3.3 Language Annotation Model with Both Annotation and Cluster Smoothing (LAM-ACS)

The LAM-ACS is the integrating of LAM-AS and LAM-CS.

$$P(q \mid d) = \prod_{i=1}^{m} \left\{ \begin{array}{l} (1-\gamma) \left\{ \begin{array}{l} (1-\alpha) P(w_i \mid d) \\ + \alpha \left[\begin{array}{l} \dfrac{(1-\beta)}{Count(tag)} \sum\limits_{j=1}^{Count(tag)} \left[sim(tag_j, w_i) \times P(w_i \mid TagClu_j) \right] \\ + \beta P(w_i \mid C) \end{array} \right] \end{array} \right\} \\ + \gamma \left[(1-\lambda) P(w \mid d_a) + \lambda P(w \mid C_a) \right] \end{array} \right\} \tag{3}$$

4 System Evaluation and Analysis

4.1 Delicious Data

For the experiment, we crawled webpages and social annotations from Del.icio.us. The dataset consists of 1,736,268 webpages with 269,566 different tags. We conducted two steps of preprocessing on the raw dataset as listed in the following.

Firstly, though the tags of Del.icio.us are easy for human to read and understand, they are not designed for machine. For the limitation of the Del.icio.us service, a tag cannot have a space. Users may concatenate several words together to form a tag like 'javaprogramming' or 'java/programming'. We split/tokenize this kind of tags with the help of WordNet. In [9], social annotation tags are grouped into 7 categories. We found Category 4, 6 and 7 too user-specific (e.g. tobuy, toread, myprefer etc.). Tags falling into these categories are of little value for generic IR so we filter them out.

Secondly, we define Del.icio.us tag similarity according to two intuitions: 1) the tag A and tag B may semantically related to each other if their frequencies in a certain document are very close. The closer the more similar. 2) The similarity may be asymmetrical. For example, *sim(Ubuntu,Linux)* may be 0.6 while *sim(Linux,Ubuntu)* may be only 0.06. Because Linux is a superset of Ubuntu. Equation (4) illustrates the intuitions,

$$\mathrm{sim}(\mathrm{tag_a}, \mathrm{tag_b}) = \sum_{j=0}^{D_a} \frac{1}{1+9\times\left|C_{aj}-C_{bj}\right| \Big/ 100+\left|C_{aj}-C_{bj}\right|} \Bigg/ \sum_{i=0}^{D_a} C_{ai} / C_{di} \tag{4}$$

where C_{aj} and C_{bj} means how many times tag_a and tag_b are used on the j^{th} document, respectively. C_{di} is the total tag amount of the i^{th} document. Here is some sample: engine-google: 0.0710; engine-game: 0.0569; google-engine: 0.1018;

4.2 Experiment Setup

In order to evaluate the LAM's performance, we asked for a group of CS students to help us collect 80 queries, such as image search engine, Beijing Olympic 2008, hydrogen fuel cell, Beatles music. The queries contain 497 relevant documents in all.

All the models are based on Lucene 2.0. We implemented BM25 [13] as the VSM baseline of our experiment: k1, k3 and b are set 1, 1 and 0.5 respectively (It's Xapian's default weighting scheme[2]). In the following experiment, MAP and Recall are used to evaluate the retrieval performance.

4.3 Evaluation of Language Annotation Model

In order to make the evaluation fair, in BM25 we merge each web page's annotations into its content to keep the five models' data source the same amount. The parameter γ in Equation (1) and (3) is selected to roughly make the score following γ and the score following (1-γ) equal.

As we can see from Table 2 and Fig 1, the three forms of LAM all outperform BM25 and TLM. To understand whether the improvement is significant, we also performed t-tests on MAP. The p-value between LAM-ACS and TLM is 0.032, indicating significant improvement. However, compare to TLM, the Recall of the three LAMs are

Table 2. Evaluation of BM25, TLM, LAM-AS, LAM-CS, LAM-ACS

Model	Map/Recall Value	Compare to BM25
BM25(baseline)	0.4157 / 430	
TLM	0.4654 / 474	+12.0% / +10.2%
LAM-AS	0.5092 / 458	+22.5% / +6.5%
LAM-CS	0.4751 / 451	+14.3% / +4.9%
LAM-ACS	0.5188 / 452	+24.8% / +5.1%

[2] http://www.xapian.org/docs/bm25.html

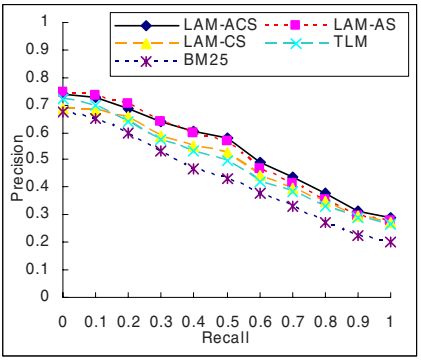


Fig. 1. 11-point precision/recall curves for TLM, LAM-AS, LAM-CS, LAM-ACS, BM25

much lower. We analyzed this phenomenon and find that the occurrence of a term in an annotation will bring more impact to the result $P(q|d)$ score than that in the web page content since most annotations are much shorter than web page contents. For those queries which have some stopword-like terms like '*software*', the smoothing scores may be too great and even contribute more than the web page.

5 Conclusion and Future Work

In this paper we study the problem of integrating social annotations into language model. The main contribution can be concluded as follows: 1) Propose to use social annotations to lighten the data sparseness within language model for IR; 2) Propose a novel Language Annotation Model to utilize social annotations. Three forms of LAM are studied. 3) The evaluation of the LAM using the Del.icio.us data, experimental results show that the LAM outperforms both the TLM and VSM significantly.

It's just a start to integrate social annotations into language model. In future, we will explore more sophisticated smoothing methods for language model and integrate social annotations into other retrieval models.

Acknowledgment

The authors would like to thank the three anonymous reviewers for their elaborate and helpful comments.

References

1. Ponte, J.M., Croft, W.B.: A Language Modeling Approach to Information Retrieval. In: *Research and Development in Information Retrieval.* (1998) 275-281
2. Zhai, C., Lafferty, J.: A Study of Smoothing Methods for Language Models Applied to Information Retrieval. In: *ACM Transactions on Information Systems* 22 (2004) 179-214

3. Song, F., Croft, W.B.: A general language model for information retrieval. In: *Proc. of CIKM'99* 316 – 321
4. Rosenfeld, R.: Two decades of statistical language modeling: Where do we go from here. In: *Proc. of the IEEE*. (2000) vol. 88, no. 8
5. Srikanth, M., Srihari, R.K.: Exploiting syntactic structure of queries in a language modeling approach to IR. In: *Proc. of CIKM'03.* 476–483
6. Bao, S., Zhang, L., Chen, E., Long, M., Yu, Y.: LSM: Language Sense Model for Information Retrieval. In: *Proc. of WAIM'06*. 97-108
7. Kurland, O., Lee, L.: Corpus structure, language models, and ad hoc information. In: *Proc of SIGIR'04* 194–201
8. Xu, J., Croft, W.: Cluster-based retrieval using language models. In: *Proc. of SIGIR'*04 186-193
9. Golder, S.A., and Huberman, B.A.: The Structure of Collaborative Tagging Systems. *http://www.hpl.hp.com/research/idl/papers/tags/*, 2005
10. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In *Proc. Of ISWC'05,* 2005
11. Wu, X., Zhang, L., and Yu, Y.: Exploring social annotations for the semantic web. In *Proc. of WWW'06.* ACM Press, New York, NY, 417-426.
12. Dmitriev, P. A., Eiron, N., Fontoura, M., and Shekita, E.. Using Annotations in Enterprise Search. In *Proc. of WWW'06.* ACM Press, New York, NY, 811-817.
13. Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A., Lau, M.: Okapi at TREC. In: *TREC'*92 21-30

Affection Factor Optimization in Data Field Clustering

Hong Yang, Jianxin Liu, and Zhong Li

North China Electric Power University
No.204, Qingnian Road, Baoding, 071003, China
yh1969719@126.com

Abstract. Although Data Field Clustering method has a lot of advantages, clustering result depends severely on affection factor that is selected in Data Field function. The purpose of the paper is to find an optimum affection factor that may not only reflect nature characteristic of clustering data sample, but also reduce influence caused by sample deviation to minimum. In this paper, an affection interval concept is defined at first. Then an optimum objective function for reducing influence of sample deviation is constructed and an approximate solution is given of optimum affection factor. In the end, a standard data set offered in the MATLAB is used to test the availability of the optimum affection factor, the result is satisfactory.

Keywords: data field clustering, affection interval, optimum affection factor, sample deviation.

1 Introduction

Date Field concept was derived from Field concept in physics and it supposed that there was a virtual field around data in sample space. This concept was firstly applied in Computer Graphics. In 1995, Deyi Li[1] introduced Data Field concept to Knowledge Discovery in Space Database. In 2002, WANG Shu-Liang[2] discussed the characteristic of Data Field in details. In 2006, a hierarchical clustering method based on data field [3] was released.

Data Field Clustering is much different from conventional clustering methods[4][5]. Conventional clustering methods use directly distances among objects as measurement of similarity. Data Field Clustering transforms distances into potential value. If the potential value of a point in sample space is low, it indicates that the point is surrounded by few objects or it is far from them. If the potential value of a point is high, it indicates that the point is surrounded by many objects or it is near from them.

Data Field Clustering has a lot of advantages, for example, 1) model is clear and is convenient to analyze, 2) it can complete arbitrary shape clustering, 3) it can reflect hierarchical relations of clustering result etc. But clustering result depends severely on selected affection factor that determine Data Field influential extension. If affection factor is smaller, clustering numbers are more. If affection factor is larger, clustering numbers are less. How to find an optimum affection factor has became a key problem in using Data Field Clustering method. The optimum affection factor must satisfy two conditions: 1) it must reflect natural clustering result of sample, 2) it must keep

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1022–1028, 2007.

clustering result stability when sample has some deviations. The purpose of the paper is to research a method that can find an optimum affection factor satisfying two conditions above.

The rest of this paper is organized as follows. In section 2, data field and affection factor are defined. In section 3, affection interval is defined. In section 4, an optimum objective function is constructed and an approximate affection factor is given. In section 5, experiments are provided to test this solution.

2 Data Field and Affection Factor

2.1 Data Field

Not any function may be used as Data Field. Data Field function must satisfy some conditions.

Definition 1 (Data Field). *Suppose that an object x_0 has a virtual field around it in sample space $\Omega \in R^p$, its function is $g(x)$. If follow conditions are satisfied:*

(1) $g(x)$ is a continuous, smooth, limited function

(2) $g(x)$ is an Isotropic function

(3) $g(x)$ is an decreasing function of distance between any point $x \in \Omega$ and field point x_0. When the distance is equal to 0, $g(x)$ has a maximum value. When the distance is equal to $+\infty$, $g(x)$ is equal to 0.

Then, $g(x)$ is called as Data Field of the object(point) x_0. The parameter that determined Data Field influential extension is called affection factor.

Although Data Field may be described by a vector function or a scalar function, a scalar function is usually used in Data Field because of its simple.

According to definition of the Data Field, any function that satisfies the three conditions can be used in Data Field. As we know, Gauss Function not only satisfies the three conditions, but also it is simple. So Gauss Function becomes first selection of scalar function used in Data Field Clustering. If not assert, Gauss Function is always used in the paper.

The formula of Gauss Function used in the paper is as follow:

$$g(x) = e^{-\left(\frac{\|x-x_0\|}{\sigma}\right)^2} \tag{1}$$

Where, $\|x-x_0\|$ is a distance between any point x and data point x_0, $\sigma \in (0,+\infty)$ is a affection factor that determine Data Field influential extension.

Based on definition of Data Field of an object, a potential function of sample space is defined as follow:

Definition 2 (Potential Function of Sample Space). *Given n objects set $X = \{x_1, x_2, \ldots, x_i, \ldots x_n\} \in \Omega \in R^p$, if $\forall x_i \in X$ has a data field $g_i(x)$, then potential value of $\forall x \in \Omega$ is as follow formula:*

$$G(x) = \sum_{i=1}^{n} g_i(x) \tag{2}$$

2.2 Affection Factor

According to principle and process of Data Field Clustering, it is easily found that there is one-to-one correspondence of clustering number and maximum value number of formula(2). When sample is given, maximum value number of formula(2) depend on affection factor σ. If affection factor is small, formula (2) has many Maximum value, that is to say, sample clustering number is many. If affection factor is large, formula (2) has few Maximum value, that is to say, sample clustering number is few. So, if maximum value number can be obtained, clustering number is obtained too.
1- order derivative of formula (2) is:

$$\nabla G_x = \frac{\partial G}{\partial x} \tag{3}$$

Let formula (3) equal 0, can acquire maximum value. Let N is maximum value set

$$N = \left\{ x \middle| \nabla G_x = 0 \right\} \tag{4}$$

Let $\|N\|$ denote element number in set N, when affection factor is changing, $\|N\|$ is correspondingly changing with affection factor σ. So, $\|N\|$ is also signed by $\|N(\sigma)\|$.

3 Affection Interval

3.1 Affection Interval

Simply spoken, affection interval is real interval of affection factor in which clustering number retain invariant.

Definition 3 (Affection Interval). *Given* $K = (k_1, k_2) \subset R$,
1) if $\forall \sigma_1 \in K$, $\forall \sigma_2 \in K$, *then* $\|N(\sigma_1)\| = \|N(\sigma_2)\|$
2) if $\forall \sigma_1 \in K$, $\forall \sigma_2 \notin K$, *then* $\|N(\sigma_1)\| \neq \|N(\sigma_2)\|$
Then K *is called an affection interval.* k_1 *is called lower borderline .* k_2 *is called upper borderline.*

3.1 Satisfactory Affection Interval

In Hierarchical Clustering, there is an inconsistency coefficient[6][7] in clustering tree. Commonly, natural clustering result of sample can be found by means of a larger inconsistency coefficient. When affection factor is increasing from a smaller value, Data Field Clustering process is similar to Hierarchical Clustering. So, natural clustering result can also be found in a larger affection interval. The larger affection interval is called satisfactory affection interval.

So far, the first condition finding optimum affection factor is satisfied. The natural clustering result of sample is in satisfactory affection interval. The second condition is to select an affection factor in the satisfactory affection interval.

4 Affection Factor Optimization

After finding satisfactory affection interval, any affection factor in the interval can reflect natural clustering result of sample. But, as is well known, all of samples have deviations. Obviously, those affection factors closing with k_1 or k_2 are sensitive to deviations. How to selecting an optimum affection factor is the purpose of this section.

According to affection interval border value, the potential border value may be acquired:

$$G_{k_1}(x) = \sum_{i=1}^{n} e^{-\left(\frac{\|x-x_i\|}{k_1}\right)^2} \tag{5}$$

$$G_{k_2}(x) = \sum_{i=1}^{n} e^{-\left(\frac{\|x-x_i\|}{k_2}\right)^2} \tag{6}$$

Correspondingly, when potential value in interval $\left(G_{k_1}(x), G_{k_2}(x)\right)$, the clustering result retain invariant too.

Suppose sample will change, deviation $D = \{d_1, d_2, \ldots, d_i, \ldots d_n\}$, when $k \in (k_1, k_2)$, the potential is:

$$G_k(x) = \sum_{i=1}^{n} e^{-\left(\frac{\|x-x_i+d_i\|}{k}\right)^2} \tag{7}$$

If $G_k(x)$ satisfies :

$$G_{k_1}(x) < G_k(x) < G_{k_2}(x) \tag{8}$$

Then sample deviation does not deconstruct clustering result.

Obviously, formula(8) is not hold. Suppose $\forall x \in S_x \subset \Omega$ be not satisfactory with formula(8).

Objective function of k optimization is:

$$\min S_x(k) \tag{9}$$

s.t.

$$\begin{aligned} & G_k(x) > G_{k_2}(x) \\ & G_k(x) < G_{k_1}(x) \\ & x \in S_x \end{aligned} \tag{10}$$

The k that is satisfactory with objective function is the optimum affection factor. Commonly, the solution is not acquired.

Using one-dimension and one data sample, objective function may be solvated. Let $\frac{\partial S}{\partial k} = 0$, acquire optimum affection factor:

$$k = \sqrt{k_1 k_2} \tag{11}$$

As usually, the solution may be approximate answer of objective function(9).

5 Experiments

5.1 Experiment Date

Fcmdata is a data set used in MATLAB to test how Fuzzy Clustering Method clustering works. It is a two-dimensional data set. The number of data is 140. It is very appropriate to research Data Field Clustering method.

5.2 Satisfactory Affection Interval

At first, satisfactory affection interval needed to be found. An improved Blocking Search algorithm is used to find satisfactory affection interval. To show relation of affection factor and clustering result, different affection factor and corresponding clustering result is listed in table 1.

The first row and the third row is affection factor. The second row and the forth row is corresponding clustering number. It is clearly that clustering number two is most natural. And corresponding affection interval is:

K=[0.13 0.22]

Table 1. Affection factor and corresponding clustering

A.F.	0.24	0.23	0.22	0.21	0.20	0.19	0.18	0.17	0.16
C	1	1	2	2	2	2	2	2	2
A.F.	0.15	0.14	0.13	0.12	0.11	0.10	0.09	0.08	0.07
C	2	2	2	3	3	4	6	7	7

A.F. denotes Affection Factor. C denotes Clustering number.

This result is same as Fuzzy Clustering result. Clustering result figure using affection factor 0.15 is as follow:

5.2 Deviation Effect

To check the validity of the approximate solution, some steps should be as follows. First, clustering result of original sample should be obtained. In this experiment,

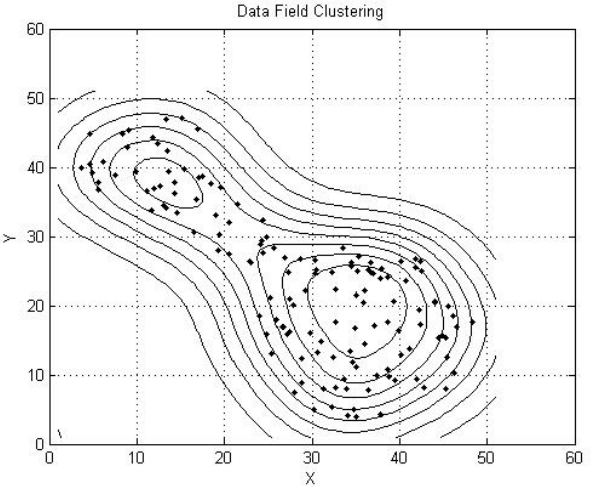


Fig. 1. The points in the figure are sample data. The curves are contour of potential value. It is obviously showed that there are two clustering results in affection factor 0.15.

affection interval is obtained above, their clustering result is 2. Second, some data are selected randomly to add random deviation. Third, clustering result is calculated again using same affection factor as first step. Forth, compare clustering result with the first, and check whether change the clustering number. All of steps should be executed many times, and different random deviation is added on sample data. The statistical result of the change of clustering number is more convictive.

Some notices of experiment have: 1) deviation value is not larger or smaller. In paper, every point is added different bound uniform distribution random value. The bound dose not exceed border of sample normalized, 2) deviation number is not less or more. Less number is not effect, and more number will cancel reciprocally. In this experiment, 70 objects are added deviation, 3) in the paper, 50 times repetition calculations are executed. Larger the invariant clustering result is, more robust the affection factor is. The result is as follow:

Table 2. Statistical number of invariant clustering result in 50 time calculation

A.F.	0.22	0.21	0.20	0.19	0.18	0.17	0.16	0.15	0.14	0.13
65t	16	22	21	21	29	30	31	32	31	24
70t	15	22	25	24	25	32	33	31	27	20
75t	18	22	23	28	27	28	29	29	29	24

A.F. denotes affection factor. 65t, 70t, 75t denotes number of data deviation.

The first row is affection factors keeping clustering number tow in the sample. The first list is date number which is disturbed by uniform distribution. The data across of rows and

lists are statistical number that clustering number is tow. It is observed that maximum rate is probably in 0.17 or 0.16. The value is according to formula (11) that is:

$$k = \sqrt{k_1 k_2} = \sqrt{0.22 * 0.13} = 0.1691$$

5 Conclusion

Data Field Clustering is much different from conventional clustering methods. Up to now, the research of Data Field Clustering is infrequent and insufficient. Because of a lot of advantages, the method is valuable to research. The paper research thoroughly mechanism of Data Field Clustering and influence of affection factor in clustering result. Considering sample deviation, an aim function of affection factor optimization is proposed and an approximate solution is given. Test result indicates that the solution is satisfactory.

As a next work of the research, the availability of objective function and approximation solution need to be checked up in actual application. In addition, time expenditure problem of Data Field Clustering need to be solved.

References

1. Li De-yi, Meng Hai. jun, Shi Xue. mei.: Membership clouds and membership cloud generators. Computer. Research and Development(1995)
2. WANG Shu-Liang.: Data Field and Cloud Model Based Spatial Data Mining and Knowledge Discovery. Ph.D. Thesis. Wuhan: Wuhan University, China (2002)
3. GAN Wen-yan, LI De-yi, WANG Jian-min.: An Hierarchical Clustering Method Based on Data Field. ACTA ELECTRONICA SINICA, Vol.34, No.2(2006)
4. Rui Xu, Donald Wunsch II.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, VOL. 16, NO. 3, (2005)
5. Jiawei Han, Mcheline Kamber.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, Inc. (2001)
6. L.Kaufman, P.J.Rousseeuw.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York (1990)
7. M. J. Zaki, R. Ramakrishnan, M. Livny.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data, Montreal, Canada(1996)

A New Algorithm for Minimum Attribute Reduction Based on Binary Particle Swarm Optimization with Vaccination

Dongyi Ye, Zhaojiong Chen, and Jiankun Liao

College of Mathematics and Computer, Fuzhou University, Fuzhou 350002, China
yiedy@fzu.edu.cn

Abstract. Computation of a minimum attribute reduct of a decision table, which is known as an NP-hard nonlinearly constrained optimization problem, is equivalently transformed in this paper into an unconstrained binary optimization problem. An improved binary particle swarm optimization algorithm combined with some immunity mechanism is then proposed to solve the transformed optimization problem. Vaccination based on the discernibility matrix of the decision table is introduced for accelerating the search process in the algorithm. Experimental results on a number of data sets show that the proposed algorithm remarkably outperforms some recent global optimization techniques based algorithms for minimum attribute reduction in both quality of solution and computational complexity.

1 Introduction

Attribute reduction of a decision table based on the theory of rough sets has proved to be a very useful approach for knowledge discovery [1][2]. Finding just a reduct is usually not a difficult task and there have been many heuristic algorithms available in the literature for this purpose [3]-[6]. Computing a minimum attribute reduct that contains the least number of attributes is however more difficult and has been proved to be an NP-hard problem by Wong and Ziarko [7]. It turns out that the above mentioned algorithms are generally not effective since there is no guarantee for them to get a minimum reduct.

Formally, the minimum attribute reduction problem is a nonlinearly constrained combinatorial optimization problem. Hence, global optimization methods could be used to solve it. As a matter of fact, several algorithms along this direction have been investigated. For instance, Wroblewski [8] and Li [9] discussed in different ways the application of genetic algorithms(GAs) to the minimum attribute reduction problem and some interesting results were reported. More recently, the authors [10] and Dai [11] independently applied the binary swarm optimization algorithm due to Kennedy and Eberhart to deal with the problem and the results seemed to be encouraging. However, these algorithms are not quite effective in the sense that the probability for them to find a minimum reduct appears to be low. For some data sets, the algorithms may even

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1029–1036, 2007.

perform much worse than Hu and Cercone's heuristic reduction algorithm [3]. One reason is that the fitness functions used in these algorithms are not suitable enough for ensuring the optimality of the final results computed. This is due to kind of inappropriate use of the penalty function method in defining the fitness functions. Actually, a suitable fitness function is crucial to a successful application of evolutionary computation methods like genetic algorithms and particle swarm optimization algorithms. In this paper, the problem of minimum attribute reduction is transformed into an unconstrained binary optimization problem. A suitable fitness function is defined and the equivalence of optimality between the original problem and the transformed one is proved. An improved binary particle swarm optimization algorithm combined with some vaccination mechanism is then presented to solve the transformed problem. Experimental results on a number of data sets obtained from the UCI machine learning repository show that the proposed algorithm has a higher possibility of finding a minimum reduction and remarkably outperforms some existing algorithms specifically designed for minimum attribute reduction in both quality of solution and computational complexity.

The rest of the paper is organized as follows. Section 2 presents some background information on the attribute reduction problem. Section 3 describes how to equivalently transform the minimum attribute reduction problem into an unconstrained binary optimization problem by defining a suitable fitness function. In section 4, we give an improved binary swarm optimization algorithm for solving the transformed problem. In section 5, we present some experimental results and finally, in section 6, we conclude.

2 Minimum Attribute Reduction Problem

A decision table can be represented as a quadruple $L = \{U, A, V, f\}$ [1], where $U = \{x_1, \cdots, x_n\}$ is a non-empty finite set of objects called universe of discourse, A is a union of condition attributes set C and decision attributes set D, V is the domains of attributes belonging to A, and $f : U \times A \longmapsto V$ is an information function assigning attribute values to objects belonging to U. Assume that C contains m condition attributes $a_1, \cdots, a_m$ and without loss of generality that D contains only one decision attribute which takes $k(> 1)$ distinct values. For a subset $P \subseteq A$, $IND(P)$ represents the indiscernible relation induced by the attributes belonging to P and there should be no confusion if we use U to represent either a set of attributes or the relation $IND(P)$. A subset $X \subseteq U$ represents a concept and the partition induced by $IND(P)$ is called a knowledge base and denoted by $U/IND(P)$. In particular, $U/IND(D) = \{Y_1, \cdots, Y_k\}$ is the knowledge base of decision classes.

Let $X \subseteq U$ and $R \subseteq C$. The R–lower approximation of X is defined as $\underline{R}X = \{x \in U : [x]_R \subseteq X\}$, where $[x]_R$ refers to an equivalence class of $IND(R)$ determined by element x. The R–approximation quality with respect to decisions is given by $\gamma_R = \sum_{i=1}^{k} \frac{|\underline{R}Y_i|}{|U|}$, where $|\cdot|$ denotes the cardinality of a set. We restrict ourself to the classic reduction as defined in the following.

Definition 1. *Let $R \subseteq C$. If R is a minimal set satisfying $\gamma_R = \gamma_C$, then R is said to be a relative reduct of C or simply a reduct. The intersection of all reducts is called the attribute core of C and denoted as $Core(C)$.*

A minimum reduct is a reduct that contains the least number of attributes. Usually, there can be more than one minimum reduct. By definition, finding a minimum attribute reduction can be formulated as a nonlinearly constrained combinatorial optimization problem as follows:

$$\min |P|$$

$$s.t. \begin{cases} P \subseteq C \\ \gamma_P = \gamma_C \\ \forall q \in P, \gamma_{P \setminus \{q\}} < \gamma_P. \end{cases} \tag{1}$$

Let $\{0,1\}^m$ be the m-dimensional Boolean space and ξ be a mapping from $\{0,1\}^m$ to the power set 2^C such that:

$$x_i = 1 \Leftrightarrow a_i \in \xi(x), i = 1, \cdots, m, a_i \in C.$$

Then, the minimum reduction problem (1) can be reformulated as the following constrained binary optimization problem:

$$\min \ S(x)$$

$$s.t. \begin{cases} x \in \{0,1\}^m \\ \gamma_{\xi(x)} = \gamma_C \\ \forall q \in \xi(x), \gamma_{\xi(x) \setminus \{q\}} < \gamma_{\xi(x)} \end{cases} \tag{2}$$

where $0 \leq S(x) = \sum_{i=1}^{m} x_i \leq m$.

Given a vector $x \in \{0,1\}^m$, if it is a feasible solution to Problem (2), then its corresponding subset of attributes $\xi(x)$ is a reduct. Furthermore, if it is an optimal solution to Problem (2), then $\xi(x)$ is a minimum reduct.

The first PSO algorithm was introduced in 1995 by Kennedy and Eberhart [12] for continuous optimization problems and since then many improved versions of it have been presented [13][14]. It is a population-based optimization algorithm inspired by the social behavior of birds and, like other algorithms of its kind, it is initialized with a population of possible solutions (called particles) randomly located in a d-dimensional solution space. A fitness function determines the quality of a particle's position. A particle at time step t has a position vector and a velocity vector. The algorithm iterates updating the trajectories of the swarm through the solution space on the basis of information about each particle's previous best performance and the best previous performance of its neighbors until a stopping criterion is met. In 1997, Kennedy and Eberhart [15] developed a binary version of PSO for solving combinatorial optimization problems.

Usually, PSO algorithms can be directly applied to solve an unconstrained optimization problem since the fitness function can be defined in a straightforward way. However, when dealing with a constrained optimization problem,

things become a bit complicated. The most commonly used approach is to transform the constrained problem into an unconstrained one via the penalty function method. This amounts to defining a fitness function by enforcing the constraints into the objective function. However, if the penalty is not properly imposed on the fitness function, the transformation will not assure the equivalence of optimality between the two problems. Thus, it is important to define a suitable fitness function for ensuring a better performance of a PSO algorithm.

3 Transformation of the Minimum Reduction Problem

We shall discuss in this section how to equivalently transform Problem (2) into an unconstrained binary optimization problem that could be directly solved by a binary particle swarm optimization method.

Let us consider the following unconstrained binary optimization problem:

$$\max_{x\in\{0,1\}^m} \quad F(x) \tag{3}$$

where the fitness function is given by

$$F(x) = \begin{cases} m - S(x) + \gamma_{\xi(x)}, \gamma_{\xi(x)} < \gamma_C; \\ \gamma_C + 2m - S(x), \ \ \gamma_{\xi(x)} = \gamma_C. \end{cases}$$

We have the following main results concerning the equivalence between Problem (2) and Problem (3).

Theorem 1. *If x^* is an optimal solution to Problem (2), then x^* is also an optimal solution to Problem (3).*

Proof. Let $P = \xi(x^*)$. By hypothesis, we have $\gamma_P = \gamma_C$ and hence $F(x^*) = \gamma_C + 2m - S(x^*) \geq \gamma_C + m$. For any $x \in \{0,1\}^m$, let $R = \xi(x)$. If $\gamma_R < \gamma_C$, then by definition, $F(x) = m - s(x) + \gamma_R < m + \gamma_C \leq F(x^*)$. If $\gamma_R = \gamma_C$ and R is a reduct, then by the optimality of x^* and the definition of F, we have $S(x^*) \leq S(x)$ and $F(x^*) = \gamma_C + 2m - S(x^*) \geq \gamma_C + 2m - S(x) = F(x)$. If $\gamma_R = \gamma_C$ and R is not a reduct, then it means that R contains a reduct $R1 = \xi(x')$. Hence, $S(x') < S(x)$, yielding by definition $F(x') > F(x)$. Since $R1$ is now a reduct, we have $F(x^*) \geq F(x')$ and so $F(x^*) > F(x)$. The proof is thus complete.

Theorem 2. *Suppose that x^* is an optimal solution to Problem (3). Let $P = \xi(x^*)$. Then, x^* is also an optimal solution to Problem (2), or P is a minimum reduct.*

Proof. First, we show that $\gamma_P = \gamma_C$. We use proof by contradiction. Assume that $\gamma_P < \gamma_C$. Then, by definition, $F(x^*) = m - s(x^*) + \gamma_P < m + \gamma_C$. Let $y = (1, 1, \cdots, 1)^T \in \{0,1\}^m$. We have $\xi(y) = C$ and $S(y) = m$. By definition, $F(y) = \gamma_C + 2m - m > F(x^*)$, contradicting the optimality of x^*. Thus, $\gamma_P = \gamma_C$. Next, assume that there exists Q, a subset of P, such that $\gamma_Q = \gamma_C$. Let

$Q = \xi(x')$, then, $S(x') < S(x^*)$ and by definition, $F(x') = \gamma_C + 2m - S(x') > \gamma_C + 2m - S(x^*) = F(x^*)$, again contradicting the optimality of x^*. We have thus shown that x^* is a feasible solution of Problem (2).

Now, for any feasible solution $\hat{x}$ of Problem (2), we have $\gamma_R = \gamma_C$ with $R = \xi(\hat{x})$. By definition, $F(\hat{x}) = \gamma_C + 2m - S(\hat{x})$. Since x^* is a maximum of F by hypothesis, we have $\gamma_C + 2m - S(x^*) = F(x^*) \geq F(\hat{x})$, leading to $S(x^*) \leq S(\hat{x})$. This implies that x^* is also an optimal solution to Problem (2). The proof is thus done.

4 An Improved Binary PSO Algorithm with Vaccination

We present in this section an improved binary PSO algorithm with some vaccination mechanism for solving Problem (3). We choose to use Skowron's discernibility matrix of a decision table [16] as a criterion for preparing vaccines. The basic steps of our algorithm is as follows:

Step 1. Initialization. $\{0,1\}^m$ is the particle space; the size of the swarm is N; the maximum number of iterations is set to T; the velocity along each dimension is bounded by v_{max}.

Initialize the swarm as $P(0) = \{x^1(0), \cdots, x^N(0)\}$ and the corresponding velocity vectors as $v^i(0), i = 1, \cdots, N$. Initialize the i^{th} particle's previous best performance position $pb^i(0)$ as $x^i(0)$ and then identify the best previous performance position of the swarm as $gb(0)$. Set $t = 0$.

Step 2. Calculate the discernibility matrix M and $Core(C)$ [17]. If $Core(C) = \{a_{i_1}, \cdots, a_{i_q}\} \neq \emptyset$, then set $IC = \{i_1, \cdots, i_q\}$. For each attribute $a \in C \backslash Core(C)$, denote by $frq(a)$ the frequency of its occurrence in matrix M and set

$$h(a) = 0.8 \times \frac{frq(a) - fmin}{fmax - fmin} + 0.1$$

where $fmax$ and $fmin$ are respectively the maximum and minimum of all these frequencies.

Compute the vaccination pattern $HC = \{h(a_j) : j \in \{1, \cdots, m\} \backslash IC\}$.

Step 3. Update the positions and velocity of particles according to the following equations:

$$v_j^i(t+1) = w(t)v_j^i(t) + c_1 r_{1j}(t)(pb_j^i(t) - x_j^i(t)) + c_2 r_{2j}(t)(gb_j(t) - x_j^i(t)),$$

$$x_j^i(t+1) = \begin{cases} 1, rand_i < sig(v_j^i(t+1)); \\ 0, rand_i \geq sig(v_j^i(t+1)), \end{cases}$$

$$i = 1, \cdots, N,\ j \in \{1, \cdots, m\} \backslash IC$$

where $pb^i(t)$ is the previous best performance position of particle i and $gb(t)$ is the best previous performance position of the whole swarm; $w(t) = \frac{T-t}{T}$ is the inertia weight; $sig(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function; c_1 and c_2 are the learning coefficients; $r_{1j}(t), r_{2j}(t)$ and $rand_i$ are all uniformly randomized numbers in the interval [0,1].

Step 4. Vaccination. Randomly select a subset of particles and inject to them the vaccine pattern HC according to the following rule:

If i^{th} particle is chosen for vaccination, then its position after vaccination, denoted by $y^i(t)$, is computed as follows:

$$bound_j = \begin{cases} h(a_j) - 0.1, x_j^i(t) = 0; \\ h(a_j) + 0.1, x_j^i(t) = 1. \end{cases}$$

$$y_j^i(t) = \begin{cases} 0, bound_j \leq rand; \\ 1, bound_j > rand. \end{cases}$$

$$j \in \{1, \cdots, m\} \backslash IC$$

where *rand* is a uniformly randomized number in the interval [0,1].

An immune selection is then performed in such a way that if the vaccination increases the fitness value of a particle, then the particle is replaced by the vaccinated one.

Step 5. If some stopping criterion is met or $t >$T, then stop and output the subset of attributes $\xi(gb(t))$. Otherwise, set $t = t + 1$, repeat Step 3.

5 Experiments

To evaluate its performance, the proposed algorithm IPSO was implemented on a 2.8GHz machine running Windows XP with 512 MB of main memory and then tested on 5 real data sets obtained from the UCI machine learning repository. These data sets were chosen because Hu's reduction algorithm [3] fails to get a minimum reduct for each of them. For comparison, three minimum reduction algorithms, denoted by GA1 [9], PSO1 [10] and PSO2 [11] respectively, were also tested. Due to page limitation, we report here only the results corresponding to a specific setting of parameters. These parameters were as follows: T = 500, N = 20, the learning coefficients $c_1 = c_2 = 2$ for all PSO based algorithms, the crossover probability $p_c = 0.7$ and mutation probability $p_m = 0.01$ in algorithm GA1. In order to test how fast an algorithm can find a solution, a minimum reduct for each of these data sets was calculated beforehand via an exhaustive search and was then used to define the optimality stopping criterion. If an algorithm terminates with a solution satisfying the stopping criterion within the allowed iterations, then the solution corresponds to a minimum reduct and we say that this run of the algorithm is successful. Each algorithm was independently run 20 times in the experiments and three values were reported, including the number of attributes contained in the best solution found during the 20 runs, the ratio of successful runs and the mean computational time. The results are respectively listed in Tables 1, 2 and 3. For each test data set, the information on the number of attributes of a minimum reduct is also included in Table 1 under the column label *Known Best*.

Table 1 shows the best results picked up from among the 20 output solutions of the 20 runs of each algorithm. We see that our proposed algorithm could find a

Table 1. Best results found by the algorithms

dataset	Known Best	GA1	PSO1	PSO2	IPSO
zoo	5	5	5	5	5
house	4	4	4	4	4
lymphography	8	10	9	9	8
Soybean	9	12	10	10	9
Lung cancer	4	6	5	5	4

Table 2. Ratio of successful runs

dataset	GA1	PSO1	PSO2	IPSO
zoo	3/20	3/20	4/20	13/20
house	2/20	2/20	3/20	12/20
lymphography	0	0	0	9/20
Soybean	0	0	0	11/20
Lung cancer	0	0	0	10/20

minimum reduct for all test data sets, while none of the other algorithms could achieve this goal within the allowed iterations. Actually, the other algorithms could obtain a minimum reduct only for the first two data sets.

Table 2 shows how frequently an algorithm could find a minimum reduct during the 20 runs. It can be seen that for each test data set, the proposed algorithm had a high ratio of successful runs or a high probability of getting a minimum reduct, whereas the other algorithms rarely had successful runs.

Table 3. Average computational time (second)

dataset	GA1	PSO1	PSO2	IPSO
zoo	43	36	31	18
house	102	93	90	61
lymphography	697	582	567	429
Soybean	996	895	854	597
Lung cancer	933	751	768	512

Table 3 shows the experimental results on the average computational time of each algorithm. Obviously, the proposed algorithm performs better than the others.

6 Conclusion

We have studied in this paper the problem of how to effectively compute a minimum reduct of a decision table based on PSO algorithms. By defining a suitable fitness function for the evaluation of a particle's quality, we developed an improved binary PSO algorithm that outperforms some recent global optimization

techniques based algorithms for minimum attribute reduction. This suggests that binary PSO based algorithms could be promising and even competent in solving the minimum attribute reduction problem.

Acknowledgement. This work was partly funded by National Science Foundation of China(No.60602052) and by Fujian Science Foundation(No.2006J0029).

References

1. Pawlak Z., Slowinski R., Rough set approach to multi-attribute decision analysis. European J. of Operational Research, **72**(1994) 443–459
2. Wang G.Y., Rough set and knowledge aquisition, Xian Jiaotong Unversity Press, 2001
3. Hu X.H., Cercone N., Learning in relational databases: a Rough Set approach, Computational Intelligence, **11**(1995)323–338
4. Jelonek J., et al, Rough set reduction of attributes and their domains for neural networks, Computational Intelligence, **11**(1995) 339–347
5. Wang J., Wang R., Miao D.Q., Data enrichment based on rough set theory, Chinese J. of Computers, **21**(1998) 393–400
6. Ye D.Y., An improvement to Jelonek's attribute reduction algorithm, Acta Electronica Sinica, **28**(2000) 81-82
7. Wong S. K. M., Ziarko W., On optimal decision rules in decision tables, Bulletin of Polish Academy of Science, **33**(1985) 693–696
8. Wroblewski J., Finding mininmal reducts using genetic algorithm. ICS Research Report 16/95. Warsaw university of Technology, 1995
9. Li D.F., et al, Genetic reduction algorithm based on feasible region, Mini-Micro Systems, **27**(2006) 312–315
10. Ye D.Y., Liao J.K., A particle swarm optimization algorithm for minimum attribute reduction, Advances of AI in China 2005, Beijing Post and Telecommunication University Press, (2005) 728–732
11. Dai J.H., et al., Particle swarm algorithm for minimal attribute reduction of decision data tables, Proceedings of IMSCCS 2006, IEEE Computer Society Press, (2006) 12–18
12. Kennedy J., Eberhart R.C., Particle swarm optimization, IEEE International Conf. on Neural Networks, Piscataway, NJ: IEEE Service Center, (1995)1942–1948
13. Shi Y.H., Eberhart R.C., A modified particle swarm optimizer, IEEE International Conf. on Evolutionary Computation, Piscataway, NJ: IEEE Press, (1998) 69–73
14. Clerc M., Discrete Particle Swarm Optimization, New Optimization Techniques in Engineering, Heidelberg, Germany: Springer-Verlag, 2004
15. Kennedy J., Eberhart R.C., A discrete binary version of the particle swarm algorithm. In: Proceedings of the International Conf. on Systems, Man and Cybernetics, Piscataway: IEEE Press, (1997) 4104–4109
16. Skowron A., Rauszer C., The discernibility matricesand functions in information systems, Slowinski I. edited, Intelligent Decision Support-Handbook of Applications and Advances of the Rough Sets Theory. Dordrecht: Kluwer, (1991) 331–362
17. Ye D.Y., Chen Z.J., A new discernibility matrix and the computation of a core, Acta Eletronica Sinica, **30**(2002) 1086–1088

Graph Nodes Clustering Based on the Commute-Time Kernel

Luh Yen[1], Francois Fouss[1], Christine Decaestecker[2], Pascal Francq[3], and Marco Saerens[1]

[1] Université catholique de Louvain, ISYS, IAG, Louvain-la-Neuve, Belgium
{luh.yen, francois.fouss, marco.saerens}@uclouvain.be
[2] Université libre de Bruxelles, Institut de Pharmacie, Bruxelles, Belgium
cdecaes@ulb.ac.be
[3] Université libre de Bruxelles, STIC, Bruxelles, Belgium
pfrancq@ulb.ac.be

Abstract. This work presents a kernel method for clustering the nodes of a weighted, undirected, graph. The algorithm is based on a two-step procedure. First, the sigmoid commute-time kernel ($\mathbf{K}_{\mathrm{CT}}$), providing a similarity measure between any couple of nodes by taking the indirect links into account, is computed from the adjacency matrix of the graph. Then, the nodes of the graph are clustered by performing a kernel k-means or fuzzy k-means on this CT kernel matrix. For this purpose, a new, simple, version of the kernel k-means and the kernel fuzzy k-means is introduced. The joint use of the CT kernel matrix and kernel clustering appears to be quite successful. Indeed, it provides good results on a document clustering problem involving the newsgroups database.

1 Introduction

This work presents a general methodology for clustering the nodes of a weighted, undirected, graph. Graph nodes clustering is an important issue that has been the subject of much recent work; see for instance [4], [5], [7], [11], [17] and [19].

On the other hand, kernel-based algorithms are characterized by two properties: they allow (i) to compute implicitly similarities in a high-dimensional space where the data are more likely to be well-separated and (ii) to compute similarities between structured objects that cannot be naturally represented by a simple set of features. In this paper we propose a new kernel matrix on a weighted, undirected, graph, which defines similarities between the nodes. These similarities take both direct and indirect links into account; they therefore take the indirect paths between the nodes into consideration. Two nodes are considered as similar if there are many short paths connecting them.

Based on this kernel matrix, nodes are clustered thanks to a kernel clustering. The kernel clustering algorithms proposed in this paper differ from existing ones ([2], [9], [10], [20], [22] and [23]) by the fact that a prototype vector is explicitly defined for each cluster. This is more natural since it allows to mimic the iterative update rules reminiscent from k-means and fuzzy k-means in the sample space,

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1037–1045, 2007.

instead of the feature space. In addition to be very similar to the original feature-based algorithms, this sample-based method can easily be extended to variable-metric or multi-prototype kernel k-means, in the same way as the original k-means and fuzzy k-means [6]. In addition to this, the resulting algorithm is very simple and natural.

The performances are evaluated on the problem of clustering newsgroups documents, and compared to the popular spherical k-means algorithm, which is especially designed for document clustering [3], as well as a classic spectral clustering method [12]. The collection of documents is viewed as a graph and the basic problem is to cluster the documents in order to eventually retrieve the newsgroups. The results indicate that the introduced algorithms perform well in comparison with the spherical k-means and the spectral clustering, with significant improvement.

The paper is organized as follows. Section 2 introduces the sigmoid commute-time kernel ($\mathbf{K}_{\mathrm{CT}}$) on a graph that will be used as similarity measure for clustering the nodes. Section 3 derives our version of the kernel k-means and kernel fuzzy k-means, while Section 4 shows the results obtained on the newsgroups database. Section 5 is the conclusion.

2 The Sigmoid Commute-Time Kernel on a Graph

Let us consider that we are given a weighted, undirected, graph, G, with symmetric weights $w_{ij} > 0$ on the edges connecting pairs of nodes i, j. The elements a_{ij} of the adjacency matrix $\mathbf{A}$ of the graph are defined in a standard way as $a_{ij} = w_{ij}$ if node i is connected to node j and 0 otherwise. Based on the adjacency matrix, the Laplacian matrix $\mathbf{L}$ of the graph is defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \mathbf{Diag}(a_{i.})$ is the degree matrix, with diagonal entries $d_{ii} = [\mathbf{D}]_{ii} = a_{i.} = \sum_{j=1}^{n} a_{ij}$. We suppose that the graph has a single connected component; that is, any node can be reached from any other node of the graph. In this case, $\mathbf{L}$ has rank $n - 1$, where n is the number of nodes. Moreover, it can be shown that $\mathbf{L}$ is symmetric and positive semidefinite (see for instance [8]).

The "commute time" kernel [14], [8] takes its name from the **average commute time**, $n(i, j)$, which is defined as the average number of steps a random walker, starting in node $i \neq j$, will take before entering a node j for the first time, and go back to i. Indeed, we associate a Markov chain to the graph in the following obvious manner. A state is associated to every node (n in total), and the transition probabilities are given by $p_{ij} = a_{ij}/a_{i.}$ where $a_{i.} = \sum_{j=1}^{n} a_{ij}$. One can show [14], [8] that, in this case, the average commute time can be computed thanks to $n(i, j) = V_G\,(\mathbf{e}_i - \mathbf{e}_j)^{\mathrm{T}} \mathbf{L}^{+} (\mathbf{e}_i - \mathbf{e}_j)$ where every node i of the graph is represented by a basis vector, $\mathbf{e}_i$ (the i-th column of the identity matrix $\mathbf{I}$), in the Euclidean space $\Re^n$ and $V_G = a_{..}$ is the volume of the graph. $\mathbf{L}^{+}$ is the Moore-Penrose pseudoinverse of the Laplacian matrix of the graph and is positive semidefinite. Thus, $n(i, j)$ is a **Mahalanobis distance** between the nodes of the graph and is referred to as the "commute time distance" or the

"resistance distance" because of a close analogy with the effective resistance in electrical networks [8].

One can further show that $\mathbf{L}^+$ is the matrix containing the inner products of the node vectors in the Euclidean space where these node vectors are exactly separated by commute time distances. In other words, the entries of $\mathbf{L}^+$ can be viewed as similarities between nodes and $\mathbf{L}^+$ can be considered as a kernel matrix:

$$\mathbf{K} = \mathbf{L}^+ \tag{1}$$

The **sigmoid commute time kernel** $\mathbf{K}_{\mathrm{CT}}$ is obtained by applying a sigmoid transform [15] on $\mathbf{K}$. In other words, each element of the kernel matrix is given by the formula

$$[\mathbf{K}_{\mathrm{CT}}]_{ij} = 1/(1 + \exp[a\ l_{ij}^+/\sigma]) \tag{2}$$

where $l_{ij}^+ = [\mathbf{L}^+]_{ij}$ and σ is a normalizing factor, corresponding to the standard deviation of the elements of $\mathbf{L}^+$. The parameter a will be set to a constant value determined by informal preliminary tests. The sigmoid function aims to normalize the range of the similarities in the interval $[0, 1]$ [15]. Notice, however, that the resulting matrix is not necessarily positive semi-definite so that, strictly speaking, it is not a kernel matrix.

3 Kernel k-Means and Fuzzy k-Means

We now introduce our kernel, prototype-based, version of the k-means and fuzzy k-means clustering algorithms.

3.1 Kernel k-Means

The goal is to design an iterative algorithm aiming to minimize a cost function which, in the case of a standard k-means, can be defined, in the feature space, as the total within-cluster inertia:

$$J(\mathbf{g}_1, \ldots, \mathbf{g}_m) = \sum_{k=1}^{m} \sum_{i \in C_k} ||\mathbf{x}_i - \mathbf{g}_k||^2 \tag{3}$$

where the first sum is taken on the m clusters, while the second sum is taken on the nodes i belonging to cluster k, $i \in C_k$. In Equation (3), $\mathbf{x}_i$ is the feature vector corresponding to node i, $\mathbf{g}_k$ is a prototype vector of cluster k in the **feature space** and $||\mathbf{x}_i - \mathbf{g}_k||$ is the Euclidean distance between the node vector and the cluster prototype it belongs to. The number of clusters, m, is provided a priori by the user.

We denote by $\mathbf{X}$ the data matrix containing the transposed node vectors as rows, that is, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^{\mathrm{T}}$. Let us now define the following change of parameter:

$$\mathbf{g}_k \rightarrow \mathbf{X}^{\mathrm{T}} \mathbf{h}_k \tag{4}$$

corresponding to the "kernel trick" (see [16]). It aims to express the prototype vectors, $\mathbf{g}_k$, as a linear combination of the node vectors, $\mathbf{x}_i$ (the columns of

$\mathbf{X}^{\mathrm{T}}$). The $\mathbf{h}_k$ will be called the prototype vectors in the n-dimensional **sample space**. Now, recompute the within-class inertia in terms of the $\mathbf{h}_k$ and the inner products:

$$\begin{aligned} J(\mathbf{h}_1,\ldots,\mathbf{h}_m) &= \sum_{k=1}^{m}\sum_{i\in C_k}(\mathbf{x}_i-\mathbf{g}_k)^{\mathrm{T}}(\mathbf{x}_i-\mathbf{g}_k) \\ &= \sum_{k=1}^{m}\sum_{i\in C_k}(k_{ii}-2\mathbf{k}_i^{\mathrm{T}}\mathbf{h}_k+\mathbf{h}_k^{\mathrm{T}}\mathbf{K}\mathbf{h}_k) \\ &= \sum_{k=1}^{m}\sum_{i\in C_k}(\mathbf{e}_i-\mathbf{h}_k)^{\mathrm{T}}\mathbf{K}(\mathbf{e}_i-\mathbf{h}_k) \end{aligned} \tag{5}$$

where $\mathbf{K}=\mathbf{X}\mathbf{X}^{\mathrm{T}}$, $k_{ii}=[\mathbf{K}]_{ii}=\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_i$, $\mathbf{k}_i=\mathbf{X}\mathbf{x}_i=\mathbf{col}_i(\mathbf{K})$.

The k-means iteratively minimizes J by proceeding in two steps, (1) re-allocation of the node vectors while keeping the prototype vectors fixed, and (2) re-computation of the prototype vectors, $\mathbf{h}_k$, while maintaining the cluster labels of the nodes fixed. Clearly, the re-allocation step minimizing J is

$$l_i = \arg\min_k \left\{(\mathbf{e}_i-\mathbf{h}_k)^{\mathrm{T}}\mathbf{K}(\mathbf{e}_i-\mathbf{h}_k)\right\} \tag{6}$$

where l_i contains the cluster label of node i.

For the computation of the prototype vector, by taking the gradient of J with respect to $\mathbf{h}_k$ and setting the result equal to $\mathbf{0}$, we obtain $\mathbf{K}\mathbf{h}_k=\frac{1}{n_k}\sum_{i\in C_k}\mathbf{k}_i=\mathbf{K}\frac{1}{n_k}\sum_{i\in C_k}\mathbf{e}_i$ where n_k is the number of nodes belonging to cluster k. By looking carefully, we immediately observe from the left-hand side of the equation that $\mathbf{K}\mathbf{h}_k$ is a linear combination of the $\mathbf{k}_i$, while the right-hand side is also a linear combination of the $\mathbf{k}_i$. Therefore, one solution to this linear system of equations is simply the following:

$$\mathbf{h}_k = \frac{1}{n_k}\sum_{i\in C_k}\mathbf{e}_i \tag{7}$$

In other words, $\mathbf{h}_k$ contains $1/n_k$ if $i\in C_k$ and 0 otherwise. This two-step procedure (equations (6) and (7))is iterated until convergence.

3.2 Kernel Fuzzy k-Means

We now apply the same procedure for deriving a kernel fuzzy k-means. This time, the cost function is

$$J(\mathbf{g}_1,\ldots,\mathbf{g}_m;\mathbf{U}) = \sum_{k=1}^{m}\sum_{i=1}^{n}u_{ik}||\mathbf{x}_i-\mathbf{g}_k||^2 \text{ with } \sum_{k=1}^{m}u_{ik}^{1/q}=1 \text{ for all } i \tag{8}$$

where the u_{ik} define the degree of membership of node i to cluster C_k. The parameter $q>1$ is controlling the degree of fuzzyness of the membership functions.

As for the kernel k-means, we perform the change of parameter (4), leading to the following update formula for the membership function.

$$u_{ik} = \left[\frac{\left((\mathbf{e}_i - \mathbf{h}_k)^{\mathrm{T}} \mathbf{K} (\mathbf{e}_i - \mathbf{h}_k)\right)^{-1/(q-1)}}{\sum_{l=1}^{m} \left((\mathbf{e}_i - \mathbf{h}_l)^{\mathrm{T}} \mathbf{K} (\mathbf{e}_i - \mathbf{h}_l)\right)^{-1/(q-1)}} \right]^{q} \tag{9}$$

and the re-computation of the prototype vectors is simply,

$$h_{ki} = [\mathbf{h}_k]_i = \frac{u_{ik}}{\sum_{j=1}^{n} u_{jk}} \tag{10}$$

4 Experiments

4.1 Data Set

In order to test the performances of the $\mathbf{K}_{\mathrm{CT}}$ k-means and the $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means, both algorithms will be assessed on a real data set and compared to classical clustering algorithms. The idea is to assess both algorithms on graph data set where only the information on relation between nodes is given. The tested graphs are extracted from the newsgroups data set (Available from http://people.csail.mit.edu/jrennie/20Newsgroups/); it is composed of 20,000 unstructured documents, taken from 20 discussion groups (newsgroups) of the Usernet diffusion list. As the data set is composed of documents, the clustering performances of both methods will be compared to the spherical k-means [3], which is a reference in text mining; and to Ng's spectral clustering [12], which presents some similarities with our approach.

For our experiment, 9 subsets including different topics are extracted from the original database, as listed in figure 1. More precisely, for each subset, 200 documents are sampled from different newsgroups. Thus, the three first subsets (G-2cl-A, G-2cl-B, G-2cl-C) contain 400 documents sampled from two newsgroups topics, the next three subsets (G-3cl-A, G-3cl-B, G-3cl-C) contain 600 documents sampled from three topics and the last three subsets (G-5cl-A, G-5cl-B, G-5cl-C) contain 1000 documents sampled from five topics. The selected topics can be related such as politics/mideast and politics/guns in subset G-5cl-A. Both the classification rate (obtained by comparing the clustering to the real newsgroups and performing an optimal assignment) and the adjusted Rand index (with values scaled in $[0, 1]$) will be reported.

4.2 Graph Definition

The newsgroups data set can be seen as a large bipartite graph between documents and terms. Each document node is connected to terms nodes contained

G-2cl-A	politics/general, sport/baseball
G-2cl-B	computer/graphics, motor/motorcycles
G-2cl-C	space/general, politics/mideast
G-3cl-A	sport/baseball, space/general, politics/mideast
G-3cl-B	computer/windows, motor/autos, religion/general
G-3cl-C	sport/hockey, religion/atheism, medicine/general
G-5cl-A	computer/windowsx, cryptography/general, politics/mideast, politics/guns, religion/christian
G-5cl-B	computer/graphics, computer/pchardware, motor/autos, religion/atheism, politics/mideast
G-5cl-C	computer/machardware, sport/hockey, medicine/general, religion/general, forsale/general

Fig. 1. Document subsets used in our experiments. Nine subsets are selected from the *Newsgroups* data set, with 2, 3 or 5 topics. For each subset, 200 documents are randomly selected from each topic.

in the document, each edge being weighted by the *tf.idf* factor [18]. After some preprocessing steps (see below) aiming to reduce the number of terms, a graph involving only documents is computed from this bipartite graph in the following way: the link between two documents is given by the sum of all document-term-document paths connecting them and passing through the terms they have in common. In other words, if $\mathbf{W}$ represents the term-document matrix containing the *tf.idf* factors, the adjacency matrix of the resulting document-document graph is provided by $\mathbf{A} = \mathbf{W}^{\mathrm{T}}\mathbf{W}$.

4.3 Preprocessing Steps

In order to reduce the high dimensionality of the feature space (terms), the following standard preprocessing steps are performed on the data set before the clustering experiment.

1. Stopwords without useful information are eliminated.
2. Porter's stemming algorithm [13] is applied so that each word is reduced to its "root ".
3. Words that occur too few times (< 3) or in too few documents (< 2) are considered as no content-bearing and are eliminated.
4. The mutual information between terms and documents is computed. For a word y, the mutual information with the documents of the data set [21] is given by
$$I(y) = \sum_x \log p(x, y)/p(x)p(y), \tag{11}$$
where x represents the documents of the data set. Words with a small value of mutual information (fixed at 20% of $I(y)$'s median) are eliminated.
5. The term-document matrix $\mathbf{W}$ is constructed with the remaining words and documents. Element $[\mathbf{W}]_{ij}$ of the matrix contains the value of *tf.idf* factor between the term i and the document j.
6. Each row of the term-document matrix $\mathbf{W}$ is normalized to 1.

Finally, the adjacency matrix of the documents graph $\mathbf{A}$ is given by the document-document matrix $\mathbf{W}^{\mathrm{T}}\mathbf{W}$. Based on $\mathbf{A}$, $\mathbf{K}_{\mathrm{CT}}$ is computed by Equation (2). For example, the subset G-2cl-A is composed of 400 documents, and 2898 terms with stopwords already eliminated. After preprocessing, only 1490 terms are kept. Thus, the clustering algorithm will be run on a 400×400 document-document matrix, instead of a 1490×400 term-document matrix for a standard feature-based algorithm.

4.4 Experimental Settings

Suppose we have a graph of n nodes to be partitioned into m clusters. First, the prototype vectors $\mathbf{h}_i$ $(i = 1, ..., m)$ are initialized by randomly selecting m columns of the identity matrix $\mathbf{I}$. Then, each algorithm is run 30 times (30 runs), and the classification rate as well as the adjusted Rand index, averaged on the 30 runs, are computed. The $\mathbf{K}_{\mathrm{CT}}$ k-means, $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means and Ng's spectral clustering are run on the document-document matrix $\mathbf{A}$, while the spherical k-means is run on the term-document matrix $\mathbf{W}$ after preprocessing.

Each run consists in 50 trials: the clustering algorithm is launched 50 times and the best solution among the 50 trials, having the minimal within-class inertia, is sent back as the solution.

Two parameters need to be tuned. The first one is the parameter a for computing the sigmoid transform of the $\mathbf{K}_{\mathrm{CT}}$ (see Equation (2)). The second one is the parameter q which controls the degree of fuzzyness for the kernel fuzzy k-means (see Equation (9)). Based on preliminary informal experiment, the parameters a and q were set to 7 and 1.2 respectively, for all experiments.

4.5 Experimental Results and Discussion

The results (the classification rate as well as the adjusted Rand index, each averaged on 30 runs) of the four clustering algorithms ($\mathbf{K}_{\mathrm{CT}}$ k-means, $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means, spherical k-means and Ng's spectral clustering) on the nine document subsets are reported in Table 1.

We observe that the $\mathbf{K}_{\mathrm{CT}}$ k-means and the $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means outperform the spherical k-means on the nine subsets. Ng's spectral clustering presents good results on the 2-classes and 3-classes data sets, but degrades when the number of clusters increases. Moreover, the $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means provides slightly better results than the two other methods. This can be partly explained by the fact that the newsgroups data set is fuzzy itself, as discussed in [1]. It is hard to define clear boundaries between the different topics: a discussion within a specific newsgroup can also be related to other domains. A close examination of the data set shows that several discussions can even be out of subject or are simply empty of useful information.

Table 1. Comparison of the clustering performances (classification rate in % and adjusted Rand index with value scaled in $[0, 1]$) for the $\mathbf{K}_{\mathrm{CT}}$ k-means, $\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means, spherical k-means and Ng's spectral clustering

	$\mathbf{K}_{\mathrm{CT}}$ k-means		$\mathbf{K}_{\mathrm{CT}}$ fuzzy k-means		Sph. k-means		Ng spec. clus.	
	class. rate	adj. Rand	class. rate	adj. Rand	class. rate	adj. Rand	class. rate	adj. Rand
G-2cl-A	97.5 %	0.95	97.8 %	0.96	91.8 %	0.85	94.5 %	0.90
G-2cl-B	90.6 %	0.83	91.5 %	0.84	81.5 %	0.70	93.0 %	0.87
G-2cl-C	95.5 %	0.91	96.0 %	0.92	94.8 %	0.90	95.7 %	0.92
G-3cl-A	93.9 %	0.91	94.5 %	0.92	89.2 %	0.85	92.7 %	0.90
G-3cl-B	93.6 %	0.91	93.5 %	0.91	86.7 %	0.82	92.0 %	0.89
G-3cl-C	93.9 %	0.91	92.8 %	0.90	87.4 %	0.83	81.7 %	0.78
G-5cl-A	83.0 %	0.80	85.4 %	0.83	80.4 %	0.79	76.7 %	0.78
G-5cl-B	74.8 %	0.77	78.4 %	0.79	64.4 %	0.69	67.7 %	0.72
G-5cl-C	76.4 %	0.75	80.1 %	0.79	64.9 %	0.69	64.0 %	0.72

5 Conclusions and Further Work

We introduced a new method allowing to cluster the nodes of a weighted graph by exploiting the links between them. It is based on a recently introduced kernel on a graph, the commute-time kernel, combined with a kernel clustering. The obtained results are promising since the proposed methodology outperforms the standard spherical k-means as well as spectral clustering on a difficult graph clustering problem. Further work will be devoted to (1) additional experiments on other text databases, and (2) developing kernel versions of the Gaussian mixture, the entropy-based fuzzy clustering, Ward's hierarchical clustering, and assessing their performances.

References

1. M. W. Berry, editor. *Survey of Text Mining*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
2. I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means, spectral clustering and normalized cuts. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM Press, 2004.
3. I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.
4. C. Ding and X. He. Linearized cluster assignment via spectral ordering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 30, New York, NY, USA, 2004. ACM Press.
5. P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56(1-3):9–33, 2004.
6. B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. Arnold Publishers, 2001.
7. G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Math*, 1(4):385–408, 2003.

8. F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):355–369,, 2007.
9. M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, May 2002.
10. D.-W. Kim, K. Y. Lee, D. Lee, and K. H. Lee. Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recognition*, 38(4):607–611, 2005.
11. M. Newman. Detecting community structure in networks. *The European Physical Journal B*, 38:321–330, 2004.
12. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processiong Systems*, volume 14, pages 849–856, Vancouver, Canada, 2001. MIT Press.
13. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
14. M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. *Proceedings of the 15th European Conference on Machine Learning (ECML 2004). Lecture Notes in Artificial Intelligence, Vol. 3201, Springer-Verlag, Berlin*, pages 371–383, 2004.
15. B. Scholkopf and A. Smola. *Learning with kernels*. The MIT Press, 2002.
16. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
17. S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.
18. S. Weiss, N. Indurkhya, T. Zhang, and F. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Springer, 2004.
19. S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *SDM*, 2005.
20. Z.-D. Wu, W.-X. Xie, and J.-P. Yu. Fuzzy c-means clustering algorithm based on kernel method. In *ICCIMA '03: Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications*, page 49, Washington, DC, USA, 2003. IEEE Computer Society.
21. H. Zha, X. He, C. H. Q. Ding, M. Gu, and H. D. Simon. Bipartite graph partitioning and data clustering. In *Proc. of ACM 10th Int'l Conf. Information and Knowledge Management (CIKM 2001)*, pages 25–32, 2001.
22. D.-Q. Zhang and S.-C. Chen. Fuzzy clustering using kernel method. In *Proceedings of the 2002 International Conference on Control and Automation, 2002. ICCA*, pages 162–163, 2002.
23. D.-Q. Zhang and S.-C. Chen. A novel kernelized fuzzy c-means algorithm with application in medical image segmentation. *Artificial Intelligence in Medicine*, 32(1):37–50, 2004.

Identifying Synchronous and Asynchronous Co-regulations from Time Series Gene Expression Data

Ying Yin, Yuhai Zhao, and Bin Zhang

Department of Computer Science and Engineering, Northeastern University
Shengyang 110004, P.R. China
yy_00000000@163.com

Abstract. The complexity of a biological system provides a great diversity of correlations among genes/gene clusters, including synchronous and asynchronous co-regulations, each of which can be further divided into two categories: activation and inhibition. Most existing methods can only identify the synchronous activation patterns, such as shifting, scaling and shifting-and-scaling, however, few focuses on capturing both synchronous and asynchronous co-regulations. In this paper, we propose a coding scheme, where two genes with the same code must be co-regulated. Based on the coding scheme, an efficient clustering algorithm is devised to simultaneously capture all known co-regulated relationships (synchronous and asynchronous) among genes/gene clusters. Furthermore, the detailed and complete co-regulation information, which facilitates the study of genetic regulatory networks, can be easily derived from the resulting clusters. Experiments from both real and synthetic microarray datasets prove the effectiveness and efficiency of our method.

1 Introduction

The complexity of biological systems provides a great diversity of correlations among genes/gene clusters. Analysis of these regulatory relationships can provide insights into the interactions of genes/gene clusters, which facilitates the study of genetic regulatory networks.

Table 1(a) shows an example of microarray dataset, D, consisting of a set of rows and a set of columns, where the rows denote genes, $G = \{g_1, g_2..., g_m\}$, and the columns denote different time points, $T = \{t_1, t_2..., t_n\}$. Note that the expression value of a gene, g_i, on a certain time point, t_j, is denoted by $d_{i,j}$. For simplicity, certain cells have been left blank in the table. We assume that these are filled by some random expression values. Table 1(b) is a transposed version of the running example in Table 1(a) after some row permutations, where two different regulation groups emerge. The first one, shadowed and enveloped by a solid polygon, is plotted in Figure 1(a) against every gene's expression profile within it. Similarly, Figure 1(b) corresponds to the second one not shadowed but enveloped by a dashed rectangle. Note: any pair of genes within a regulation group must have one of known regulatory relationships. For example, when T=$\{t_1, t_2, t_4, t_5\}$, in Figure 1(a), genes 1 and 4 present the shifting pattern [1,2]

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1046–1054, 2007.

since $d_{1,T}$=$d_{4,T}$+25. In Figure 1(b), where $T' = \{t_1, t_3, t_5, t_6, t_7\}$, genes 3 and 6 present the scaling pattern [3] since $d_{6,T'}=3 \times d_{3,T'}$, and genes 3 and 8 present the shifting-and-scaling pattern [3,4] since $d_{8,T'}=4 \times d_{3,T'} + 5$. All co-regulations shown in Figure 1(b) are *synchronous* since the products of a gene immediately affect other genes' expression. Moreover, all synchronous co-regulations [5] can be generalized into two categories: *activation* and *inhibition*. In the activation process, an increase (resp. decrease) in certain genes' expression levels will increase (resp. decrease) some other genes' expression levels, such as the simultaneous pattern [6] between genes 1 and 4 in Figure 1(a), but during the inhibition process, the case is just the reverse, such as the inverted pattern [6] between genes 1 and 5.

Table 1. A Matrix for a Simple Microarray Dataset

(a) Example Microarray Dataset.

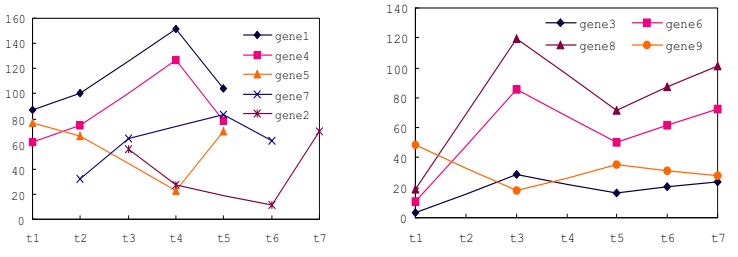
	t_1	t_2	t_3	t_4	t_5	t_6	t_7
g_1	86.9	100		151.5	104		
g_2			55.4	27		11	78
g_3	3.5		28.6		16.65	20.5	24.125
g_4	61.9	75		126.5	79		
g_5	76.6	65.9		23.0	70.0		
g_6	10.5		85.8		49.95	61.5	72.37
g_7		32	64.6		83.1	62.8	72.37
g_8	19		119.4		71.6	87	101.5
g_9	48.5		18.2		35.35	31.5	27.875

(b) Some Clusters.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
g_1	86.9	100		151.5	104		
g_4	61.9	75		126.5	79		
g_5	76.6	65.9		23	70		
g_7		32	64.6		83.1	62.8	
g_2			55.4	27		11	78
g_3	3.5		28.6		16.65	20.5	24.125
g_6	10.5		85.8		49.95	61.5	72.375
g_8	19		119.4		71.6	87	101.5
g_9	48.5		18.2		35.35	31.5	27.875

In fact, from time-series gene expression data, it is apparent that most genes do not co-regulate each other simultaneously but after a certain time lag [7], which we call *asynchronous co-regulation* as shown in Figure 1(a). Also, it is divided into activation, such as time-shifting pattern between genes 1 and 7, and inhibition, such as inverted-time-shifting pattern between genes 1 and 2.

Existing methods used for identifying regulatory relationships from microarray data fall into two major categories: the pattern/tendency-based subspace clustering [8,9] and the 'two genes, one relationship per alignment' approach [6].



(a) The first regulation group. (b) The second regulation group.

Fig. 1. Two regulation groups

However, the former usually considers gene expression levels for pure shifting [2,1] or pure scaling [3] patterns under the same subset of conditions, and *does not take any synchronous relationships into consideration.* So it ignores many additional relationships implicit in expression time-course. The process of the latter can be characterized as 'two genes, one relationship per alignment', which means that each alignment can decide only one relationship between two genes. Such an approach in some ways is not very computationally efficient. Moreover, since this approach evaluates the expression profile similarity of genes over all conditions, it is not sensitive to the case where a small but interesting part of the genes is co-regulated while there is no distinct relationship between the remaining parts.

The main contributions of this work are: (1) We propose a new clustering model, namely $\mathcal{R}$eg-$\mathcal{C}$luster, to capture all synchronous and asynchronous co-regulation patterns in a holistic manner, which is a generalization of the pattern/tendency-based subspace clustering. (2) We propose a new coding-based approach with which two genes are co-regulated if they have the same gcode, and propose a new tree-based clustering algorithm, i.e. FBLD, with some pruning rules, to efficiently find all significant reg-clusters. (3) Based on the proposed coding schema, the more detailed co-regulation information can be easily derived from the resulting clusters, such as activation or inhibition and how many time points lagged between activated or inhibited genes. (4) We conducted extensive experimental studies on both real data sets and synthetic data sets to confirm the effectiveness and efficiency of our algorithm.

The remainder of this paper is organized as follows: Section 2 presents the $\mathcal{R}$eg-$\mathcal{C}$luster model and the problem statement. Section 3 gives the FBLD algorithm in detail. Experimental results and analysis are shown in Section 4. Finally, Section 5 concludes this paper.

2 The $\mathcal{R}$eg-$\mathcal{C}$luster Model

Let $G = \{g_1, g_2, ..., g_m\}$ be a set of m genes, and $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ be a set of n experimental time points. A two dimensional microarray time series dataset is a real-valued $m \times$n matrix $D = G \times T = \{d_{ij}\}$, where $i \in [1, m]$, $j \in [1, n]$, two dimensions of which correspond to genes and times respectively. Each entry d_{ij} records the expression value of gene g_i at time point t_j.

Definition 1. *l-segment. Suppose the original time sequence* $T = <t_1, t_2, ..., t_n>$ *and its subsequence* $T' = < t_{i_1}, t_{i_2}, \cdots, t_{i_{l+1}} >$. *There are l neighboring prototypal subsequence of length 2 in* T', *i.e.* $<t_{i_1}, t_{i_2}>$, $<t_{i_2}, t_{i_3}>$, $\cdots$, $<t_l, t_{l+1}>$. *We call* T' *to be an l-segment regarding the number of prototypal subsequence of length 2, because a prototypal subsequence of length 2 is a basic regulation unit. The number of elements in* T', *denoted* $|T'|$, *is called the length of* T'.

Definition 2. *Significant Regulation. Given a gene,* g_a, *and a* 1*-segment,* $<t_{i_j}, t_{i_k}>$, *we say the regulation of gene* g_a *from time point* t_{i_j} *to* t_{i_k} *is significant. A significant regulation is* up-regulated *when* $d_{a,i_k} - d_{a,i_j} > \delta$, *denoted* $\mathcal{R}eg(g_a, (<t_{i_j}, t_{i_k}>)) =$ '↗', *and* down-regulated *when* $d_{a,i_k} - d_{a,i_j} < \delta$, *denoted* $\mathcal{R}eg(g_a, (<t_{i_j}, t_{i_k}>)) =$ '↘'.

Definition 3. *gCode. Given a gene, g_a, and an l-segment, $T_l = <t_{i_1}, t_{i_2}, \cdots, t_{i_{l+1}}>$, the sequence generated in such a way that orderly connects all results of $\mathcal{O}(t_{i_k}, t_{i_{k+1}}, t_{i_{k+2}})$ for k=1 to l-1 is called the gCode of gene g_a on the l-segment T_l, denoted $gCode(g_a, T_l)$, where $\mathcal{O}(<t_{i_j}, t_{i_k}, t_{i_l}>)=\mathcal{O}(\mathcal{R}eg(<t_{i_j}, t_{i_k}>), \mathcal{R}eg(<t_{i_k}, t_{i_l}>))$ and $\mathcal{O}$-operation has the following properties:*
(1) $\mathcal{O}(\nearrow, \nearrow) = 1$; $\mathcal{O}(\searrow, \searrow) = 1$;
(2) $\mathcal{O}(\nearrow, \searrow) = 0$; $\mathcal{O}(\searrow, \nearrow) = 0$;

Definition 4. *Synchronous Co-regulation. For two given genes, g_a and g_b, if there exists an l-segment, T_l=$<t_{i_1}, t_{i_2}, \cdots, t_{i_{l+1}}>$, such that $gCode(g_a, T_l)$=$gCode(g_b, T_l)$, then we say g_a and g_b to be synchronous co-regulated each other on T_l. Furthermore, if $\mathcal{R}eg(g_a, <t_{i_k}, t_{i_{k+1}}>)$=$\mathcal{R}eg(g_b, <t_{i_k}, t_{i_{k+1}}>)$, where $k \in [1, l]$, the synchronous co-regulation between g_a and g_b is activation. Otherwise, if $\mathcal{R}eg(g_a, <t_{i_k}, t_{i_{k+1}}>)$=$-\mathcal{R}eg(g_b, <t_{i_k}, t_{i_{k+1}}>)$, the synchronous co-regulation between g_a and g_b is inhibition.*

Definition 5. *Asynchronous Co-regulation. Given two genes, g_a and g_b, if exists two l-segments, T_l=$<t_{i_1}, t_{i_2}, \cdots, t_{i_{l+1}}>$ and T'_l=$<t'_{i_1}, t'_{i_2}, \cdots, t'_{i_{l+1}}>$, such that $gCode(g_a, T_l)$=$gCode(g_b, T'_l)$ and $t'_{i_1} - t_{i_1} = t'_{i_2} - t_{i_2} = \cdots = t'_{i_{l+1}} - t_{i_{l+1}} = d$, where $d(>0)$ is a constant time-lag, then we say g_a and g_b to be asynchronous co-regulated on T_l and T'_l. Further, if $\mathcal{R}eg(g_a, <t_{i_k}, t_{i_{k+1}}>)$=$\mathcal{R}eg(g_b, <t'_{i_k}, t'_{i_{k+1}}>)$, where $k \in [1, l]$, we say g_a asynchronous activation co-regulate g_b after d time-lags. Otherwise, if $\mathcal{R}eg(g_a, <t_{i_k}, t_{i_{k+1}}>)$=$-\mathcal{R}eg(g_b, <t'_{i_k}, t'_{i_{k+1}}>)$, we say g_a asynchronous inhibition co-regulate g_b after d time-lags.*

Definition 6. *Reg-Cluster. Let $C = \bigcup_{i=1}^{r} G_i \times T_i$, where G_i is a subset of genes ($G_i \subseteq G$), and T_i is a prototypal subsequence ($T_i \subseteq T$), then C is a reg-cluster if and only if: (1) $\forall T_i, T_j$, $1 \leq i \leq j \leq r$, $|T_i| = |T_j|$, and (2) $\forall g_a \in G_i$, $\forall g_b \in G_j$, $1 \leq i \leq j \leq r$, the condition $gCode(g_a, T_i) = gCode(g_b, T_j)$ holds, and $t_{j_1} - t_{i_1} = t_{j_2} - t_{i_2} = \cdots = t_{j_k} - t_{i_k}$, where suppose T_i=$<t_{i_1}, t_{i_2}, \cdots, t_{i_k}>$ and T_j=$<t_{j_1}, t_{j_2}, \cdots, t_{j_k}>$.*

Problem Statement. Given: (1) D, a microarray data matrix, (2) δ, a user-specified maximum regulation threshold, (3) min_t, a minimal number of time points, and (4) min_g, a minimal number of genes, the task of mining is to find all maximal reg-clusters that satisfy all the given thresholds.

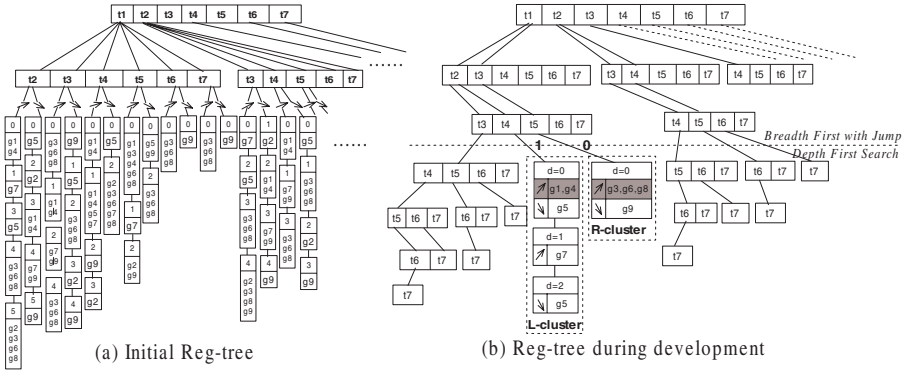
3 Algorithm

The $\mathcal{R}$eg-$\mathcal{C}$luster algorithm has two main steps: (1) Construct initial $\mathcal{R}$eg-tree. The preliminary reg-clusters on 1-segments are preserved in this step, (2) Develop initial $\mathcal{R}$eg-tree recursively to find all maximal reg-clusters. Unlike the previous algorithms, we take a "first breadth-first and last depth-first" searching strategy to make the algorithm more efficient while the pruning rules special for breadth-first and depth-first are applied respectively.

3.1 Construct Initial $\mathcal{R}$eg-Tree

We first look at the appearance of the initial $\mathcal{R}$eg-tree, and then describe how it is developed recursively.

Figure 2 (a) shows the initial $\mathcal{R}$eg-tree constructed from Table 1, which contains the reg-clusters on 1-segments according to Definition 6. There are two branches under each leaf node. One, with '↗', represents all genes under it are significantly up-regulated, and the other, with '↘', represents all genes under it are significantly down-regulated. Each reg-cluster $C = \bigcup_{i=1}^{r} G_i \times T_i$ is composed of a set of numbered buckets. We call the buckets with number '0' *baseline buckets* since the prototypal subsequence, T_1, of a baseline bucket is composed of the time points in the path from the root to the node that the reg-cluster C linked to. The number within each bucket denotes the time intervals that T_i is lagged behind T_1. For example, in Figure 2 (a), the leftmost reg-cluster under t_1t_3 is composed of five buckets. The prototypal subsequence of the baseline bucket, i.e. T_1, is $<t_1, t_3>$, and thus the prototypal subsequence of the second bucket is $<t_2, t_4>$ since the bucket's number is 1. Similarly, the prototypal subsequence of the third bucket is $<t_3, t_5>$, and so on.



(a) Initial Reg-tree (b) Reg-tree during development

Fig. 2. $\mathcal{R}$eg-tree

3.2 $\mathcal{R}$eg-Tree for 2-Segments

In this subsection, we construct a $\mathcal{R}$eg-tree with height 2 for all 2-segments, based on the initial $\mathcal{R}$eg-tree with height 1 for all 1-segments.

In a $\mathcal{R}$eg-tree with height 2, a 2-segment, T_2=$<t_i, t_j, t_k>$, is generated by concatenating $<t_i, t_j>$ and $<t_j, t_k>$ in the initial $\mathcal{R}$eg-tree for 1-segments. There are two clusters for every 2-segment, T_2, in a leaf node. For convenience, we denote the two clusters as L-cluster and R-cluster respectively. The L-cluster maintains all genes if their $<t_i, t_j>$ and $<t_j, t_k>$ have the same regulations (both up or both down), i.e. the gCodes of genes in the L-cluster are all 1. Similarly, the R-cluster maintains all genes if their $<t_i, t_j>$ and $<t_j, t_k>$ have different regulations (up/down or down/up), i.e. the gCodes of genes in the R-cluster are

all 0. It is important to know that all co-regulated genes are clustered into either L-cluster or R-cluster.

Next, we explain the $\mathcal{R}$eg-tree construction for 2-segments below using an example. Consider the $\mathcal{R}$eg-tree for 1-segments in Figure 2 (a). There are two baseline buckets for the 1-segment $<t_1, t_2>$. One, with '↗', contains $\{g_1, g_4\}$, and the other, with '↘', contains $\{g_5\}$. Also, there are two baseline buckets for the 1-segment $<t_2, t_4>$. One, with '↗', contains $\{g_1, g_4\}$, and the other, with '↘', contains $\{g_5\}$. In the $\mathcal{R}$eg-tree for 2-segments in Figure 2 (b), the 2-segment, $<t_1, t_2, t_4>$, has two clusters, L-cluster and R-cluster, each of which consists of a set of sub-clusters with different time-lag d. They are constructed as follows.

- Every sub-cluster with time-lag d within the L-cluster of $<t_1, t_2, t_4>$ is the union of all genes in the intersection of both ↗-cluster with time-lag d for $<t_1, t_2>$ and $<t_2, t_4>$ and the intersection of both ↘-cluster with time-lag d for $<t_1, t_2>$ and $<t_2, t_4>$. For example, the first sub-cluster, where d=0, is $(\{g_1, g_4\} \cap \{g_1, g_4\}) \cup (\{g_5\} \cap \{g_5\}) = \{g_1, g_4, g_5\}$, and so on.
- Every sub-cluster with time-lag d within the R-cluster of $<t_1, t_2, t_4>$ is the union of all genes in the intersection of ↗-cluster with time-lag d for $<t_1, t_2>$ and ↘-cluster with time-lag d for $<t_2, t_4>$ and the intersection of ↘-cluster with time-lag d for $<t_1, t_2>$ and ↗-cluster with time-lag d for $<t_2, t_4>$. For example, the first sub-cluster, where d=0, is $(\{g_1, g_4\} \cap \{g_5\}) \cup (\{g_5\} \cap \{g_1, g_4\}) = \varnothing$, and so on.

3.3 $\mathcal{R}$eg-Tree for l-Segments($l > 2$)

Hereafter, we start to develop the $\mathcal{R}$eg-tree recursively. Unlike the previous algorithms, we propose a "first breadth-first and last depth-first" searching strategy to make the reg-cluster algorithm more efficient. As its name implies, the development consists of two phases, i.e. the first phase, BFD("breadth-first development"), and the second phase, DFD("depth-first development").

In BFD phase, different from previous work [3], there is no need to grow $\mathcal{R}$eg-tree level by level until $\mathcal{R}$eg-tree is with height $min_t - 1$. We can skip several levels of $\mathcal{R}$eg-tree based on the following min_t-based jumping pruning rule.

Pruning Rule 1. $\min_t$-based jumping. *Given a k-segment* $<t_{i_1}, t_{i_2}, ..., t_{i_{k+1}}>$ *and an l-segment* $<t_{j_1}, t_{j_2}, ..., t_{j_{l+1}}>$, *we can directly obtain a* $MIN(min_t, (k + l))$*-segment, jumping over* $(k + 1)$*-segment*$\sim MIN(min_t, (k + l - 1))$*-segment, if and only if* $t_{i_{k+1}} = t_{j_{l+1}}$.

Once the height of $\mathcal{R}$eg-tree grows up to $min_t - 1$, the development switches to the next phase, DFD. DFD allows a special pruning:

Pruning Rule 2. *Given two pathes* X *and* Y *of a* $\mathcal{R}$*eg-tree, where* X *corresponds the segment* $<t_{i_1}, t_{i_2}, t_{i_m}>$ *and* Y *corresponds the segment* $<t_{j_1}, t_{j_2}, t_{j_n}>$. *If* $X \subseteq Y$ *and the reg-clusters under* X *is the same as those under* Y, *then the reg-clusters on* $<c_{i_1}, c_{i_2}, c_{i_m}>$ *are not maximal and all searches down the path* $t_{i_1}, t_{i_2}, t_{i_m}$ *can be pruned because they are guaranteed not to contain any maximal reg-cluster.*

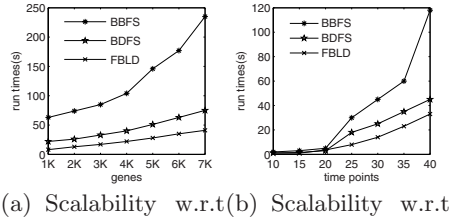
FBLD is a hybrid of BFD and DFD. It first develops $\mathcal{R}$eg-tree in a breadth-first way. Once the height of $\mathcal{R}$eg-tree grows up to $min_t - 1$, the development switches to the next phase, i.e. DFD. Pruning rule 1 and pruning rule 2 can be used in FBLD successively, so it outperforms single BFD or single DFD in performance. Limited by space, our complete pseudocode of FBLD is omit.

4 Experiments

We implemented our approaches in C++. For simplicity, the basic breadth-first approach is called BBFS, the basic depth-first approach is called BDFS, and the first breadth-first and last depth-first approach is called FBLD. Since *no other work to our best knowledge can discover the reg-clusters in the same manner*, we only compare the efficiency and the effectiveness of the three approaches on a 2.4-GHz DELL PC with 512 MB main memory running Windows XP. For the real dataset we use Spellman's yeast dataset(downloaded from http://genome-www.stanford.edu/cellcycle/data/rawdata/), which contains 6178 genes at 35 time points. The synthetic datasets can be obtained by a data generator algorithm [10] with three input parameters: number of genes (#gene), number of conditions (#sample), and number of embedded clusters (#cluster).

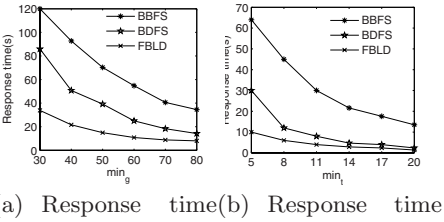
4.1 Efficiency

We first evaluate the performance of the three approaches, i.e. BBFS, BDFS and FBLD, on synthetic data sets as we increase the number of genes and the number of time points in the data sets. The average run times of the three algorithms are illustrated in Figure 3 respectively, where we vary the parameters invoked with min_g=30, min_t=5, and δ=0.01 .



(a) Scalability w.r.t # of genes (b) Scalability w.r.t # of times

Fig. 3. Evaluation of efficiency



(a) Response time vs. min_g (b) Response time vs. min_t

Fig. 4. Response time

Figure 3(a) shows the scalability for three approaches under different number of genes, when the number of time points is fixed to 6. Figure 3(b) shows the scalability for three approaches under different number of time points, when the number of genes is fixed to 30. FBLD cuts down the search space significantly, so it spends the least response time. BBFS need to decide which buckets(reg-clusters) can be joined with a given bucket during the development of $\mathcal{R}$eg-tree, however, BDFS need not. So BBFS will spend more time than BDFS.

Next, we study the impact of the parameters(min_g and min_t) towards the response time on the real datasets. The results are shown in Figure 4. As min_g and min_t increase, the response time shortened.

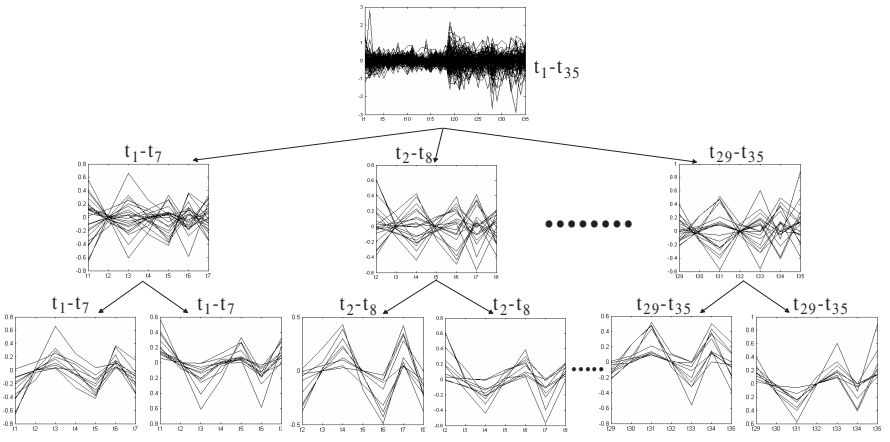


Fig. 5. One example of reg-cluster result

4.2 Effectiveness

Every reg-cluster can provide a further insight into the concrete relationships between co-regulated genes. Figure 5 delivers a hierarchical structure of the reg-cluster 18 discovered from Spellman's dataset. The root node summarizes all expression profiles of 179 genes on 35 time points, which corresponds to the whole reg-cluster 18. We first derive a set of sub-clusters, based on the time-lag d, at the second level to identify the synchronous/asynchronous co-regulation. Genes taken from the same sub-cluster of level 2 must synchronous co-regulate each other, either activation or inhibition. Genes taken from different sub-clusters of level 2 must asynchronous co-regulate each other, either activation or inhibition, and the accuracy value of time-lag can be inferred by the difference of starting time point of the two different sub-clusters. For more details, i.e. activation or inhibition, we drill down to level 3. Genes from the same sub-cluster must synchronous activation co-regulate each other; Genes from different sub-clusters but with the same parent node, must synchronous inhibition co-regulate each other; Genes from different sub-clusters and with different parent nodes must asynchronous co-regulate. If they are all the left(right) children, the relationship is activation. If some are the left children and some are the right children, the relationship is inhibition.

5 Conclusions

In this paper, we have proposed a new maximal subspace co-regulated gene clustering model, $\mathcal{R}$eg-$\mathcal{C}$luster, for simultaneously identifying all synchronous and

asynchronous co-regulations from time series gene expression data. Based on a proposed coding schema, i.e. gCode, genes with any of the known co-regulation relationships, i.e. synchronous activation, synchronous inhibition, asynchronous activation and asynchronous inhibition, are grouped together. A "first breadth-first and last depth-first" searching strategy with several useful pruning rules is also devised to make the maximal reg-clusters mining more efficient. Further, the detailed and complete co-regulation information, which facilitates the study of genetic regulatory networks, can be easily derived from the resulting clusters.

Acknowledgement

This work is supported by National Key Technologies Research and Development programming in the 10th Five-year(2004BA721A05) of P.R.China.

References

[1] H. Wang, W. Wang, J.Y., Yu, P.S.: Clustering by pattern similarity in large data sets. In: ACM SIGMOD Conference. (2002) 394–405

[2] J. Pei, X. Zhang, M.C.H.W., Yu, P.S.: Maple: A fast algorithm for maximal pattern-based clustering. In: Proc. of ICDM, Florida. (2003) 259–266

[3] Zhao, L., Zaki, M.J.: Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In: ACM SIGMOD Conference. (2005) 51–62

[4] X. Xin, Y. Lu, A.K., Wang, W.: Mining shifting-and-scaling co-regulation patterns on gene expression profiles. In: ICDE. (2006)

[5] Y. Zhao, G. Wang, Y.Y., Yu, G.: A novel approach to revealing positive and negative co-regulated genes. In: BIBE. (2006) 86–93

[6] J. Qian, M.D.F., Gerstein, M.: Beyond synexpression relationships: Local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions. In: JMB. (2001) 1053–1066

[7] H. Yu, N. Luscombe, J.Q.: Genomic analysis of gene expression relationships in transcriptional regulatory networks. In: Trends Genet. (2003) 422–427

[8] Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. of ISMB 2000 Conference. (2000) 99–103

[9] Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. In: IEEE/ACM Transactions on Computational Biology and Bioinformatics. (2004) 1 (1): 24–45

[10] D. Jiang, J.P., Zhang, A.: Interactive exploration of coherent patterns in time-series gene expression data. In: KDD. (2003) 565–570

A Parallel Algorithm for Learning Bayesian Networks

Kui Yu[1], Hao Wang[1], and Xindong Wu[1,2]

[1] Department of Computer Science and Technology, Hefei University of Technology, Hefei, Anhui 230009, China
ykui713@hotmail.com
[2] Department of Computer Science, University of Vermont, Burlington, VT 05405, USA
xwu@cems.uvm.edu

Abstract. Computing the expected statistics is the main bottleneck in learning Bayesian networks in large-scale problem domains. This paper presents a parallel learning algorithm, PL-SEM, for learning Bayesian networks, based on an existing structural EM algorithm (SEM). Since the computation of the expected statistics is in the parametric learning part of the SEM algorithm, PL-SEM exploits a parallel EM algorithm to compute the expected statistics. The parallel EM algorithm parallelizes the E-step and M-step. At the E-step, PL-SEM parallel computes the expected statistics of each sample; and at the M-step, with the conditional independence of Bayesian networks and the expected statistics computed at the E-step, PL-SEM exploits the decomposition property of the likelihood function under the completed data to parallel estimate each local likelihood function. PL-SEM effectively computes the expected statistics, and greatly reduces the time complexity of learning Bayesian networks.

Keywords: Bayesian networks, structural EM, parallel processing, MPI library, parallel EM.

1 Introduction

Bayesian networks[1] (BN) are a graphical representation for probability distributions. They are a popular framework in AI and uncertainty processing. Eliciting BN from domain experts can be a laborious and expensive process in large-scale applications, and sometimes it is simply not possible. Therefore, in recent years there has been a growing interest in learning BN from data.

A BN consists of a graph structure and a set of local probability distributions. Learning BN can be decomposed into two parts: discovering the graph structure and then the parameters for the graph structure. Current methods are effective in learning both the graph structure and parameters when data are complete, and can learn the parameters from incomplete data when the BN structure is known. However, learning BN structure from incomplete data is still a challenging problem.

Current techniques for learning BN are mostly based on a scoring approach which is characterized by devising a score metric for a candidate network structure and searching the space of network structures for the best-scoring structure. Most of the commonly used metrics can be decomposed into independent terms each of which

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1055–1063, 2007.

corresponds to one variable, such as the BIC, BDe or MDL metric[5]. When data are incomplete, we can no longer decompose the scoring function in BN construction. As a consequence, we are unable to perform local search for the network – in other words, a local change in one part of the network can affect the evaluation of a change in another part of the network. On the other hand, because some statistics are unknown, we cannot compute the scores of the network directly. There have been some methods proposed to solve those problems. Heckerman et al. presented some methods for the latter problem[2]. Those methods first use either EM (expectation-maximization) or gradient ascent (a gradient-based optimization) to compute the MAP parameters, then use either Laplace approximation or Bayesian information criterion (BIC)[3] to compute the approximate scores of the network using larger samples and approximation methods. Unfortunately, because of the large search space of the network and the errors produced by the approximate scores, the efficiency of learning BN is very low and the learned BN do not have enough confidence[4].

Friedman improved the methods presented by Heckerman et al. and proposed a structural EM (SEM) algorithm to learn BN from incomplete data[5]. The SEM algorithm consists of two parts: learning parameters and searching for the structure. In fact, the parametric learning part in SEM is a parametric EM algorithm that mainly computes the expected statistics of missing data. When searching for the BN structure, the SEM algorithm uses expected statistics instead of sufficient statistics that are unknown to make the scoring function have a closed form under certain assumptions. It can improve the expected score of the learned network at each iteration and make the structures converge at an optimal structure. Although able to improve the learning efficiency to some extent, SEM always stops at local optima.

Besides halting at local optima, Friedman also pointed out that the computation of the expected statistics is the main bottleneck in applying this technique to large-scale domains. Aiming at the problem of local optima, various researchers have presented several variants of SEM[6-8]. Unfortunately, those variants don't take into account the complexity of the time. When computing the expected statistics, we need to use an inference procedure of BN. But with the increase of the samples and missing data, the computation of the expected statistics is very huge[5]. Heckerman has proved that large-sample learning of Bayesian networks is NP-Hard[9]. Therefore, how to reduce the computation of the expected statistics is crucial for applying SEM or its variants to learning complex BN. So far there is little research on the problem.

Recently, parallel processing has become a useful technique for scaling up huge computations[10], and there has been some work on parallel learning BN[11-13]. Chu and Xiang presented a technique for using parallelism to speed up leaning decomposable Markov networks[11]. Lately, W. Lam et al. explored parallel algorithms for BN construction based on the K2 algorithm[12-13]. Although those algorithms can speed up learning BN, they have relied on the assumption that data are complete. This assumption is not very realistic, since most real world situations involve incomplete information. To the best of our knowledge, there is little work in the literature on using parallel learning algorithm for BN to deal with incomplete data.

In this paper, a parallel learning algorithm, called Parallel Learning using Structural EM (PL-SEM), is proposed to learn BN with incomplete data. PL-SEM adopts a parallel parametric EM algorithm to parallelize the parametric learning part of SEM. The parallel EM algorithm parallelizes the E-step and M-step of the SEM

algorithm to compute expected statistics. PL-SEM uses a parallel algorithm to compute the expected statistics and the parameters of the candidate network. It effectively computes the expected statistics, and greatly reduces the time complexity of learning BN.

The rest of the paper is organized as follows. Section 2 briefly reviews the framework for learning BN based on the EM algorithm. In Section 3, we present the parallel SEM algorithm for learning BN, called PL-SEM. Section 4 provides our experimental results and an analysis. Finally, Section 5 gives a summary.

2 Learning BN Structure Using EM Algorithm

Learning BN structure uses a training set $D=\{x_1, \ldots, x_n\}$and possible prior information to find a network that fits the database D as much as possible.

With the complete data, we can decompose the scoring function that evaluates the candidate network into a summation of terms, where each term consists of local family structures (a variable and its parents), and a local change in one part of the network doesn't affect the evaluation of a change in another part of the network — that is, the scoring function only needs to compute the scores of local structures that are changed.

For example, let BN=(G,θ) be a Bayesian network, and a finite set $U=\{X_i, 1\le i\le n\}$ be discrete random variables. Suppose $D=\{x_1, \ldots, x_n\}$ is a training set where each x_i has a value for some (or all) variables in U, and $N_X(x)$ is the number of instances in D. Note that $N_X(.)$ is well-defined only for complete datasets. Given a training data set D, we use the Bayesian Information Criterion (BIC)[3] to rank candidate network structures, using the BIC score of each candidate BN, written Score (BN : D), by the following equation:

$$\begin{aligned} Score\ (BN : D) &= \sum_{i=1}^{N} \log(P_{BN}(x_i)) - \frac{\log_2 N}{2} Dim\ [G] \\ &= \sum_i \sum_{x_i, \prod_{x_i}} N(x_i, \prod_{x_i}) \log(\theta_{x_i} | \prod_{x_i}) \\ &\quad - \frac{\log_2 N}{2} Dim\ [G] \end{aligned} \tag{1}$$

We can further decompose the Score (BN : D) as follows.

$$\begin{aligned} Score\ (BN : D) &= \sum_i Score_i(\prod_{x_i}, \theta_i : D) \\ &= \sum_i \sum_n (\log_2 \theta_i - \frac{\log_2 N}{2} Dim\ [i]) \end{aligned} \tag{2}$$

Where n is the size of the dataset and i is the node in the graph in Eq.(2). Dim[G] is the number of independent parameters in the graph G, and

$$\hat{\theta}_{x_i|\prod_{x_i}} = N(x_i, \prod_{x_i}) / N(\prod_{x_i}) \tag{3}$$

$$Dim\ [G] = \sum_i \sum_{p \in Pa(X_i)} (|X_p| \times |X_i| - 1) \tag{4}$$

When the data are incomplete, we can no longer decompose the likelihood function because some sufficient statistics $N_X(.)$ are unknown. Then we are no longer able to perform local search for the network - that is, a local change in one part of the network can affect the evaluation of a change in another part of the network. In order to be able to learn BN with incomplete data, Friedman presented an algorithm to learn BN structure from incomplete data based on a framework of EM, called the Structural EM algorithm (SEM). The basic idea of SEM is as follows.

Let $O \in U$ be a set of observable variables and H=U-O be a set of hidden variables. We assume that we have a class of models $G=\{M_0,\ldots,M_n\}$ such that each model $M \in G$ is parameterized by a vector θ_M where each (legal) choice of values of θ_M defines a probability distribution $P_r(.: M_h, \theta_M)$ over possible data sets, where M_h denotes the hypothesis that the underlying distribution is in the model M. From now on, we use θ^* as a shorthand for θ_M when the model M is clear from the context. $Pa(X_i)$ denotes the parents of X_i.

Given M^* and a set of observed data of O, to find the parameters θ^* of M^* is equivalent to find the choice of $(\theta^* : M^*, D)$ that maximizes the following scoring function.

$$\begin{aligned} Q(\theta^* : M^*, D) &= \underset{\theta}{Max} \sum_{X_i \in X} Q_{X_i}(\theta_i : M^*, X_i = x_i^j) \\ &= \underset{\theta}{Max} \sum_{X_i, Pa(X_i)} E[N(X_i, Pa(X_i)) \mid X_i = x_i^j)] \log_2(\theta_{X_i \mid Pa(X_i)}) \end{aligned} \tag{5}$$

Then with fixed θ^* and D, the SEM algorithm tries to find the choice of $(M: \theta^*, D)$ that maximizes the scoring function of the structure, i.e. searching for the space of the models for a model (or models) that maximizes Eq. (1) or Eq. (2). Therefore, the SEM algorithm consists of two steps: learning parameters and the structure.

Step 1 (referred to as the E-step). The SEM algorithm exploits the current model and parameters to compute expected statistics using a parametric EM algorithm, and then uses them to complete the incomplete data and re-evaluate the parameters of the current model.

Step 2 (referred to as the M-step). With the completed data, the scoring function has the decomposition property. The SEM algorithm performs a local change on the candidate network until it finds a network that most fits the completed data. Go to Step 1 until it stops at local optima.

According to the above-mentioned algorithm, it's at Step 1 that SEM exploits a parametric EM algorithm to learn the parameters of the current network. It's also the most expensive step because of the computation of the expected statistics. Fortunately, Step 2 is a local search procedure that changes one arc at each move and can efficiently evaluate the gains by adding or removing an arc. Such a procedure can also re-use the computations performed in the previous stages to evaluate changes to the parents of all variables that have not been changed in the last move. Therefore, how to reduce the computations of Step 1 is crucial for reducing the time complexity of SEM.

Based on the above observations, starting with a parallelized procedure for Step 1 to compute the expected statistics, our parallel SEM algorithm is proposed in this paper to reduce the time complexity of learning BN structure.

3 PL-SEM: A Parallel SEM Algorithm

At Step 1, most of the time of SEM algorithm is spent on choosing the optimal parameters for M_i using the EM algorithm. Therefore, PL-SEM uses a parallel parameter learning algorithm for SEM. PL-SEM reduces the computations in this step by parallel computing the expected statistics and the parameters of the underlying BN, so it improves the efficiency of learning BN.

3.1 Parallel Computing of the Expected Statistics

When learning the current network parameters using the EM algorithm with missing data or hidden variables, at the E-step, EM uses expected statistics instead of sufficient statistics to complete missing data. Computing the expected statistics needs to compute

$$\sum_{l=1}^{N} p(y_i, pa_j(y_i) \mid D_l, \theta^{(t)}) \tag{6}$$

where N is the number of samples and y is the missing data or hidden variables. A simple example is shown in Figure 1.

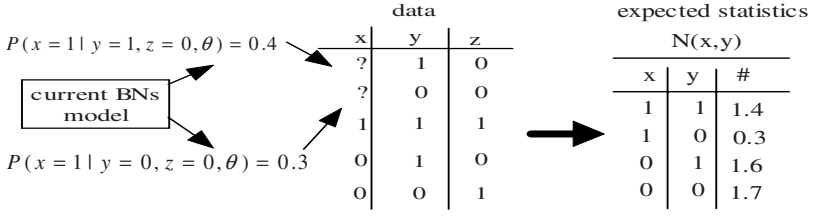


Fig. 1. Computation of expected statistics

According to Figure 1, we can conclude that the E-step consists of two components: 1) constructing a proper inference engine; and 2) computing the expected statistics.

If the parameters θ are the joint probability distribution of the network and are available for each processor, the same operation can be performed on each sample simultaneously. We parallelize the loop by evenly distributing the samples across parallel processors. If we partition the N samples into P blocks, each processor handles roughly N/P samples. The j'th processor is given a responsibility for samples N_i, where i=(j)(N/P)+1, …, (j+1)(N/P).

Data parallelization means that the same operation can be performed on different data items simultaneously. The E-step also repeats computing Eq.(6) on each sample

at the same time. Therefore, the E-step is inherently data parallel. From Figure 1, the computation of Eq.(6) for each sample is independent. Therefore, the E-step avoids the cost of communication between different processors and makes it up to maximal parallelism because of data parallelization. It is an important property of data parallelization that the data parallelism will arise with the increase of the scale of samples. Therefore, we can exploit more processors to improve the learning efficiency of PL-SEM.

3.2 Parallel Computing of BN Parameters

With complete data, it is easy to compute sufficient statistics at the E-step. Then the M-step uses the sufficient statistics to learn the MLE (Maximum Likelihood Estimation) parameters for the current network. With the complete data, the M-step can decompose the likelihood function L(θ:D) by exploiting the inherent conditional independence of BN. Let $X=\{X_1,X_2,\ldots X_n\}$be the random variables of BN, $x_i[m]$ be a instance of X_i in the m'th sample, $Pa_i[m]$ be the instances of the parents of X_i, and the parameters θ be ready to be evaluated, Then L(θ:D) is as follows.

$$\begin{aligned} L(\Theta : D) &= \prod_m P(x_1[m], \ldots, x_n[m] : \Theta) \\ &= \prod_m \prod_i P(x_i[m] \mid Pa_i[m] : \Theta_i) \\ &= \prod_i \prod_m P(x_i[m] \mid Pa_i[m] : \Theta_i) \\ &= \prod_i L_i(\Theta_i : D) \end{aligned} \tag{7}$$

Therefore the likelihood function is decomposed into n independent local likelihood functions. With the sufficient statistics, we can further decompose $L_i(\theta_i:D)$. Let $N(X_i,Pa_i)$ be the sufficient statistics of X_i.

$$\begin{aligned} L_i(\Theta_i : D) &= \prod_m P(x_i[m] \mid Pa_i[m] : \Theta_i) \\ &= \prod_{pa_i} \prod_{m, Pa_i[m]=pa_i} P(x_i[m] \mid pa_i : \Theta_i) \\ &= \prod_{pa_i} \prod_{x_i} P(x_i \mid pa_i : \Theta_i)^{N(x_i, pa_i)} = \prod_{pa_i} \prod_{x_i} \theta_{x_i \mid pa_i}^{N(x_i, pa_i)} \end{aligned} \tag{8}$$

Unfortunately, with incomplete data, the sufficient statistics can't be computed from samples directly, and the likelihood function L(θ:D) no longer has the property of decomposition.

But the good news is that the M-step can exploit the expected statistics computed at the E-step instead of the sufficient statistics to make the likelihood function have the decomposition property. Therefore, the M-step can also be parallelized.

Assuming θ_L and θ_G are the local and global parameters of the current BN model, respectively, in the training process, a processor P_0 first gets a random structure of the BN and learns parameters θ_G from it and the data D. Then, P_0 distributes these parameters to other available processors by using MPI_Bcast, which is one of the basic functions of the MPI (Message-Passing Interface) library[14] and all other nodes

must also call to receive the data. Next, each processor P_j uses the current global parameters θ_G to compute the expected statistics for its partition. Thirdly, each processor P_j needs to exchange its expected statistics with others using MPI_Bcast to compute the local parameters θ_L. At last, according to the number of random variables, we again assign the available processors to calculate each local parameter θ_L and call MPI_Allreduce[14] to sum up the local parameters θ_L to obtain the new global parameters θ_G. MPI_Allreduce is also a MPI function which combines values from all the other nodes to the root node and distribute the results back to all processors. Since each processor has the same global parameters θ_G, it can independently decide when it should exit the loop. Unfortunately, each processor needs to communicate with others at the M-step, and this leads to some time cost.

3.3 Outline of the PL-SEM Algorithm

According to the above analysis, the outline of PL-SEM is as follows.

Step 1. Input the training data D; i=1; initiate the structure of BN and θ_G at P_0; and distribute it to all available processors by using MPI_Bcast.

Step 2. Assign the samples to processor P_j.

Step 3. Parallel compute the optimal parameters of the current network.

Parallel E-step: according to θ_G, P_j first constructs the inference engine, then computes Eq.(6) for its partitions.

Parallel M-step: each processor P_j exchanges its expected statistics using MPI_Bcast. According to the number of random variables, PL-SEM assigns the available processors to calculate each local parameter θ_L and calls MPI_Allreduce to sum up the local parameters θ_L to obtain the new global parameters θ_G.

Step 4. With the completed data, PL-SEM performs a local change on the structure of the candidate network until it finds a network that most fits the data.

Step 5. i=i+1.

Step 6. If i<Maxitetaion or θ_G has not converged, goto step 3; otherwise return.

4 Experimental Results and an Analysis

We have developed our parallel source code on a high performance computing cluster that has the following configurations: (Xeon 3.0G(2M)*2, 1G DDR400*2, SCSI 73G)*9, which means that there is a master node and 17 computing nodes, Linux OS, OSCAR cluster management system and the cluster is connected by Gigabit Ethernet. The processors communicate with each other by using MPI.

We have chosen two Bayesian networks on the Web [15]: Asian and Alarm. Asian network is a popular Bayesian Network with 8 discrete nodes/valuables taking 2 values, which could be used to diagnose patients arriving at a chest clinic. The Alarm network was constructed from expert knowledge as a medical diagnostic alarm message system for patient monitoring. The domain has 37 discrete nodes/valuables taking between 2 and 4 values, connected by 46 directed arcs.

Fig.2 shows the execution time of PL-SEM for the Asia network with 10% and 30% missing data and 1000, 2000 and 3000 samples, respectively. From this figure, we can see that the execution time of PL-SEM correlates with the missing rate, and

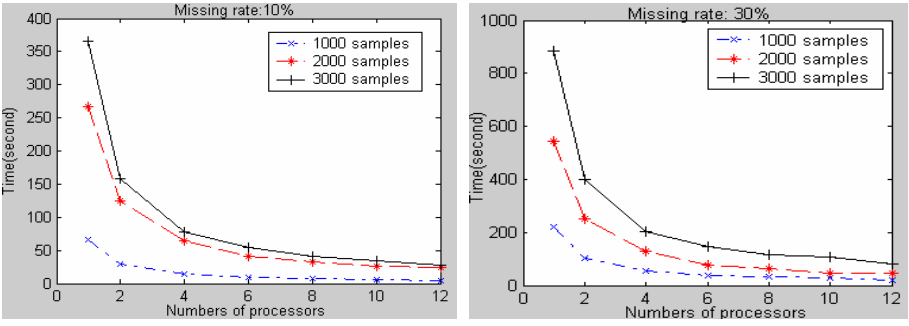


Fig. 2. The execution time for the PL-SEM algorithm with missing rates 10% and 30%

that the increase of the samples also augments the complexity of time. But with the increase of the processors, especially the number of the processors up to 12, the difference in the execution time is minimal between 1000 and 3000 samples when the missing date are 30%.

Fig.3 depicts the execution time of PL-SEM on the Alarm network which is far more complex than the Asia network. With 10% missing data and 1000 samples, the execution time at one processor is more than 1800 seconds. With the increase of processors, especially when the number of the processors is up to 16, the execution time is reduced to around 100 seconds. With 5000 samples and 20 % missing data, the time reaches 58080 seconds (about 16 hours). Fortunately, PL-SEM reduces it to 2000 seconds or so (about 0.6 hours) and the speed-up also arrives at 20 when the number of processors is up to 16. Therefore, PL-SEM effectively reduces the execution time. (Note that all missing data of the samples are produced at random.)

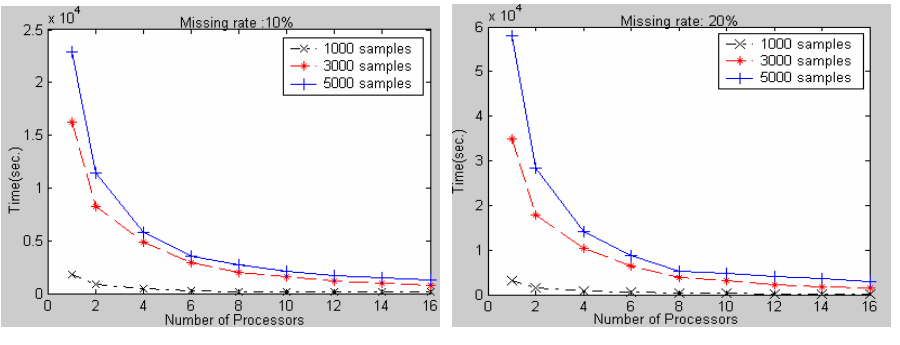


Fig. 3. The execution time of the PL-SEM algorithm with 10% and 20% missing data

Due to data parallelization at the E-step, PL-SEM effectively avoids the cost of communication with the increase of the processors. Unfortunately, at the M-step the processors need to communicate with each other. The time cost of communication will ascend with the increase of the processors. This affects the performance of PL-SEM to some extent.

5 Conclusion

Learning Bayesian networks with incomplete data is currently a hot research topic. Existing research efforts lay a heavy emphasis on how to avoid stopping at local optima. There is little research on improving the time complexity. With the increases of missing data and hidden variables, the computation is very huge. Therefore, many algorithms for learning Bayesian networks (with missing data) can't effectively learn from large-scale samples. In this paper, the PL-SEM algorithm has been presented based on the EM framework, and it exploits a parallel algorithm to reduce the computation of the expected statistics. PL-SEM does not take into account the problem of local optima. Fortunately, at Step 4, PL-SEM can choose another stochastic simulation such as the genetic algorithm, simulated annealing or MCMC (Monte Carlo Markov Chain) to perform local search to avoid local optima. Therefore, PL-SEM has provided a framework for parallel structure learning.

References

1. Ghahramani Z. An Introduction to Hidden Markov Models and Bayesian Networks. *IJPRAI* 15 (1): 9-42, 2001.
2. Chickering DM, Heckerman D. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Machine Learning*, 1997, 29(2-3): 181-221
3. Schwarz G. Estimating the dimension of a model. Ann. Stat., 6: 461-464, 1978.
4. Wang S-C, Yaun S-M. Research on Learning Bayesian Networks Structure with Missing Date, *Journal of Software*, 2004, Vol. 15(7), 1042-1048.
5. Friedman N. The Bayesian Structural EM Algorithm. *UAI-98*, 1998.
6. Tian F, Lu Y, Shi C. Learning Bayesian Networks with Hidden Variables Using the Combination of EM and Evolutionary Algorithms, *PAKDD 2001*, 568-574.
7. James W, Myers K, Laskey B et al. Learning Bayesian networks from incomplete data using evolutionary algorithms. *Proc of the 15th Conf on Uncertainty in Artificial Intelligence*. Stockholm, Sweden, 1999.
8. Luna JEO, Zanusso MB. Revisited EM Algorithms for Learning Structure and Parameters in Bayesian Networks. *IC-AI 2005*, 572-578.
9. Chickering DM, Heckerman D, Meek C. Large-Sample Learning of Bayesian Networks is NP-Hard. *Journal of Machine Learning Research* 5: 2004, 1287-1330.
10. Anderson TE, Culler DE, Patterson DA. A Case for NOW. *IEEE Micro*, vol. 15, no. 1, 54-64, 1995.
11. Chu T, Xiang Y. Exploring Parallelism in Learning Belief Networks, *Proc. of Conference on Uncertainty in Artificial Intelligence*, 90-98, 1997.
12. Lam W, Segre AM. A Distributed Learning Algorithm for Bayesian Inference Networks, *IEEE Transactions on Knowledge and Data Engineering*, 2002,Vol.14(1): 93-105
13. Munetomo F, Murao N, Akama K. Empirical studies on parallel network construction of Bayesian optimization algorithms, The IEEE Congress on Evolutionary Computation, 2-5 Sept. 2005, pp.1524 – 1531
14. Gropp W., Lusk E., Skjellum A. *Using MPI: portable parallel programming with the message-passing*. *The MIT Press, Cambridge*, MA, 1999.
15. Norsys Software Corp. http://www.norsys.com, 2006.

Incorporating Prior Domain Knowledge into a Kernel Based Feature Selection Algorithm

Ting Yu, Simeon J. Simoff, and Donald Stokes

The Faculty of Information Technology and the School of Accounting
University of Technology, Sydney, Broadway, NSW 2007, Australia
Capital Markets Cooperative Research Centre, Australia
{yuting, simeon}@it.uts.edu.au, donald@cmcrc.com

Abstract. This paper proposes a new method of incorporating prior domain knowledge into a kernel based feature selection algorithm. The proposed feature selection algorithm combines the Fast Correlation-Based Filter (FCBF) and the kernel methods in order to uncover an optimal subset of features for the support vector regression. In the proposed algorithm, the Kernel Canonical Correlation Analysis (KCCA) is employed as a measurement of mutual information between feature candidates. Domain knowledge in forms of constraints is used to guide the tuning of the KCCA. In the second experiments, the audit quality research carried by Yang Li and Donald Stokes [1] provides the domain knowledge, and the result extends the original subset of features.

1 Introduction

In the machine learning community, there is a popular belief that an increasing amount of features will enhance the performance of learning machineries, where the feature selection always reduces the information contained by the resultant models. Some research shows that irrelevant features do not increase the information, but introduce additional noise that eventually harms the performance of the resultant models. Peter Cheeseman suggests that all effects that have not been modelled add to the noise term [2]. Irrelevant features introduce noise into the learning process, also degrading the performance. Too many features cause the curse of dimensionality, which is always a negative result in machine learning. Simultaneously, the loss of strong relevant features degrades the performance of the resultant model too.

Many of existing feature selection algorithms emphasizes the discovery of the relevant features but ignore the elimination of redundant features. They suffer from quadratic, or even higher complexity about N, such that it is difficult to scale up high dimensionality. This paper proposes an approach to construct an optimal subset of features for a given machine learning algorithm. The optimal subset of features contains the majority of relevant information with less redundancy. Mark Hall defined feature selection as "successful if the dimensionality of the data is reduced and the accuracy of a learning algorithm improves or remains the same" [3]. Daphne Koller el at formally defined the purpose of feature selection: let μ and σ be two distributions over some probability space Ω. The cross-entropy of μ to σ is defined as

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1064–1071, 2007.

$D(\mu,\sigma) = \sum_{x\in\Omega}\mu(x)log\frac{\mu(x)}{\sigma(x)}$, and then $\delta_G(f) = D(Pr(C|f), Pr(C|f_G))$. The optimal subset is a feature set G for which $\Delta_G = \sum_f Pr(f)\delta_G(f)$ is reasonably small [4]. It is quite difficult to measure the difference, Δ_G, especially in case of continuous data. Thus in practice some alternative ways to measure the difference Δ_G are required to define the optimal subset.

The second section introduces some basic concepts of feature relevance and feature redundancy. The third section introduces the method of mutual information measurements and ML and MP constraints. The fourth section presents the algorithm and the fifth section follows by experiments. The sixth section concludes the whole paper.

2 Feature Relevance and Feature Redundancy

Before discussing the proposed method, it is necessary to define the related concepts. Considering supervised learning, the task of the induction algorithm is to induce a structure (a decision tree or SVM) such that, given a new instance, it is possible to accurately predict the target Y. George John et al defined two concepts about relevance: *Strong Relevance*: A feature X_i is relevant iff there exists some x_i, y and s_i for which $p(Y = y|X_i = x_i, S_i = s_i) \neq p(Y = y|S_i = s_i)$. *Weak Relevance*: A feature X_i is weakly relevant iff it is not strongly relevant, and there exists a subset of features S_i' of S_i for which there exists some x_i, y, s_i' with $p(Y = y|S_i' = s_i') > 0$ such that $p(Y = y|X_i = x_i, S_i' = s_i') \neq p(Y = y|S_i' = s_i')$ [5]. The weak relevance implies that the feature can *sometimes* contribute to prediction accuracy, but the strong relevance indicates that the feature is crucial and not replaceable with respect to the given task. It is obvious that an optimal subset of features must include strong relevant features. But in term of weak relevant features, so far there is no principle indicating which weak relevant features are included.

Thus in order to extract the optimal subset of features, it is necessary to introduce two other concepts: Markov blanket and redundant features. Given a feature $F_i \in F$, let $M \subset F$ be a set of features that does not contain F_i. We say that a set of features M is a *Markov blanket* for F_i if and only if F_i is conditionally independent of a subset of features that does not contain the M and feature F_i, $F - M - F_i$, given M, $P(F - M - F_i|F_i, M_i) = P(F - M - F_i|M_i)$ [4]. If M is a Markov Blanket of F_i, denoted as $MB(F_i) = M$, then it is also the case that class C is conditionally independent of the feature F_i given M: $p(Y = y|M = m, F_i = f_i) = p(Y = y|M = m)$, that is $\Delta_M = \Delta_{M+F_i}$. *Redundant feature*: Let G be the current set of features, a feature is redundant and hence should be removed from G iff it is weakly relevant and has a Markov Blanket M_i within G [5].

It is worthwhile highlighting that the optimal set of features is approximately equivalent to the Markov Blanket, which contains the majority of the direct or most influential features with respect to the target y. Consider two very simple examples, where the two structures seem very similar only from the data set (see Figure 1). In the figure 1a, two strong relevant features directly impact the target with the probability $P(T|f_1) = 0.12$ and $P(T|f_2) = 0.6$ respectively. In the figure 1b, the two features are weak relevance and can replace each other. It is clear that the feature f_1 impacts the target T through the feature f_2. But if mutual information between the

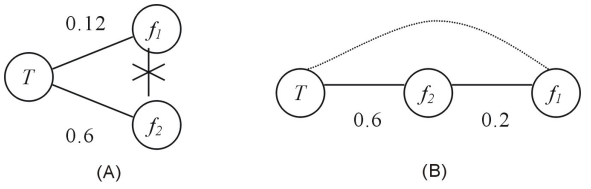


Fig. 1. $\{f_1, f_2\}$ are relevant features, but the structures of influence are different. T stands for the target (or dependent variable) and $\{f_1, f_2\}$ is a set consisting of two features.

target T and $\{f_1, f_2\}$ is measured, it is exactly the same as the previous example: $P(T|f_1) = P(f_2|f_1)P(T|f_2) = 0.2 \cdot 0.6 = 0.12$. Without considering the interrelationship between two features, it is impossible to distinguish these two graphs, so that the resultant subset of features will be the same, $\{f_2\}$. It is not optimal in the first case where the information loss occurs.

Classical backwards feature selection, such as Recursive Feature Elimination (RFE) proposed by Guyon et al, implicitly removes the redundant features, and might not uncover the optimal set [6]. For example, a backward feature selection with a high threshold, say greater than 0.12, does not construct the optimal subset of features by excluding the feature f_1 in the first case (Figure 1a). But in the second case, it functions well. Contrary to the classical backwards feature selection, if the feature selection is based on these two graphs, the resultant optimal subsets are explicitly correct. At the same time, the backwards feature selection is very time-consuming. Some researchers suggests that for computational reasons it is more efficient to remove several features at a time at the expense of possible classification performance degradation [6]. Therefore another issue rises: how does one partition features and reduce the negative impacts on the classification performance.

3 Mutual Information Measurement

In order to eliminate redundant features from the candidates, it is important to measure the mutual information between a pair of features. In classical statistics and information theory, the correlation coefficient and cross entropy are two often-used measurements of the influence between features. However, in some practical application, especially the continuous data set, these two methods require discretisation as a pre-process. This is due to the fact that the cross entropy only works in case of discrete data set. The quality of the discretisation relies heavily on a users' setting as during the process, and the important information might be lost. At the same time, regular correlation analysis only measures linear correlation coefficients. Therefore both of them are unable to measure the nonlinear correlation coefficients within the continuous data sets. The core idea of the proposed method is to convert a nonlinear problem into a linear problem via kernel mapping. Within the resultant feature space, the regular canonical correlation is employed to measure the impact between mapped features. As a result, canonical correlation is kernalized and extended into a nonlinear problem.

The kernalization of canonical correlation is not a completely new idea, and some unsupervised kernel methods already implement it. Canonical correlation analysis is

a multivariate extension of correlation analysis, and is employed by the Independent Component Analysis (ICA) as a contrast function to measure the independence between resultant latent variables. Beyond the linear Canonical Correlation Analysis (CCA), the kernelized version of CCA works in a feature space. It utilizes extra information from a higher order element of moment function than the second order in the linear canonical correlation. The contrast function used in the kernel ICA developed by Bach and Jordan is employed as an approximation of mutual information between two features (or variables) [7]. The mutual information between two features can be written as: $I(x_1, x_2) = -\frac{1}{2}\sum_{i=1}^{p} log(1-\rho_i^2)$, where ρ_i are the canonical correlations, and this method is employed in the Kernel-based ICA algorithms.

3.1 ML and MP Constraints

It is important to highlight that the Kernel CCA (KCCA) is an approximation to the real mutual information between features. The accuracy of the approximation relies on the users' tuning, as with other kernel methods. To some degree, domain knowledge such as known relevant features, usefully helps to tune and reduce the approximation error. In this part, known related features in the formats of constraints are introduced into the training process of the Kernel CCA, such as a grid search, to get optimal values of parameters. In some domains, a significant amount of relevant features are known. For example, in the research of audit quality, a linear formula is available and provides sets of features, clearly demonstrating their influences to the target.

Ian Davidson proposed must-link as a constraint for clustering. The Must-Link (ML) requires two instances to be part of the same cluster [8]. In this feature selection problem, the must-link constraints are slightly different from the concept defined by Davidson. The new ML requires that the results from the KCCA must show the higher correlation between known feature and the target than a given number: $ML : \forall F_i \in S_{known}, Corr(F_i, T) > \mu$. For example, the below equation has been known and verified by the current research of audit quality: $LnAF = \beta_1 + \beta_2 LnTA + \beta_3 LnSub + \beta_4 DE + \beta_5 Quick + \beta_6 Foreign + \beta_7 CATA + \beta_8 ROI + \beta_9 Loss + \beta_{10} Opinion + \beta_{11} YE + \beta_{12} Intangible + \beta_{13} InverseMillsRatios + e$ [1], and then the ML can be defined as $\forall F_i \in \{LnTA, LnSub, \ldots, InverseMillsRatios\}$, $(Corr(F_i, LnAF) > \mu)$.

Another constraint represents the order of influence with each known relevant feature to the target. This is the Must-Precede (MP): $\{F_i, F_j\} \subseteq S_{known}$,$(Corr(F_i, T) \geq Corr(F_j, T)) \Rightarrow MP : \{F_i, F_j\}$. The feature F_i must precede F_j, if the correlation between F_i and the target is larger than the correlation between F_j and the target. Considering the previous example, if the correlation between the feature $LnTA$ and the target $LnAF$ is bigger than the correlation between the feature $LnSub$ and the target $LnAF$: $\{LnTA, LnSub\} \subseteq S_{known}$,$(Corr(LnTA, LnAF) \geq Corr(LnSub, LnAF)) \Rightarrow MP : \{LnTA, LnSub\}$.

In this proposed feature selection method, the domain knowledge in the format of ML and MP constraints guides the tuning process of the KCCA to measure the nonlinear correlation coefficient between features. By overcoming the difficulty of measuring the nonlinear correlation coefficients, one gains more insight into the interrelationship between features.

4 Algorithms

The proposed algorithm assembles the previous discussion and consists of three major steps. The first step is a grid search which repeats the KCCA between feature candidates until the results are consistent with the ML and MP constraints. The second step consists of a feature ranking by using a Recursive Feature Elimination (RFE) SVM proposed by Guyon et al [6]. The RFE is a kernel-based backwards feature selection method. With every iteration, it eliminates one or multiple features, testing the impact of elimination on the model coefficients learned by the SVM. If the impact of one removal feature is minimal among the candidates, it has the least influence in the resultant model. The output of the RFE is a list of ordered features. The order is determined by the influence of the features. For example, in the descending list, the most influent feature is the first one in the list. The third step follows Lei Yu and Huan Liu's Fast Correlation-Based Filter (FCBF) to remove the redundant features based on the results from the previous two steps [9]. In the case of a descending list, a search starts from the beginning of the list. Suppose a feature F_j precedes another feature F_i; if the correlation coefficient $SU_{i,j}$ between F_j and F_i is greater than the correlation coefficient $R_{i,c}$ between F_i and the target T, the feature F_i is removed from the list (see Figure 2). The search carries on until the end of the list, the result being an optimal subset of features. The pseudo code of the algorithm is outlined below:

Algorithm 1. Feature Selection Algorithm

procedure FEATURESELECTION($S(F_1, F_2, ..., F_N, T), ML, MP, \delta$)
▷ S is a training data set; δ is a predefined threshold; ML: the set of must-link constraints; MP: the set of must-precede constraint.
  S'_{list}=FeatureRanking(S)
  while $(\forall SU_i \in SU) SU_i \nvdash \{ML, MP\}$ **do**
    SU= GridSearch(KCCA(S, S_{known}))
  end while
  F_j = getFirstElement(S'_{list})
  while $F_j \neq NULL$ **do**
    F_i = getNextElement(S'_{list}, F_j)
    while $F_i \neq NULL$ **do**
      if $SU_{i,j} > R_{i,c}$ **then**
        remove F_i from S'_{list}
      end if
      F_i=getNextElement(S'_{list}, F_i)
    end while
    F_j=getNextElement(S'_{list} ,F_j)
  end while
  S_{best}=S'_{list}
  return S_{best}
end procedure

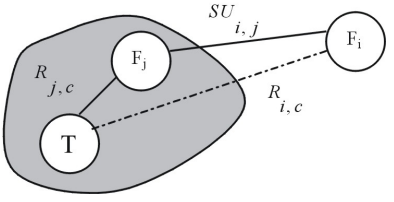


Fig. 2. The measurements in the FCBF algorithm proposed by Lei Yu and Huan Liu [9]

In this algorithm, prior knowledge in the format of known relevant features is represented as two types of constraints: 1) a set of must-link constraints $ML(F_i, T)$ between the target T and each of known relevant features $F_i \in S_{known}$, and 2) a set of must-precede constraints $MP(F_i, F_j)$ between known relevant features, where $Corr(F_i, T) \geq Corr(F_j, T)$ and $F_i, F_j \in S_{known}$. These constraints play a crucial role in directly determining the accuracy of the measurement of KCCA. For example, if the RBF is employed as the kernel functions, the grid search aims to detect the optimal value of the coefficient γ in the RBF $k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$. The coefficient γ determines mapping between the input space and the feature space.

5 Experiments

First, the author modifies a data generation provided by Leo Breiman [10] to produce an artificial data set. The artificial data set consists of 30 features and 60 pairs of observations. The modification is: within ten features, assign the 0.2 time of the value of fifth feature to the forth feature, $F_4 = 0.2F_5$, $F_{14} = 0.2F_{15}$ and $F_{24} = 0.2F_{25}$. Thus, among the thirty features of this artificial data set, there are six weak relevant features, three strong relevant features and twenty-one irrelevant features. The index of weak relevant features is 4, 14, 24, 5, 15, 25 and the index of strong relevant features is 6, 16, 26. The Recursive Feature Elimination (RFE) SVM produces a list of ranked features and the values of R-square while eliminating one feature every iteration (See Figure 3). The resultant list of ranked features is $\{2, 1, 18, 30, 29, 13, 20, 19, 11, 12, 8, 23, 21, 7, 10, 3, 9, 22, 17, 6, 14, 4, 24, 28, 27, 26, 5, 16, 25, 15\}$. In the Figure 3, at the flat top part of the curve, the weak relevant features and strong relevant features mix with a few irrelevant features. It is difficult for the backward sequential feature selection to discover the redundant features from the weak redundant features and then uncover the optimal subset of features. By taking account of the mutual information between feature candidates, the proposed feature selection discovers the strong correlation between feature candidates. At the same time, the correlation between the 5th feature and the target is larger than the correlation between the 4th feature and the target. Therefore the optimal subset of feature produced by the proposed feature selection algorithm includes the features $\{6, 28, 27, 26, 5, 16, 25, 15\}$. The mean R-square of 10 times cross-validation of a SVR with the same hyper-parameter as the previous SVRs is 0.689527. That is close to the best R-square value of the previous backwards sequential feature selection.

Secondly, a real-world data set is employed to test the proposed feature selection algorithm. The data set is collected from the auditing and accounting reports from listed companies in the Australian Stock Exchange (ASX) in 2003. To ensure an appropriate data set for the experiments, we first need to exam whether the data set contains both relevant and redundant features. The RFE SVM produces an ascending list of 39 ranked features (more important features are close to the end): {35, 30, 29, 34, 33, 32, 17, 1, 18, 4, 8, 31, 14, 36, 37, 10, 13, 15, 3, 25, 12, 28, 24, 26, 22, 27, 16, 20, 21,11, 2, 23, 38, 7, 9, 6, 19, 5, 39}. The same as previous experiment, Figure 3 demonstrates the test result of nonlinear Radial Basis Function Support Vector Regression (RBF SVR), with each iteration removing one feature. The order of feature removal follows the order of feature-ranking list. The flat middle part of the curve indicates that the eliminated features do not have strong impacts on the test accuracy of the resultant model, and this result indicates the existence of redundant features.

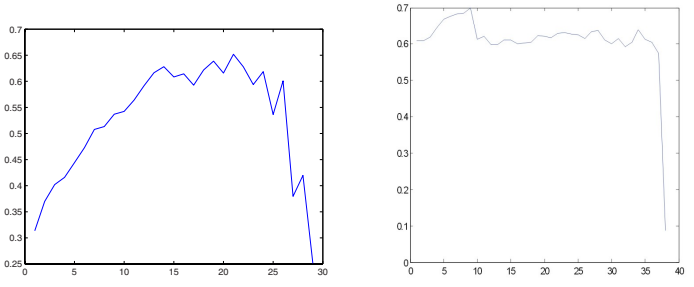


Fig. 3. The test accuracy of models constructed by a SVR while eliminating one feature every iteration. The x-axis is the index of feature in the ascending list, and the y-axis is the R-square. The first figure is for the first experiment, and the second figure is for the second experiment.

According to the proposed algorithm, the Kernel CCA is employed to measure the correlation coefficients between features. If the correlation coefficient between features is greater than 0.05 ($KCCA > 0.05$), the coefficient is retained in the final KCCA table. Otherwise, this experiment sets two of the corresponding features as irrelevant. In the final KCCA table, the index of the features without strong correlations to the known relevant features is (10, 14, 25, 29-32, 36-37). Based on these results, the proposed algorithm generates an optimal subset of features consisting of 16 features: {Loss Indicator (1), YE (8), OFOA (31), Current Liabilities (14), 12 Months (36), Currency (37), Current Assets (10), DE (3), QUICK (2), AF/EBIT (38), CATA (7), FOREIGN (9), LnSUB(6), Partner Code (19), LnTA (5), AF/TA (39)}. Using the SVR with 100 random sampling 80% of the original data set as the training data, the average test result (R-square) is 0.8033 with the standard deviation 0.0398. This result is slightly higher than 0.774, the best test result (R-square) using the SVR with the same hype-parameters but the subset of features produced by the backwards feature sequential selection.

6 Conclusion

This research expands Lei Yu and Huan Liu's Fast Correlation-Based Filter (FCBF) [9]. It does so primarily in that it implements the redundancy measurement in feature selection for the non-linear regression. As domain knowledge, known relevant features are included in the process to guide the process of the KCCA, under the condition that a sufficient amount of known relevant features is available. Considering the human involvement, it is worth ensuring whether the KCCA produces an appropriate approximation to the real mutual information. In this research, however, domain knowledge from experts is utilized to guide tuning of the parameters of selected kernels.

The results of these experiments show that the optimal set of features increases the accuracy to a relatively high level with relatively small optimal subset of features. Similar idea can be found in Carlos Soares's meta-learning methods to select kernel width in SVR [11]. Their meta-learning methodology exploits information about past experiments to set the width of the Gaussian kernel. In the feature selection experiment, domain knowledge collected from domain experts' past experiments is included to set the width of the kernel. At this stage, the result still relies heavily on the given domain knowledge with the assumption that the given domain knowledge is perfect. The most pertinent area for further investigation is the negative impacts of given domain knowledge on the resultant subset of features. It is still not clear that known related features will become redundancy when other features are included. The desired optimal set of features may contain all or a subset of known features. It is still an open question.

References

1. Li, Y., Stokes, D., Hamilton, J.: Listed company auditor self-selection bias and audit fee premiums. (2005)
2. Cheeseman, P., Stutz, J.: Bayesian classification (autoclass): Theory and results. In: Advances in Knowledge Discovery and Data Mining. (1996) 153–180
3. Hall, M.A.: Correlation-based Feature Selection for Machine Learning. Phd thesis, Waikato University (1999)
4. Koller, D., Sahami, M.: Toward optimal feature selection. In: the Thirteenth International Conference on Machine Learning, Morgan Kaufmann (1996) 284–292
5. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: International Conference on Machine Learning. (1994) 121–129
6. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning **46**(1-3) (2002) 389 – 422
7. Bach, F.R., Jordan, M.I.: Kernel independent component analysis. Journal of Machine Learning Research **3** (2002) 1–48
8. Davidson, I., Ravi, S.S.: Hierarchical clustering with constraints: Theory and practice. In: the 9th European Principles and Practice of KDD, PKDD. (2005)
9. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research **5** (2004) 1205–1224
10. Breiman, L.: Heuristics of instability and stabilization in model selection. The Annals of Statistics **24**(6) (1996) 2350–2383
11. Soares, C., Brazdil, P.B., Kuba, P.: A meta-learning methods to select the kernel width in support vector regression. Machine Learning **54** (2004) 195–209

Geo-spatial Clustering with Non-spatial Attributes and Geographic Non-overlapping Constraint: A Penalized Spatial Distance Measure

Bin Zhang, Wen Jun Yin, Ming Xie, and Jin Dong

IBM China Research Lab, Beijing, China
{zbin,yinwenj,xieming,dongjin}@cn.ibm.com

Abstract. In many geography-related problems, clustering technologies are widely required to identify significant areas containing spatial objects, particularly, the object with non-spatial attributes. At most of times, the resultant geographic areas should satisfy the geographic non-overlapping constraint. That is, the areas should not be overlapped with other areas. If without non-spatial attributes, most spatial clustering approaches can obtain such results. But in the presence of non-spatial attributes, many clustering methods can not guarantee this condition, since the clustering results may be dominated in non-spatial attribute domain which can not reflect the geographic constraint. In this paper, a new spatial distance measure called penalized spatial distance (PSD) is presented, and it is proofed to satisfy the condition which can guarantee the constraint. PSD achieves this by well adjusting the spatial distance between two points according to the non-spatial attribute values between them. The clustering effectiveness of PSD incorporated with CLARANS is evaluated on both artificial data sets and a real banking analysis case. It demonstrates that PSD can effectively discover the non-spatial knowledge and contribute more reasonably to spatial clustering problem solving.

1 Introduction

Spatial clustering technologies are widely used in geography-related analysis to identify significant areas for business decision making, especially in advanced analysis based on Geographic Information Systems (GIS). Beyond those traditional spatial clustering problems of only considering object positions [1,2,5,7], many extended algorithms [4,9] have been proposed to deal with the non-spatial attributes. Meanwhile, several algorithms [3,6,10] were improved in order to handle the clustering problem in the presence of obstacles, which can be a kind of geographic constraint.

In many applications, clustering methods are utilized to discover the geographic areas containing spatial points. We often have to face one kind of geographic constraint: the discovered geographic areas should not be overlapped and each area should be connected. If an area is not connected, it must be truncated

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1072–1079, 2007.

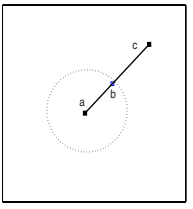


Fig. 1. Condition to avoid overlapping

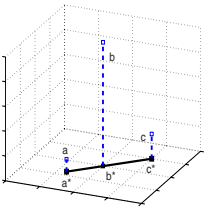


Fig. 2. Euclid distance with non-spatial attribute

by other areas. This case also can be treated as that this joint area is overlapped with other areas. (In the following paragraphs, we will not mention that an area should be connected.) For example, Given the household distribution with their income in a district, the task is to partition the area into two sub areas: one mainly contains low income people and the other contains high income people, so that each sub area can take different policies. In this case, it requires that the two sub areas should not be overlapped. If else, the mutual area will not be clear to make decision.

Many approaches have been proposed for the spatial clustering problem with non-spatial attributes, but they may not keep the geographic non-overlapping constraint. In [4], the approach applies CLARANS to spatial attributes and DBLEARN to non-spatial attributes separately. In [9], DBRS is designed to handle non-spatial attributes. Beyond using "density" for spatial attributes, it introduces "purity" and "purity-density-reachable" for non-spatial attributes. The mechanism that DBRS handles non-spatial attributes can be interpreted as a two-step process. At the first step, it splits the whole data set into several groups by their non-spatial attributes, and each group corresponds one value of non-spatial attribute. At the second step, it processes each group by normal DBRS approaches without non-spatial attribute and obtains final clusters. The difference in this step is that the new density (the density of data points with same non-spatial attributes) threshold is dynamically decided by local density (the density of all data points). Both above methods can be treated as two-step approaches, which process the non-spatial attributes in non-spatial attributes domain and spatial attributes in spatial attributes domain. The processing in non-spatial attributes domain can not consider the geographic constraints which should be reflected in spatial domain. Thus, the above approaches may obtain overlapped cluster areas. Another direct clustering method is to consider non-spatial attributes as extra dimensions of data points, and leverage a general multi-dimension clustering algorithm (using Euclid or other common distance measures). It treats the spatial attributes and non-spatial attributes homogenously, and brings the similar problem that the data points may be dominant in non-spatial attribute domain and loses geographic constraint in spatial domain.

In this paper, a new spatial distance measure is proposed to work with common spatial clustering algorithms (e.g. CLARANS) to solve the above geo-spatial

clustering problem with non-spatial attributes and geographic non-overlapping constraint. It properly adjusts (or penalizes) the distance of two spatial points by the differences of their non-spatial attributes, and it can be proved that this distance working with CLARANS can satisfy the non-overlapping constraint.

This paper is organized as follows. Section 2 illustrates the motivation and core idea of our approach. Section 3 describes the detailed implementation of our distance measure. In section 4, the distance measure is embedded with CLARANS to deal with both testing data sets and real banking data. The final section 5 concludes our work.

2 Motivation

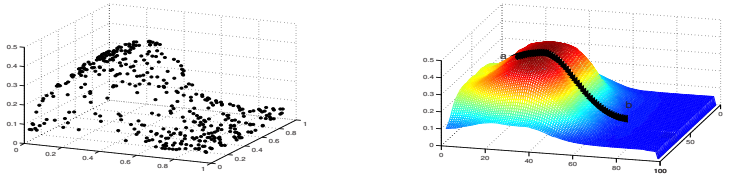
Pure spatial clustering approaches (without non-spatial attributes) seldom obtain such overlapped clusters, where Euclid distance (ED) is usually used to measure the distances of spatial points, and ED obviously has the following character:

– For every triple of points $\{a, b, c\}$ on one line in **spatial** domain, if b is in the middle of ac, then $D(a, b) < D(a, c)$.

If the above condition can be satisfied, the resultant clusters will not be overlapped. As shown in Figure 1, a is the medoid of a cluster A. Since $D(a, b) < D(a, c)$, if c belongs to cluster A, then b will be assigned to cluster A. Thus, the points in the middle of a and c will be in the same cluster of A, so ac will not be truncated by other clusters. That is, cluster A will not be overlapped with other cluster in the direction ac. Because the condition is for every triple of points on one line, so A will not be overlapped in any direction.

Considering the non-spatial attribute as the third dimension and using 3D Euclid distance, the above condition will not be kept if non-spatial attribute is dominant. As in Figure 2, a^*, b^*, c^* are three spatial points on one line, and a, b, c are their non-spatial attribute values respectively. b^* is in the middle of a^*c^*, but the 3D-Euclid distances are $D(a, b) > D(a, c)$. Thus, b^* may be in a different cluster from a^* and c^*. In geo-spatial plane, the cluster of a^* and c^* may be overlapped with the cluster of b^*.

The feasible distance measure should both satisfy the above condition and reflect the effect of non-spatial attributes. We present a new distance measure termed penalized spatial distance (PSD), which well adjusts the spatial distance between two points by the change of the non-spatial attribute values between them. If the non-spatial attribute values between two points change larger, the PSD will be penalized (increased) larger. If else, the distance will be penalized less. Our distance measure has the character: if a, b, and c are on one line in spatial domain, and b is in the middle of ac, then $D_{PSD}(ac) = D_{PSD}(ab) + D_{PSD}(bc)$. Because $D_{PSD}(bc) > 0$, $D_{PSD}(ac) > D_{PSD}(ab)$ can be always kept. So that this distance measure can satisfy the above condition.



(a) Discrete distribution of non-attribute values

(b) Surface from discrete non-attribute values

Fig. 3. Non-attribute value surface construction

3 Penalized Spatial Distance

Following the above idea, it is important to find the non-spatial attribute value series between two points. Since there is probably no point exactly on the line, a possible way is to construct the "surface" of non-spatial attribute values and extract the surface values on the line as the value series between the two points. Figure 3(b) is the constructed surface of 3(a), and curve ab is the series of non-spatial attribute values between spatial point a and b. The next task is to adjust the distance according to the non-spatial attribute value series. As the non-spatial attribute values in neighbor geographic area should not change too large because of the geographic continuity character, e.g. the income of people in a same area will not vary a lot, it is reasonable to construct this non-spatial value surface. The whole process consists of two steps below.

3.1 Construct Non-spatial Attribute Surface and Find Value Series Between Two Points

Many elaborate surface approximation approaches can be leveraged to reconstruct surface from points. Since it does not mean to compute an exact surface, but only adjust the distance between two points by the changing of non-spatial attribute values, we here choose the commonly used kernel based interpolation approach.

The grid length L decides how well the surface matches the discrete points. It should be selected to make a majority of grids have points in them. L could be chosen to satisfy: $\frac{\#\{P_i|d_i>L\}}{N} = k_L$. N is the number of points. k_L controls the number of points that can be assigned in separate grids, and can be empirically selected around $0.2 - 0.3$. d_i indicates the minimum distance from one point P_i to its k neighbor points. To be more resilient, it can be set as the mean distance from this point to its k nearest points.

The values of a part of mesh grids can be assigned directly as the mean values of the point values in them. For each unassigned grid (a, b), its value is set according to its nearest assigned grids: $S(a, b) = K_{(x,y)}(a - x, a - y)$, where $K_{(x,y)}$ is usually a RBF kernel, e.g. gaussian function. After the interpolation processing, a smoothing operation can be performed to reduce the noises.

Suppose the mesh surface is $S(\cdot,\cdot)$. Two points a and b are in grid $G_a(x_a, y_a)$ and $G_b(x_b, y_b)$ separately. We should find the grids P_{ab} on the direct path connecting G_a and G_b in 2D spatial plane. The value series of P_{ab} is: $V_{ab} = \{S(g_i)|g_i \in P_{ab}\}$. In Figure 3(b), the series of black triangle points are V_{ab}.

3.2 Adjust Spatial Distance by Value Series

Suppose the value series is $V_{ab} = \{S(g_0), S(g_1), S(g_2),\cdots, S(g_n)\} = \{s0, s1, s2,\cdots, s3\}$, where $s_i = S(g_i)$. g_0 is the start point, and g_n is the end point. Considering the i-th value and the $(i+1)$-th value, if the difference between s_{i+1} and s_0 (denoted as $s_{i+1} - s_0$) is larger than $s_i - s_0$, the distance will be penalized more. If else, the distance will be penalized less. By starting from the first value in V_{ab} and processing each value one by one, the total PSD can be formulated in Equation 1.

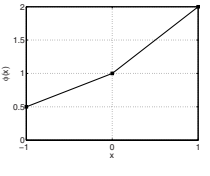
$$D_{PSD}(a,b) = \sum_{s_i \in V_{ab}} D_{geo}(g_{i+1}, g_i) * p(s_{i+1}, s_i) * k_d \tag{1}$$

where, k_d is a control factor, which describes how much the non-spatial attribute values influence the distance. Larger k_d will make larger effect of penalty. As g_{i+1} and g_i are neighbors, $D_{geo}(g_{i+1}, g_i)$ may be $1*L$ or $\sqrt{2}*L$. The penalizing factor $p(s_{i+1}, s_i)$ depends on s_{i+1} and s_i by Equation 2.

$$p(s_{i+1}, s_i) = \begin{cases} \phi\left(\frac{|s_{i+1}-s_0|-|s_i-s_0|}{T}\right), & \text{if } (s_{i+1}-s_0)(s_i-s_0) \geq 0 \\ \frac{s_{i+1}-s_0}{s_{i+1}-s_i}\phi\left(\frac{|s_{i+1}-s_0|}{T}\right) + \frac{s_0-s_i}{s_{i+1}-s_i}\phi\left(-\frac{|s_i-s_0|}{T}\right), & \text{if } (s_{i+1}-s_0)(s_i-s_0) < 0 \end{cases}$$
$$(i = 0, 1, 2, \ldots, n-1) \tag{2}$$

where, T is the maximum difference between the values of arbitrary two neighbor grids. The function $\phi(\cdot)$ is define as Fig. 4. As T is the maximum difference value, so the input range of ϕ is $[-1, 1]$. It is noted that $\phi(-1)$ is less than $\phi(1)$, which ensures that the PSD value of changing backward is less than that of changing forward. It is obvious that PSD is symmetrical according to Equation 1 and 2.

The above description can be directly extended to multiple non-spatial attributes. A surface should be constructed for each attribute. The spatial distance is adjusted by the total change of every non-spatial attribute. Suppose there are



$$\phi(x) = \begin{cases} 0.5x + 1, & -1 \leq x < 0 \\ x + 1, & 0 \leq x \leq 1 \end{cases}$$

Fig. 4. The figure and formula of $\phi(x)$

Q non-attributes and k_q is the weight for p-th non-attribute, Equation 1 can be extended as Equation 3.

$$D_{PSD}(a,b) = \sum_{g_i \in P_{ab}} \left\{ D_{geo}(g_{i+1}, g_i) * \sum_{q \in Q} \{k_q * p\left(S_q(g_{i+1}), S_q(g_i)\right)\} \right\} \quad (3)$$

4 Experiment Evaluation

In our experiments, the penalized spatial distance is embedded in CLARANS [2] to deal with the clustering of points with non-spatial attributes. We first compare the the penalized spatial distance with Euclid distance by an artificial data set, then we show the result of PSD based clustering in a real banking analysis case.

4.1 Penalized Spatial Distance vs. Euclid Distance

Three distance measures are compared in this test:

1. *PSD*: CLARANS with PSD.
2. *Euclid*2: Only spatial attributes (x and y) are considered regardless of non-spatial attribute. The distance measure is 2-dimension Euclid.
3. *Euclid*3: Spatial attributes and non-spatial attribute are treated homogenously. The distance measure is 3-dimension Euclid.

50 data sets are automatically generated, and each of them contains 500 points. The spatial positions of them are selected in a 1×1 square following uniform distribution. The non-spatial values are generated from three gaussian functions with their random center points $c_i, i = 1, 2, 3$. Each point v_j can be assigned with three values from the three kernel functions by $s_{(v_j,c_i)} = 10 * \exp\left\{-D^2_{\text{Euclid}}(v_j, c_i)/\sigma_i\right\}$, where σ_i is randomly generated in $[0.1, 0.3]$. The non-spatial value of this point v_j is the maximum one in $\{s_{(v_j,c_1)}, s_{(v_j,c_2)}, s_{(v_j,c_3)}\}$. Three CLARANS parameters: the cluster number k, the maximum number of neighbors examined and the number of local minima obtained are set as 3, 300, 150 respectively. According to the generation of non-spatial values, points can be assigned into three clusters (corresponding to the three kernels), which are considered as the reference clusters. The number of the mis-clustered points (denoted as E) is used to evaluate the clustering results. Smaller mis-clusterd point number means better clustering result. We can see PSD performs better than *Eulid*2 and *Euclid*3 in Figure 5. *Euclid*3 almost can not obtain well clustering results as the dominant non-spatial attributes. In theory, 2-dimension Euclid distance results can not reflect non-spatial attribute distribution. In some test cases, e.g. the 7th, 15th, and 47th test samples, *Euclid*2 works as well as PSD. The reason is that the non-spatial attribute distribution luckily meets the spatial distribution of points.

4.2 A Real Banking Analysis Case

In banking market study, it is often required to identify the areas with similar attributes to help further business analysis. Fig. 6(a) shows the residential points

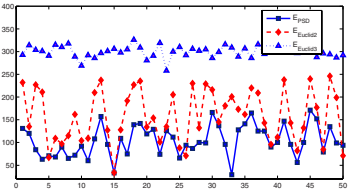


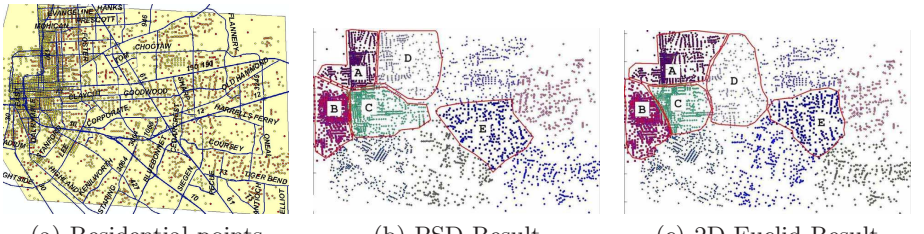
Fig. 5. Mis-clustered numbers by PSD, 2-D Euclid and 3-D Eculid

in *Baton Rouge, Louisiana, USA*. It contains about 2000 residential points with their population numbers, which are represented by gradual colors, where red color means high population. The task is to discover the areas which have similar population number. Assuming there are 10 clusters, we use CLARANS to cluster the points. Fig.6(b) and 6(c) show the clustering results by PSD and *Euclid*2.

The significant advantage is that PSD clustering can reflect a certain natural or geographic features. We select five significant differences between two clustering results, labeled by A to E. In Fig.6(b), Cluster A, B, C are properly separated by highway *110*. A and D are divided by a long straight driveway named *Foster*. While in the result by Euclid distance (Fig.6(c)), those features can not be exhibited. The cluster C in PSD result has a salient in its right part. In the map Figure 6(a), it is found that this salient area is extended from its left aggregated part (the major part of C) by several horizontal roads and streets, such as *Claycut*, and *Goodwood*. While the cluster C in Euclid result has not such characters. The upper boundary of Cluster E by PSD is actually formed by highway *190*, while Euclid distance can not character this.

4.3 Complexity

The complexity mainly depends on the clustering algorithm incorporated with PSD and the size of non-spatial attribute surface. As surface construction runs only one time, its time cost is not so consuming if we select the given interpolation algorithm or other efficient ones. If there are m grids from point a to b, PSD_{ab} can be finished in m iterations. Suppose the surface size is S, which mainly depends on the spatial distribution of points, and there are totally N points. The average "grid distance" of



(a) Residential points (b) PSD Result (c) 2D Euclid Result

Fig. 6. The residential points distribution in Baton Rouge, Louisiana, USA

each pair of points is approximately $\sqrt{2S/N}$. If we use the same clustering algorithm with both PSD and Euclid, the complexity of PSD is approximately $C = \sqrt{2S/N}$ times of that of Euclid. In our experiments, $S = 80 * 80 = 6400$, $N = 500$, so PSD's theoretically costs $C = \sqrt{2S/N} = 5.06$ times of Euclid. Actually, the PSD CLARANS averagely cost 5.3 times of Euclid CLARANS in the experiments.

5 Conclusion

In this paper, we present the geographic non-overlapping constraint which may commonly exist in real geography-related analysis works. Many existing clustering approaches with non-spatial attributes may not guarantee this requirement. It is also discussed that why this constraint can not be kept when non-spatial attributes exist, and we present the condition which can assure a distance measure to obtain the non-overlapping cluster areas. Then, a new spatial distance measure (PSD) is proposed and proofed to satisfy the condition. PSD adjusts the spatial distance by all non-spatial attribute values between the two points, and it can reflect the changing process of non-spatial attribute values from one point to another. PSD can be incorporated with CLARANS or other spatial clustering algorithm to solve knowledge discovery tasks. In the experiments, it performs better on artificial data sets and discovers the knowledge which meets the actual geographical characters in the real banking analysis case.

References

1. Rui Xu, Donald Wunsch II: Survey of Clustering Algorithms. IEEE Trans. on Neural Networks. **16** (2005) page(s): 645- 678
2. R. Ng, J. Han: CLARANS: A method for clustering objects for spatial data mining. IEEE Trans. on Knowledge and Data Engineering. **14** (2002). Pages: 1003- 1016.
3. Tung A. K. H., Hou J. and Han J.: Spatial clustering in the presence of obstacles. In Proc. of ICDE'01. Pages: 359 - 367.
4. R. Ng, J. Han: Efficient and Effective Clustering Methods for Spatial Data Mining. In Proc. of VLDB'94. Pages: 144 - 155.
5. V. Estivill-Castro and I. Lee: AUTOCLUST: Automatic Clustering via Boundary Extraction for Mining Massive Point-Data Sets. In Proc. of the 5th International Conference on Geocomputation. (2000)
6. V. Estivill-Castro and I. Lee: Autoclust+: Automatic clustering of point-data sets in the presence of obstacles. In Proc. of the International Workshop on Temporal, Spatial and Spatio-Temporal Data Mining. (2000). Pages: 133 - 146.
7. M. Ester, H.-P. Kriegel, J. Sander and X. Xu: A density-based algorithm for discovering clusters in large spatial databases with noise. In Proc. of SIGKDD'96 Pages: 226 - 231
8. O. R. Zaiane and C.-H.Lee: Clustering spatial data in the presence of obstacles and crossings: a density-based approach. In Proc. of International Database Engineering and Applications Symposium. (2002)
9. Xin Wang and H. J. Hamilton: DBRS: A Density-Based Spatial Clustering Method with Random Sampling. In Proc. of PAKDD'03. Pages: 563–575
10. Xin Wang, C. Rostoker, H. J. Hamilton: Density-Based Spatial Clustering in the Presence of Obstacles and Facilitators. In Proc. of PKDD'04. Pages: 446 - 458.

GBKII: An Imputation Method for Missing Values*

Chengqi Zhang[1,2], Xiaofeng Zhu[3], Jilian Zhang[3], Yongsong Qin[3], and Shichao Zhang[3]

[1] Faculty of Information Technology, University of Technology, Sydney
PO Box 123, Broadway NSW 2007, Australia
[2] Department of Information Systems, City University of Hong Kong, China
[3] Department of Computer Science, Guangxi Normal University, China

Abstract. Missing data imputation is an actual and challenging issue in machine learning and data mining. This is because missing values in a dataset can generate bias that affects the quality of the learned patterns or the classification performances. To deal with this issue, this paper proposes a Grey-Based K-NN Iteration Imputation method, called GBKII, for imputing missing values. GBKII is an instance-based imputation method, which is referred to a non-parametric regression method in statistics. It is also efficient for handling with categorical attributes. We experimentally evaluate our approach and demonstrate that GBKII is much more efficient than the k-NN and mean-substitution methods.

1 Introduction

In supervised learning, a learning system is given a training set of labeled instances, where each instance consists of a feature vector (conditional attributes) and an output value (class label). In real world applications, however, the training set often contains missing values that can generate bias that affects the quality of the supervised learning process or the performance of classification. However, most learning (or mining) algorithms are based on the assumption without missing values.

This paper proposes a new imputation algorithm to handle missing attributes values, called *GBKII* (***G**rey-**B**ased **K**NN **I**teration **I**mputation*), which is an instance-based imputation method and be referred to a non-parametric method in statistics. It is also efficient for dealing with categorical attributes. Specifically, this approach uses a grey relational grade (denoted as *GRG*) to substitute for Minkowski distance or other alternative similarity measures during the process of searching for the nearest neighbor under the assumption which requires the instance i to have a same class label as the instance j when calculating $GRG(i, j)$. This can efficiently reduce the time complexity. In addition, this approach can get over the slow convergence rate of EM

* This work is partially supported by Australian Research Council Discovery Projects (DP0449535, DP0559536 and DP0667060), a China NSF major research Program (60496327), China NSF grants (60463003, 10661003), an Overseas Outstanding Talent Research Program of Chinese Academy of Sciences (06S3011S01), an Overseas-Returning High-level Talent Research Program of China Hunan-Resource Ministry, and a Guangxi Postgraduate Educational Innovation Plan (2006106020812M35).

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1080–1087, 2007.

algorithm through an EM-like iteration imputation method and makes an optimal use of all observed values including those instances with missing values. We evaluate the performance of our method using several UCI datasets. The experimental results show that our approach is superior to *k-NN* and mean substitution methods.

The rest of this paper is organized as follows. Section 2 briefly recalls related work. In Section 3, we design our GBKII algorithm. Section 4 describes our experiments on UCI dataset [1]. We conclude this paper in Section 5.

2 Related Work

Currently, there are two mainstream directions for dealing with the missing values. One of these is based on machine learning. However, the methods based on machine learning perhaps destroy the original distribution of dataset during the process of imputing. Moreover, some methods (such as C4.5) usually only handled the discrete values. The other is based on statistics. These methods are usually targeted handling continuous attributes with missing values in class label. Our *GBKII* can handle with missing values occurring in categorical attributes and continuous ones.

Multiple Imputation fills in missing values with repeating independently M times [3]. EM algorithm [4] repeatedly alternates depend on parametric models. Our *GBKII* is an EM-like iteration imputation method. However, GBKII is different from the MI and EM algorithms. In the first iteration imputation, we use the mean (or mode) values of all the observed attribute values to fill in the missing values in order to make the best use of the all information. From the second imputation process, iteration imputation is based on the result of the last imputation. This procedure won't stop until the average change of imputed values is approximately stabilization, or satisfies a given user requirement. On the other hand, the nonparametric imputation [5] is efficient when the model of data is unknown a priori. In fact, we have usually not any priori knowledge about the data. Our *GBKII* algorithm is a non-parametric method that is different from EM algorithm in which both the E and M steps depend on parametric models.

3 The *GBKII* Algorithm

3.1 The Nearest Neighbor Imputation Method

Usually, calculating the nearest neighbor instance is based on the Minkowski distance or other distance measures. However, Caruana [2] deemed that it is sometimes difficult to devise a distance metric that combines distances measured between symbolic and numeric variables. Generally, Minkowski distances or other metrics are mainly suitable for some application domains, such as domains with numeric attributes. Caruana demonstrated grey relational analysis, which is more appropriate to determine the 'nearness' (or relationship) between two instances than Minkowski distances or others do [6], can deal with categorical attributes[1] and numeric attributes. Our algorithm uses

[1] Whose domain is totally ordered are called numeric, whereas attributes whose domain is not ordered are called categorical, the categorical attributes include symbolic attributes and discrete attributes.

GRG instead of Minkowski distances (or other distance metrics) during the process of searching for the similarities between instances.

3.2 The Grey Relational Analysis

Grey Relational Analysis (*GRA*), which is founded upon measuring the similarity of emerging trends among instances, is a method of *GST*. Consider a set of observations $\{x_0, x_1, x_2, \ldots, x_n\}$, where x_0 is the reference instance and $x_1, x_2, \ldots, x_n$ are the compared instances. Each instance x_i has *m* conditional attributes and is referred to $x_i = (x_i(1), x_i(2), \ldots, x_i(m)),\ \text{i=0,1,2,} \ldots \text{n}$ and a class label D_i. The grey relational coefficient (GRC) is defined as:

$$GRC(x_0(p), x_i(p)) = \frac{\min_{\forall j} \min_{\forall k} \left|x_0(k) - x_j(k)\right| + \rho \max_{\forall j} \max_{\forall k} \left|x_0(k) - x_j(k)\right|}{\left|x_0(p) - x_i(p)\right| + \rho \max_{\forall j} \max_{\forall k} \left|x_0(k) - x_j(k)\right|} \tag{1}$$

where $\rho \in [0,1]$ (ρ is a distinguishing coefficient, normally, let ρ =0.5), $i = j = 1, 2, \ldots, n,$ and k=p=1, 2, …, *m*. The grey relational grade(*GRG*) is referred to as follow:

$$GRG(x_0, x_i) = \frac{1}{m} \sum_{k=1}^{m} GRC(x_0(k), x_i(k)),\ \text{i=1,2,} \ldots \text{,n.} \tag{2}$$

If $GRG(x_0, x_1)$ is greater than $GRG(x_0, x_2)$, then the difference between x_0 and x_1 is smaller than that between x_0 and x_2; otherwise the former is larger than the latter.

3.3 *GBKII* Algorithm

Previous work, such as kernel method, imputes the missing values utilizing the instances without missing value as the reference instances. It may possibly ignore two facts: (1) There are less observed data in database. In practice, most of industrial databases have a more serious problem about values missing especially in industrial database, such as in [7], of the 4383 records in this database, none of the records were

The *GBKII* algorithm is presented as follows:

```
   The First Iteration
1.0    // t-th iteration
   Repeat
   2.0.1  Compute GRG(i,j) base on Equation (2)
   2.0.2  Get k Nearest Instances
   2.0.3  Imputation Ming Values
   2.0.4   t - -;            // t is the iteration time
   Until (convergence or t<=0)
```

Fig. 1. The Pseudo-code of GBKII algorithm

complete and only 33 variables out of 82 have more than 50% of the records complete. (2) An incomplete instance may already contain enough information for model construction, even though it still contains missing values. In Table 1, the value in 5-th attribute are missing and their class label are '1', that will be wrong if we compute *GRG*(*a, i*) (*i* is the instance without missing values, such as instance *d* and *e*) because the class label is different from the missing instance. So it is reasonable for us to impute missing values with all observed values including those instances that contain some missing values based on the above analysis.

However, we cannot impute missing values with all the information of the observed values before the missing values have not yet been patched up. For example, we cannot compute the *GRG*(*a,b*) as the 2nd attribute in instance *b(denoted as MV*(*b,2*)*)* is missing and only one value in the 2nd attribute (denoted as *V(a,2)*)is observed in Table 1. In *GBKII*, we apply the first imputation strategy to make the best use of all the information of observed values; and the (*t+1*)*-th* (*t*>1) iteration imputation is based on the imputation results of the *t-th* imputation until convergence or satisfying the demand of users. We compute the mean (or the mode if the attribute is categorical) for each continuous-attributed observed values whose class labels are the same, i.e., we use the mean (or mode) as the initial imputed value of the missing value. Imputation with mean (or mode) is a popular and reasonable imputation method in machine learning and statistics. However, [8] thought to impute with the mean (or mode) is valid if and only if the dataset is chosen from a population with a normal distribution. However, in real world application, we cannot know really the real distribution of the dataset in advance. So running the extra iteration imputations is reasonable based on the first imputation for dealing with the missing values.

In the second iteration imputation, for example in Table 1, we assume all the values are all observed (the missing values have got in the first iteration imputation by the mean or the mode) except the value of MV(a,1) if we want to impute the missing value in attribute C_1 in instance *a*. We calculate *GRG*(*a,i*) (*i* is *b* or *c in Table 1*) among these instances whose class label are '*1*'as same as the missing instance *a*, then we impute MV(a,1) based on method in Figure 1 through the step 2.0. We regard the attribute C_3 in instance *a* as missing when we have imputed the value in attribute C_1 in instance *a* and want to impute MV(a,3) in second iteration, in this case we regard the value in attribute C_1, C_4, C_5 in instance *a* are observed. Then we impute the C_4 and C_5 in instance *a* by turn utilizing the same method after imputing C_3 in second iteration.

Table 1. '-'denotes observed values and '?'denotes missing values in a relation database

	C_1	C_2	C_3	C_4	C_5	D
a	**?**	**-**	**?**	**?**	**?**	1
b	**-**	**?**	**-**	**-**	**?**	1
c	**?**	**-**	**-**	**-**	**?**	1
d	**-**	**-**	**-**	**-**	**-**	0
e	**-**	**-**	**-**	**-**	**-**	0

And so forth, we can impute all missing values in the dataset. During the third iteration imputation or the next ones, we can repeat the second iteration imputation until the imputation results satisfy the demand of users or the algorithm reaches convergence.

At last, we present in detail how the proposed approach is extended to deal with symbolic attributes. Similar to the methods used in [6], if x_0 and x_i have the same values for symbolic attribute p, GRC($x_0(p)$, x_i (p)) =1 (i.e., the similarity between $x_0(p)$ and x_i (p) is maximal). By contrast, if x_0 and x_i have different values for symbolic attribute p, GRC(x_0(p), x_i (p)) =0 (i.e., the similarity between $x_0(p)$ and x_i (p) is minimal). Thus, the proposed approach can be applied to numeric and categorical attributes with missing values.

3.4 Convergence of the Imputed Values

In our algorithm, the first iteration, which uses the mean (or mode) as the initial filled-in value of the missing values under the assumption of same class label, is obviously convergence in statistics, but in the process of the other iterations, we are not able to make similar proof for the non-parametric method. The reason is that there are few theoretical results regarding the validity of *k-NN* in the literature due to the difficulty of building a mathematical proof. In this section we empirically show the convergence of the *GBKII* method on UCI datasets.

Note that in this paper, the algorithm of iteration imputation with grey based *k-NN* method is denoted as 'Noclassified'; the algorithm of iteration imputation based Euclidean distance *k-NN* method under the same class label between the missing instance and its nearest neighbor is denoted as '*k-NN*'; the algorithm which filled in missing attribute values based mean or mode under the same class label is denoted as 'MeanMode'.

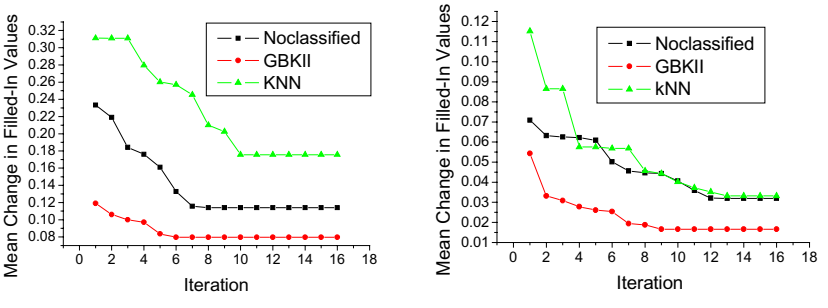


Fig. 2. Mean change in filled-in values for "the Hepatitis Diagnosis Problem" (left) and "Water-Treatment Domain"(right) data set for successive iteration

The imputation method converges when the "mean change in filled-in values" reaches to zero. The meaning of "mean change in filled-in values" is the distance between the mean of all imputed in last iteration imputation and the mean of all imputed in the current. Caruana [8] thinks that the "mean change in filled-in values" usually does not drop all the way to zero and only approximate to a value that is as

close as possible to zero in the non-parametric model (such as *k-NN*, kernel method) in practice. Figure 2 shows that the "mean change in filled-in values" moves to zero in the successive iteration imputations when the algorithm is applied to the hepatitis diagnosis data set and water-treatment domain data set respectively. The results show that all the three algorithms are convergence because the "mean change in filled in values" remains stable after some iterations. However, our algorithm '*GBKII*' is the fastest with respect to convergence among these three algorithms for the *hepatitis* dataset and *water-treatment* dataset. The "mean change in imputed values" of our *GBKII* is the smallest among the three algorithms when running the two UCI datasets.

4 Experimental Studies

4.1 Experimental Evaluation on Prediction Accuracy

First, the *GBKII* approach is evaluated on Iris dataset and the *Pima* dataset in order to demonstrate the approach's effectiveness. There are no missing values in the datasets and the attribute data are missing at random and the missing rate are fixed to 5%. As we had no prior information about the optimal k for a specified application, the optimal value of k will be obtained by experimental tests in our algorithm, i.e. k, varied from 1 to 30. We iterate imputation 20 times based on the analysis of Figure 2. The accuracy of prediction was measured using the Root Mean Square Error (RMSE) as follows:

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(e_i - \tilde{e}_i)^2} \tag{3}$$

where e_i is the original attribute value; $\tilde{e}_i$ is the estimated attribute value, and m is the total number of missing values. The larger the RMSE is, the worse the prediction accuracy is.

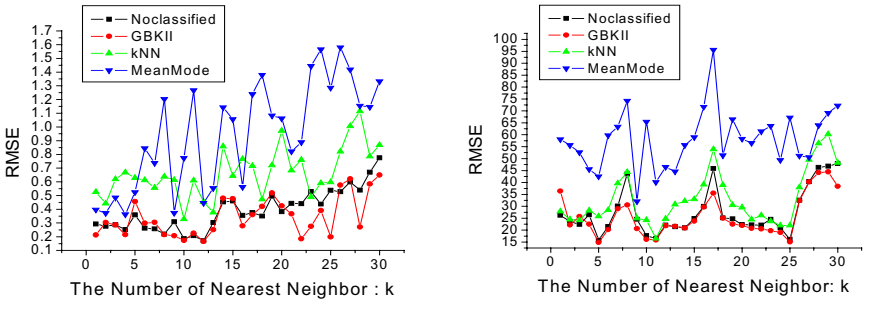


Fig. 3. Experimental result on the Iris dataset (left) and Pima dataset (right) for four algorithms (the missing rate is 5%, the number of iteration time is 20)

From Figure 3, we can see that the RMSE for grey base *k-NN* iteration imputation algorithm (including 'Noclassified' and *GBKII*) is smaller than the Euclidean based

k-NN iteration imputation algorithm and mean algorithm regardless of the varied *k* (the number of the nearest neighbors), the performance of RMSE for *GBKII* is better than the 'Noclassified' between the grey based methods.

4.2 Experimental Evaluation on Classification Error Rate

Two UCI datasets (i.e., *Hepatitis Diagnosis* Problem dataset and *Water-Treatment Domain* dataset) are applied to compare the performances of the five algorithms. At first, we get the classification accuracy on original dataset (denoted as 'Origin' in Table 2) which is an incomplete dataset and hasn't filled up with any imputation methods by C5.0 (available at www.rulequest.com), then we get four completed datasets by four imputation methods, for example 'Noclassified', *GBKII*, *k-NN,* and 'MeanMode''. We present the results of classification Error Rate of these four algorithms.

Table 2. The Classification Error Rate of four imputation methods and the Classification Error Rate of incomplete dataset for Hepatitis and Water-Treatment *dataset*

	Origin	MeanMode	kNN	Noclassified	GBKII
Hepatitis	0.348	0.324	0.269	0.224	0.203
Water-Treatment	0.529	0.553	0.365	0.334	0.258

From Table 2, the results of the four imputation methods are significantly well than the method no imputing, this show we maybe impute the missing values rather than no imputing it. The classification error rate for grey base *k-NN* iteration imputation algorithm (including 'Noclassified' and *GBKII*) obviously outperforms the Euclidean based *k-NN* iteration imputation algorithm and mean algorithm in classification error rate. For the grey-based methods, the classification error rate of *GBKII* is less than 'Noclassified' method.

4.3 Experimental Evaluation on Single Imputation and Iteration Imputation

In this subsection, we compare the performances of our algorithm with the single imputation method (the single imputation method imputes missing values by using the grey based *k-NN* method, but it searches the nearest neighbors within instances that have same class label). The results in Figure 4 show that it is reasonable for us to adopt the iteration imputation method to deal with missing attributes.

	Iris	Pima
Single	0.24596	28.452
Iteration	0.16542	14.8889

	Hepatitis	Water
Single	0.291	0.463
Iteration	0.203	0.258

Fig. 4. Experimental result on the RMSE (left) and Classification Error Rate (right) for single imputation and GBKII (the number of all iterations is 8)

5 Conclusions and Future Work

As we have seen, *GBKII* is an instance-based imputation method and a nonparametric method in statistics. Different from existing imputation methods, *GBKII* is able to deal with categorical attributes with missing values. In this approach, the grey relational analysis, which is more appropriate to determine the 'nearness' (or relationship) between two instances than Minkowski distance does, has been used to describe the relational structure of all instances and can accelerate the convergence rate of iteration imputation. On the other hand, *GBKII* searches for the nearest neighbor instance with the same class label between the instance and the missing instance, which can reduce the time complexity, and improves the prediction errors. In particular, this EM-like iteration imputation method can get over the problem of the slow convergence rate of EM algorithm and make the best use of all the information of observed values including the values with missing values. Experimental results of four UCI datasets have showed that our method is superior to *k-NN* and mean (or mode) substitution in convergence rate, RMSE for prediction accuracy and classification error rate.

References

[1] Blake, C. and Merz, C. *UCI Repository of machine learning databases*.1998.
[2] Caruana, R. A Non-parametric EM-style algorithm for Imputing Missing Value. *Artificial Intelligence and Statistics*, January 2001.
[3] Rubin, D.B., *Multiple Imputation for Nonresponse in Surveys*, Wiley: New York, 1987.
[4] Dempster, A.P., Laird, N.M. and Rubin, D.B., Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, series B, Vol. 39, pp. 1–38, 1977.
[5] Zhang, S.C., Qin, Y.S., Zhu, X.F., Zhang, J.L., and Zhang, C.Q. Optimized Parameters for Missing Data Imputation. *PRICAI06*, 2006: 1010-1016.
[6] Huang, C.C. and Lee, H.M. An instance-based learning approach based on grey relational structure. In: *Proc. of the UK Workshop on Computational Intelligence* (UKCI-02), Birmingham, Sept. 2002.
[7] Lakshminarayan, K. et al., Imputation of missing data in industrial databases, *Applied Intelligence*, Vol. 11, pp. 259-275, 1999.
[8] Brown, M.L. Data mining and the impact of missing data. *Industrial Management & Data Systems*, Vol. 103/8, pp. 611-621, 2003.

An Effective Gene Selection Method Based on Relevance Analysis and Discernibility Matrix*

Li-Juan Zhang[1], Zhou-Jun Li[2], and Huo-Wang Chen[1]

[1] National Laboratory for Parallel and Distributed Processing, Changsha, China
[2] School of Computer Science & Engineering, Beihang University, Beijing, China
nudtzlj@126.com

Abstract. Selecting a small number of discriminative genes from thousands of genes in microarray data is very important for accurate classification of diseases or phenotypes. In this paper, we provide more elaborate and complete definitions of feature relevance and develop a novel feature selection method, which is based on relevance analysis and discernibility matrix to select small enough genes and improve the classification accuracy. The extensive experimental study using microarray data shows the proposed approach is very effective in selecting genes and improving classification accuracy.

Keywords: gene selection, relevance analysis, discernibility matrix.

1 Introduction

Recent advanced technologies in DNA microarray analysis are intensively applied in disease classification, especially for cancer classification [1], [2]. However, classification based on microarray data is very different from previous classification problems in that the number of genes (typically tens of thousands) greatly exceeds the number of samples (typically less than one hundred), which result in the known problem of "curse of dimensionality" and over-fitting of the training data [2]. It is thus important for successful cancer classification to select a small number of discriminative genes from thousands of genes [1], [3].

Recently, Feature selection, an effective dimensionality reduction technology in machine learning and data mining, has been extensively applying to gene selection for cancer classification. Feature selection algorithms can broadly fall into the filter model and the wrapper model [4]. The filter model relies on general characteristics of the data to evaluate and select gene subsets without involving any mining algorithm. The wrapper model requires one predetermined mining algorithm and uses its performance as the evaluation criterion. It is computationally expensive for data with a large number of features [4]. For this reason filter model is widely used in gene selection.

Among filter-based gene selection methods, earlier methods are gene ranking. They usually evaluate each gene individually by assigning a discriminative score to each

* This work is supported by the National Science Foundation of China under Grants No. 60573057, 60473057 and 90604007.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1088–1095, 2007.

gene, then rank genes by their scores and select the top ranked ones [1],[5],[6]. These methods are efficient for high-dimensional data due to linear time complexity in terms of dimensionality. However they cannot handle redundancy among genes, and require user to determine the threshold for the number of selected genes.

Many latest methods handle redundancy by considering the correlation between genes. These methods can be classified into three categories. One category is based on clustering, which firstly group similar genes into clusters, genes in the same cluster are considered to be high correlated, and then select the most relevant genes from each cluster to represent this cluster [7], [8]. Another category integrates the metric for measuring the gene-class relevance and that for measuring the gene-gene redundancy into a single criterion function and then selects genes so that the criterion function is optimized [9], [10]. The above categories of methods can remove redundant genes to certain extent but are all time consuming. The third category uses a new framework that decouples relevance analysis and redundancy analysis. They usually measure the gene-class relevance to obtain relevant genes, and then use a methodology called *redundant cover* to remove redundancy. Methods in this category have been proved to be effective and efficient on many high dimensional data sets [3], [11].

In this paper, we develop a novel gene selection approach based on relevance analysis and discernibility matrix, which can select genes and improve the classification accuracy more effectively. The remainder of this paper is organized as follows. In section 2, we provide more elaborate definitions of relevance. In section 3 we describe the proposed method. Section 4 evaluates the performance of our method via extensive experiments. Section 5 concludes this work.

2 Feature Relevance

Let F be a full set of features, C the class label and $S \subseteq F$. In general, the goal of feature selection can be formalized as selecting a minimum subset S such that $P(C|S)$ is equal or as close as possible to $P(C|F)$, where $P(C|S)$ is the probability distribution of different classes given the feature values in S and $P(C|F)$ is the original distribution given the feature values in F [12]. Such a minimum subset is usually called an optimal subset. Kohavi and Sommerfield [13] show that an optimal feature subset must be from relevant features.

But what features are relevant? About this problem, there are many contributions [4], [14]. The definitions identify a feature as either relevant or irrelevant to a concept or task. John and Kohavi [4] show that these definitions give unexpected results, and that the dichotomy of relevance vs irrelevance is not enough. An alternative definition of relevance is then proposed which distinguishes between strong relevance and weak relevance. The distinction of strong relevance and weak relevance has the advantage of flexibility: using this distinction, we can select either strongly relevant features or weakly relevant features to satisfy different learning requirements [13]. Taking this one step further, it is then reasonable to expect a finer distinction of relevance.

We here propose a finer distinction of relevance based on the influence of a feature on classification performance. The presence of a feature has three influences on classification performance: improvement, deterioration and no change. Therefore we divide feature relevance into positive relevance, negative relevance and irrelevance.

Positive (negative) relevance implies that a feature has a positive (negative) effect on classification performance, while irrelevance means that a feature has no effect on classification performance. In addition, if the influence of a feature on classification performance is independent of any other feature, we call such influence strong relevance, weak relevance otherwise.

Based on the above analysis, we classify feature relevance into six categories: strong positive relevance, strong negative relevance, strong irrelevance, weak positive relevance, weak negative relevance and weak irrelevance. Let f_i be a feature, $S_i = F - \{f_i\}$ .These categories of relevance can be formalized as follows.

Definition 1 (Strong positive relevance): a feature f_i is strongly positive relevant *iff*

$\forall S_i' \subseteq S_i, P(C \mid f_i, S_i') > P(C \mid S_i')$

Definition 2 (Strong negative relevance): a feature f_i is strongly negative relevant *iff*

$\forall S_i' \subseteq S_i, P(C \mid f_i, S_i') < P(C \mid S_i')$

Definition 3 (Strong irrelevance): a feature f_i is strongly irrelevant *iff*

$\forall S_i' \subseteq S_i, P(C \mid f_i, S_i') = P(C \mid S_i')$

Definition 4 (Weak positive relevance): a feature f_i is weakly positive relevant *iff*

$\exists S_i' \subseteq S_i, P(C \mid f_i, S_i') > P(C \mid S_i')$ and $\neg \forall S_i' \subseteq S_i, P(C \mid f_i, S_i') > P(C \mid S_i')$

Definition 5 (Weak negative relevance): a feature f_i is weakly negative relevant *iff*

$\exists S_i' \subseteq S_i, P(C \mid f_i, S_i') < P(C \mid S_i')$ and $\neg \forall S_i' \subseteq S_i, P(C \mid f_i, S_i') < P(C \mid S_i')$

Definition 6 (Weak irrelevance): a feature f_i is weakly irrelevant *iff*

$\exists S_i' \subseteq S_i, P(C \mid f_i, S_i') = P(C \mid S_i')$ and $\neg \forall S_i' \subseteq S_i, P(C \mid f_i, S_i') = P(C \mid S_i')$

Definition 7 (Strongly positive relevant feature subset): a feature subset $S \subseteq F$ is a strongly positive relevant feature subset, *if* $\forall f_i \in S$, f_i is strongly positive relevant.

Strong relevance means an absolute or unconditional relationship between a feature and class. A strongly positive relevant feature is absolutely beneficial to classification and its removal will result in performance deterioration. Reversely a strongly negative relevant feature is completely harmful to classification and its presence will result in performance deterioration. A strongly irrelevant feature is completely irrelevant with classification. Weak relevance implies a relative or conditional relationship between a feature and class. A weakly relevant feature may be multi-role, since two or three of the following conditions may hold simultaneously for different S_i' : $\exists S_i' \subseteq S_i, P(C \mid f_i, S_i') > P(C \mid S_i')$, $\exists S_i' \subseteq S_i, P(C \mid f_i, S_i') < P(C \mid S_i')$ and $\exists S_i' \subseteq S_i, P(C \mid f_i, S_i') = P(C \mid S_i')$. Whether a weakly relevant feature is beneficial to classification depends on the other features already selected and on the evaluation measure that has been chosen. Once the current selected features and the evaluation measure are both given, we can decide whether a weakly relevant feature is weakly positive relevant. If it is weakly positive relevant, then it is beneficial to classification, otherwise harmful or useless.

4 Methodology

In this section we are ready to develop a simple and effective gene selection method. Based on the above definitions and analysis, only strongly positive relevant and weakly positive relevant features are beneficial to classification. The feature subset selected by our method should only include all strongly positive relevant and weakly positive relevant features. Therefore our method must solve the following two questions: (1) how to identify strongly positive relevant features, and (2) how to decide whether a feature is weakly positive relevant.

The answer to the first question can be using discernibility matrix from rough set theory [15]. Given a data set *S*, feature set $F=\{ f_1, f_2, \ldots, f_n \}$, class attribute is *C*, sample set $U=\{ x_1, x_2, \ldots, x_m \}$, let $f_i(x_j)$ denote the value of the sample x_j for the feature f_i, then the discernibility matrix of data set *S* is a symmetric $|U| \times |U|$ matrix with entries c_{ij} defined as $\{ f_k \in F \mid f_k(x_i) \neq f_k(x_j) \}$ if $C(x_i) \neq C(x_j)$, $\varnothing$ otherwise.

According to the definition of discernibility matrix, if there is only one element in an entry c_{ij}, it must be a strongly positive relevant feature because it is the only feature that can differentiate sample x_i and x_j. Based on this observation, we can examine each entry in the discernibility matrix to obtain all the strongly positive relevant features. It is possible that no entry contains only one feature, so the strongly positive relevant feature subset may be an empty set.

The answer to the second question is more complicated, because whether or not a feature is weakly positive relevant depends on the current selected features and the evaluation measure. Among existing evaluation measures, the correlation measure has been widely used and shown effective [11]. We here adopt the correlation measure to evaluate the goodness of a feature subset.

We denote the correlation value of a feature f_i and the class *C* as $Corr(f_i, C)$, and let $F_S \subseteq F$ be a feature subset, then we can use $Corr(f_i, C)$ to define the correlation between F_S and class *C* as follows:

$$Corr(F_S, C) = \frac{1}{|F_S|} \left(\sum_{f_i \in F_S} Corr(f_i, C) \right) \tag{1}$$

Given F_S is the current selected feature subset, *E* is the selected correlation measure. Based on the previous analysis, a feature f_i ($f_i \notin F_S$) is weakly positive relevant and should be included *iff* $Corr_E(\{ f_i \} \cup F_S, C) > Corr_E(F_S, C)$.

Besides the above two questions, we must also decide the search starting point and search strategy for the process of feature selection. Because strongly positive feature subset is absolutely necessary, we start search with it to prevent losing some strongly positive features in the successive process of search. If the strongly positive feature subset is an empty set, search will start with an empty set. It is important for high dimensional microarray data to search feature space quickly. We therefore rank

features according to the selected correlation measure *E* and adopt sequential forward search strategy, which is simple to implement and fast.

Based on the above analysis, we can easily obtain a deterministic procedure that can effectively identify discriminative genes in microarray data set. Our method can be summarized by an algorithm FRADM (Filter based on Relevance Analysis and Discernibility Matrix) shown in Figure 1. As in Figure 1, given a microarray data set *S* with *n* genes, *m* samples and a class *C*, the algorithm consists of three parts. In the first part (line 1-2), it constructs a discernibility matrix from the input data set, finds the strongly positive gene subset based on the discernibility matrix. In the second part (line 3-4), it calculates the correlation between the starting set and the class *C* and ranks genes using the predefined correlation measure *E*. In the third part (line 5-6), it iteratively adds genes from the ranked list *R*. The iteration starts from the first element in the ranked list *R* and continues as follows.

Input: S (F, C)-training data;
F-original feature set; E-measure
Output: F_{best} -best feature subset
1. Let $F_{best}=\varnothing$. And construct discernibility matrix DM for S.
2. For each entry c_{ij} in DM, If c_{ij} contains only one feature, then add the feature into F_{best}.
3. If $F_{best}=\varnothing$ then $Corr_E(F_{best}, C)=0$; else calculate $Corr_E(F_{best}, C)$ according to formula 1.
4. For each feature f_i in F-F_{best}, calculate $Corr_E(f_i, C)$ and rank them in descending $Corr_E(f_i, C)$ value, form the ranked list R.
5. Get next feature f_i from the list R, if $Corr_E(\{f_i\} \cup F_{best}, C) > Corr_E(F_{best}, C)$ then add f_i into F_{best}.
6. Repeat 5 until the end of the list R.

Fig. 1. Filter based on Relevance Analysis and Discernibility Matrix

5 Experiments and Results

In this section, we evaluate our approach in terms of degree of dimensionality and classification accuracy on selected genes. For gene selection in sample classification, it is perfect to select small enough genes which can lead to high enough classification accuracy.

We perform extensive experiments using ten microarray data sets[1]. The main characteristic of these data sets is the great number of genes and the relatively small number of samples. The details of these data sets are summarized in Table 1.

Table 1. Summary of data set

Title	#genes	# samples	#class
colonTumor	2000	62	2
breastCancer	24481	97	2
lungCancer harvard	12533	181	2
MLL_Leukemia	12582	72	3
DLBCLTumor	7129	77	2
DLBCLOutcome	7129	58	2
prostate_outcome	12600	21	2
prostate_tumorVSNormal	12600	136	2
centralNervousSystem	7129	60	2
Stjude_Leukemia(BCR-ABL)	12558	15	2

Three representative filter algorithms are chosen in comparison with FRADM. One algorithm representing feature ranking methods is ReliefF [6], which searches for nearest neighbors of instances of different classes and ranks features according to their importance in differentiating instances of different classes. Another algorithm is a variation of CFS [10], denoted by CFS-SF (Sequential Forward), which used some correlation measure and sequential forward search to obtain optimal subset. A third one is FCBF [11], which used the correlation measure *symmetrical uncertainty* to obtain relevant genes and to remove redundancy. In addition, two widely used classification algorithms, C4.5 and NaiveBayes, are adopted to evaluate the predictive accuracy of the selected genes. The experiments are conducted using WEKA's implementation of all these existing algorithms and our algorithm is also implemented in the WEKA environment [16].

Table 2. Number of genes selected by each feature selection algorithm

Title	Full set	FRADM	FCBF	ReliefF	CFS-SF
colonTumor	2000	5	14	5	26
breastCancer	24481	7	90	7	N/A
lungCancer harvard2	12533	7	128	7	N/A
MLL_Leukemia	12582	5	97	5	N/A
DLBCLTumor	7129	5	73	5	N/A
DLBCLOutcome	7129	5	27	5	N/A
prostate_outcome	12600	4	27	4	N/A
prostate_tumorVSNormal	12600	6	38	6	N/A
centralNervousSystem	7129	4	28	4	N/A
Stjude_Leukemia(BCR-ABL)	12558	5	89	5	N/A

[1] http://sdmc.lit.org.sg/GEDatasets/Datasets.html

For each data set, we first run all the feature selection algorithms, and obtain the selected genes for each algorithm. Note that in FRADM, the evaluation measure is set to *symmetrical uncertainty*. For ReliefF, we use 5 neighbors and 30 instances throughout the experiments, and to compare the performance of ReliefF and FRADM, the number of genes selected by ReliefF is set to be the same as that of FRADM. We then apply C4.5 and NaiveBayes respectively on each original data set and each newly obtained data set only containing the selected genes, and obtain the overall classification accuracy by leave-one-out cross-validation.

Table 3. Leave-one-out cross-validation accuracy of C4.5 on selected genes for each feature selection method (%)

Title	Full set	FRADM	FCBF	ReliefF	CFS-SF
colonTumor	80.65	85.48	88.71	82.26	87.10
breastCancer	57.73	78.35	67.01	58.76	N/A
lungCancer harvard2	96.13	97.24	98.90	98.90	N/A
MLL_Leukemia	86.11	90.28	93.06	86.11	N/A
DLBCLTumor	80.52	90.91	85.71	84.42	N/A
DLBCLOutcome	37.93	65.52	46.55	48.28	N/A
prostate_outcome	23.81	85.71	33.33	38.10	N/A
prostate_tumorVSNormal	77.94	93.38	83.82	82.35	N/A
centralNervousSystem	50	75	66.67	50	N/A
Stjude_Leukemia(BCR-ABL)	92.97	97.55	93.27	96.33	N/A

Table 4. Leave-one-out cross-validation accuracy of NaiveBayes on selected genes for each feature selection method(%)

	Full set	FRADM	FCBF	ReliefF	CFS-SF
colonTumor	75.81	90.32	77.42	82.26	80.65
breastCancer	71.13	90.72	55.67	69.07	N/A
lungCancer harvard2	98.34	100	99.45	98.34	N/A
MLL_Leukemia	91.67	95.83	94.44	87.5	N/A
DLBCLTumor	85.71	92.21	92.21	85.71	N/A
DLBCLOutcome	51.72	91.38	53.45	51.72	N/A
prostate_outcome	33.33	95.24	33.33	61.90	N/A
prostate_tumorVSNormal	63.97	88.97	65.44	61.76	N/A
centralNervousSystem	65	80	56.67	66.67	N/A
Stjude_Leukemia(BCR-ABL)	98.47	97.86	95.11	93.88	N/A

Table 2 records the number of genes selected by each feature selection algorithm. We can see that FRADM on average selects the smallest number of features. Table 3-4 reports the leave-one-out accuracy by C4.5 and NaiveBayes respectively. For most of the data sets, we can observe that, (1) CFS-SF is not available due to its $O(n^2)$ complexity in terms of the number of genes n. (2) FRADM can increase or maintain the accuracy of C4.5 and NaiveBayes; and (3) none of other three algorithms can enhance the accuracy of C4.5 and NaiveBayes to the same level as FRADM does. In summary, the above experimental results suggest that FRADM is effective in gene selection and is practical for use in sample classification of high dimensional microarray data.

6 Conclusions

In this work, we have provided more elaborate definitions of feature relevance and proposed a novel filter method that is based on relevance analysis and discernibility matrix to select small enough number of genes and improve the classification accuracy. Extensive experiments on microarray data have demonstrated the superior performance of FRADM.

References

1. T. R. Golub et al. Molecular classifications of cancer: Class discovery and class prediction by gene expression monitoring. Science, 286(5439):531–537, 1999.
2. E. R. Dougherty. Small sample issue for microarray-based classification. Comparative and Functional Genomics, 2:28-34, 2001.
3. E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. In Proceedings of the 18th International Conference on Machine Learning, pages 601-608, 2001.
4. R. Kohavi and G. John. Wrappers for feature subset selection. Artiticial Intelligence, 97(1-2):273-324, 1997.
5. Li-Juan Zhang and Zhou-Jun Li, Gene Selection for classifying microarray data using grey relational analysis. Proceedings of Discovery Science'2006, Lecture Notes in Computer Science Vol.4265: 378-382, 2006
6. M. Robnik-Sikonja and I. Kononenko. Theoretical and empirical analysis of Relief and ReliefF. MachineLearning, 53:23-69, 2003.
7. Li-Juan Zhang, Zhou-Jun Li, Huo-Wang Chen and Jian Wen, Minimum Redundancy Gene Selection based on Grey Relational analysis. In Workshops Proceedings of ICDM'2006 pages 120-124, IEEE Computer Society, 2006.
8. J. Jaeger, R. Sengupta, and W. L. Ruzzo. Improved gene selection for classification of microarrays. In Proc. PSB, 2003.
9. Xiaoxing Liu, Arun Krishnan and Adrian Mondry, an entropy-based gene selection method for cancer classification using microarray data, BMC Bioinformatics 2005,6: 76
10. M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In Proceedings of the 17th International Conference on Machine Learning, pages 359-366, 2000.
11. L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learning Res. 5 (2004) 1205–1224.
12. D. Koller and M. Sahami. Toward optimal feature selection. In Proceedings of the 13th International Conference on Machine Learning, pages 284-292,1996.
13. Kohavi, R.&Sommerfield, D. (1995). Feature subset selection using the wrapper method: Overfitting and dynamic search space topology. In U. M. Fayyad & R. Uthurusamy (Eds.), Proceedings of KDD'95 (pp. 192–197).
14. Blum, A. (1994). Relevant examples & relevant features: thoughts from computational learning theory. In Relevance:Proc. 1994 AAAI Fall Symposium pages 14–18, AAAI Press.
15. Skowron A, Rauszer C. The discernibility matrices and functions in information systems. In: Slowinski R ed. Intelligent Decision Support—Handbook of Applications and Advances of the Rough Sets Theory, Kluwer Academic Publishers, 1992: 331-362.
16. I. Witten and E. Frank. Data Mining – Pracitcal Machine Learning Tools and Techniques with JAVAImplementations. Morgan Kaufmann Publishers, 2000.

Towards Comprehensive Privacy Protection in Data Clustering

Nan Zhang

Department of Computer Science and Engineering
University of Texas at Arlington, Arlington, TX 76019-0015, USA
nzhang@cse.uta.edu

Abstract. We address the protection of private information in data clustering. Previous work focuses on protecting the privacy of data being mined. We find that the cluster labels of individual data points can also be sensitive to data owners. Thus, we propose a privacy-preserving data clustering scheme that extracts accurate clustering rules from private data while protecting the privacy of both original data and individual cluster labels. We derive theoretical bounds on the performance of our scheme, and evaluate it experimentally with real-world data.

1 Introduction

Data clustering is the process of grouping data tuples into different clusters, such that tuples within one cluster are similar with each other but are dissimilar to tuples in other clusters. It is an important problem in data mining, and has been successfully used in various domains such as image processing, market research, and bioinformatics [2]. In many cases, it is important to conduct data clustering on private data without violating the privacy of data owners.

In this paper, we address the privacy protection problem for a distributed system in which data being mined are stored across multiple autonomous entities. In our previous work [7], we classified such distributed systems into two categories based on their infrastructures, namely *Server-to-Server* (S2S) and *Client-to-Server* (C2S), respectively. An S2S system consists of several servers, each of which holds a private database. The servers collaborate to perform data mining across their databases without disclosing private information to each other. A C2S system consists of one data miner (server) and multiple data providers (clients), each of which holds a private data tuple. The data miner is supposed to collect data from the data providers and perform data mining on the collected data. Online survey is a typical example of a C2S system, as there is one survey analyzer (data miner) and numerous survey respondents (data providers).

Both C2S and S2S systems have a broad range of applications. Nevertheless, we focus on C2S systems in this paper. Most previous work in C2S systems aims to protect (only) the values of the original private data (i.e., the data being mined) in such systems [5]. While this is an important task, we find that in many cases, the data owners have privacy concerns on not only the original data, but also the cluster labels of their individual data. For example, a customer may be willing to provide (the privacy-protected version of) his/her purchase record for a company to analyze its customer

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1096–1104, 2007.

bases and thereby reduce advertisement cost. Nonetheless, the customer may not want the company to label him/her in a special group (e.g., gamer, new-product enthusiast, or deal hunter).

In order to provide comprehensive privacy protection in data clustering systems, a privacy-preserving mechanism must *simultaneously* prevent the disclosure of original data as well as the labels of such data. In particular, the data miner should not learn which data tuples belong to the same cluster. Ideally speaking, the only information that the data miner can learn from the collected data should be the *clustering rules* that can accurately classify different clusters. For example, if k-means clustering algorithm is used, the data miner should learn the accurate center point of each cluster, and nothing else, after the data collection and clustering process.

In this paper, we propose an algebraic-approach-based scheme that protects the privacy of both the original data and their cluster labels, while allowing the data miner to generate accurate clustering rules. We will show that our new scheme has the following important features that distinguish it from previous approaches:

- Up to our knowledge, this paper is the first to address the protection of both original private data and individual cluster labels in C2S data clustering systems.
- Our scheme allows each data provider to choose a different level of privacy disclosure based on his/her individual privacy concern.
- Our scheme is transparent to the clustering algorithm used and can be readily integrated into existing systems as middle-ware.

The algebraic-techniques-based approach was first proposed in our previous work for other data mining problems (e.g., data classification [7]). There are significant differences between the work presented in this paper and our previous work which include:

- The data mining problem is different: we are dealing with data clustering problem instead of association rule mining and data classification problems.
- The private information is different: we are preserving not only the privacy of original data (as in [7]), but also the privacy of individual cluster labels.

The rest of the paper is organized as follows. We introduce the system model and problem definition in Section 2. In Section 3, we present the baseline architecture of our new scheme. We introduce the basic components of our scheme in Section 4. In Section 5, we evaluate the performance of our scheme theoretically and derive bounds on privacy and accuracy measures. We experimentally evaluate the performance of our scheme on real data set in Section 6, and conclude with final remarks in Section 7.

2 System Model and Problem Definition

Let there be one data miner S and m data providers $P_1, \ldots, P_m$ in the system. Each data provider P_i holds a data tuple t_i ($i \in [1, m]$) with n attributes $a_1, \ldots, a_n$. The clustering process is consisted of two steps. In this first step, the data miner collects data from the data providers. As in an online survey system where each survey respondent joins the system at a different time, we consider this step to be iteratively carried out by a group of independent processes, within each of which a data provider transmits its data to the

data miner. In a system with privacy concerns, a data provider may perturb its data first and transmit (only) the perturbed data to the data miner.

Let T be the set of all m original data tuples. Let $d(t_i, t_j)$ be the distance function between two data tuples t_i and t_j. Suppose there are k clusters within T. As is commonly assumed, both the data miner and the data providers know k as pre-knowledge. We use $f(t_i) \in [1, k]$ to denote the cluster label of t_i. The objective of clustering without privacy concern is for the data miner to obtain $f(t_i)$ for all $i \in [1, m]$.

As we mentioned in Section 1, in order to provide comprehensive privacy protection in data clustering systems, our objective is to achieve the following three goals:

- (*Value Privacy*) Minimize the amount of private information disclosed about t_i.
- (*Label Privacy*) Minimize the amount of private information disclosed about $f(t_i)$.
- (*Accuracy*) Enable the data miner to generate a clustering function $f_{\mathrm{R}}(\cdot) : t \rightarrow [1, k]$, such that for all possible values of t_i, $f_{\mathrm{R}}(t_i) = f(t_i)$.

Note that the accuracy goal does not contradict the label privacy goal. Although the data miner can generate the clustering function $f_{\mathrm{R}}(\cdot)$, it cannot derive $f(t_i)$ because due to value privacy, the data miner does not know the value of t_i.

3 Our New Scheme

In this section, we introduce our new scheme that protects both value and label privacy while maintaining the accuracy of clustering rules. Figure 1 depicts the baseline architecture of our new scheme.

Note that due to sociological survey results, different people have different levels of privacy concern on their data [3]. Thus, we introduce a important parameter for each data provider P_i called the *maximum acceptable disclosure level*, denoted by l_i. The maximum acceptable disclosure level measures the level of privacy concerns of each data provider. Generally speaking, if we consider the original data tuples as random

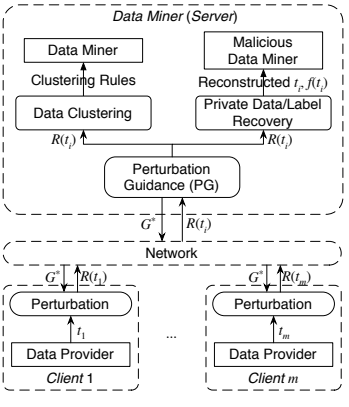


Fig. 1. Our New Scheme

vectors, then l_i is the degree of freedom of the perturbed vector $R(t_i)$, which in most cases is much smaller than the degree of freedom of the original data tuple t_i. The relationship between l_i and the level of privacy disclosure is further analyzed in Section 5.

When a data provider joins the system, it first inquires the data miner about the current *system disclosure level* l^*, which is minimum disclosure level (currently) required by the data miner to maintain an accurate estimation of the clustering rules. The computation of l^* is presented in Section 4. If the data provider cannot accept l^* (i.e., $l^* > l_i$), it can wait for a certain amount of time and try again. As we demonstrate in Section 6, the value of l^* decreases rapidly when the data miner receives more data tuples. Since l_i varies among different data providers, the system disclosure level will soon be acceptable to most data providers.

If a data provider accepts l^*, the data provider then inquires the data miner about the current *perturbation guidance* G^*. Basically speaking, G^* tells the data provider how to project its original data tuple t into an l^*-dimensional subspace, such that 1) the cluster label information is removed from the projected data, and 2) the private information retained in the subspace is the minimum necessary to generate clustering rules. The computation of G^* is presented in Section 4. After receiving G^*, the data provider first checks the validity of G^*, and then compute the perturbed data $R(t)$ based on t and G^*. Details of the validation and perturbation process are also presented in Section 4. The data provider then transmits $R(t)$ to the data miner.

After the data miner receives all data tuples, it can directly use the collected data as input (to any clustering algorithm) to generate clustering rules. As we can see, our scheme is transparent to the clustering algorithm used, and can be integrated into existing systems as middle-ware. There are two key components in our scheme: *perturbation guidance* of the data miner and *perturbation* of the data providers. We introduce these two components in detail in the next section.

4 Basic Components

The basic components of our scheme are 1) the perturbation guidance component of the data miner, which generates the current system disclosure level l^* and the perturbation guidance G^*; and 2) the perturbation component of the data providers, which validates the received G^* and computes $R(t)$ based on t and G^*. In this section, we first introduce some basic notions, and then introduce the design of these two components respectively.

4.1 Basic Notions

Recall that there are m data providers in the system, each of which holds a private data tuple t_i with n attributes $a_1, \ldots, a_n$. We assume that all attributes are categorical. If an attribute is continuous, it can be discretized first. Let d_i be the number of distinct values of a_i. Without loss of generality, we assume that $a_i \in \{1, \ldots, d_i\}$. We can denote a data tuple t_i by a $(d_1 + \cdots + d_n)$-dimensional vector as follows.

$$t_i = \overbrace{0, \ldots, 1, \ldots, 0}^{d_1 \text{ bits for } a_1}, \ldots, \overbrace{0, \ldots, 1, \ldots, 0}^{d_n \text{ bits for } a_n} \tag{1}$$

Within the d_i bits assigned for a_i, the j-th bit is 1 if and only if $a_i = j$. All other bits are 0. For example, a binary attribute $a_i = 0$ has two corresponding bits as $1, 0$.

Let there be $n_0 = d_1 + \cdots + d_n$. Each data tuple t_i can be represented by an n_0-dimensional vector. As such, we can represent the set of all data tuples by an $m \times n_0$ matrix $T = [t_1; \ldots; t_m]$. The i-row of T is the corresponding vector of t_i. We use $\langle T \rangle_{ij}$ to denote the element of T with indices i and j (i.e., the j-th bit of t_i). We denote the transpose of T by T'.

4.2 Perturbation Guidance

As we are considering systems where the data tuples are iteratively fed to the data miner, the data miner needs to maintain a copy of all received (perturbed) data tuples. Let T^* the current matrix of such received data tuples. When a new (perturbed) data tuple is received, it is directly appended to T^*. Our scheme computes l^* and G^* based on T^*. In order to compute them for the first-come data provider, we assume that the initial value of T^* is an $m_0 \times n_0$ matrix ($m_0 \geq k \cdot n_0$), which is consisted of either data tuples collected from privacy-unconcerned data providers or randomly generated data tuples, or a combination of them.

When a new data provider joins the system, the computation of l^* and G^* is consisted of two steps[1]. In the first step, the data miner groups the received (perturbed) data tuples in T^* into k clusters $T_1^*, \ldots, T_k^*$. Let the current clustering function be $f^*(\cdot)$. Let the number of data tuples in T_i^* be m_i^*. Then, the data miner computes the singular value decomposition (SVD) of each cluster as follows.

$$T_i^* = U_i \Sigma_i V_i', \tag{2}$$

where U_i is an $m_i^* \times n_0$ unitary matrix (i.e., $U_i' U_i = I$ where I is the $n_0 \times n_0$ identity matrix), $\Sigma_i = \mathrm{diag}(\sigma_{i1}^*, \ldots, \sigma_{in_0}^*)$ is an $n_0 \times n_0$ diagonal matrix with singular values of T_i^*: $\sigma_{i1}^* \geq \cdots \geq \sigma_{in_0}^*$, and $V_i = [v_{i1}^*, \ldots, v_{in_0}^*]$ is an $n_0 \times n_0$ unitary matrix that contains the right singular vectors of T_i^*. Note that when m is large, it is always possible to incrementally compute the SVD of T_i^* when new data tuples are received.

In the second step, the data miner computes l^* and G^* based on Σ_i^* and V_i^*. In particular, l^* is the minimum integer in $[1, n_0]$ that satisfies $\forall i \in [1, k]$,

$$\sigma_{i(l^*+1)}^* \leq \mu \sigma_{i1}^*, \tag{3}$$

where $\mu \in [0, 1]$ is a parameter pre-determined by the data miner. A data miner that can tolerate a relatively lower level of accuracy can choose a large μ to reduce l^*. A data miner that requires a higher level of accuracy can maintain a small μ to ensure accurate clustering rules. In order to choose a cut-off μ that reduces l^* rapidly while maintaining accurate clustering rules, a textbook heuristic is to set $\mu = 0.15$.

Given l^*, the perturbation guidance G^* is a set that contains: 1) the clustering function $f^*(\cdot)$, 2) l^*-dimensional vectors $s_1^*, \ldots, s_k^*$, and 3) $n_0 \times l^*$ matrices $V_1^*, \ldots, V_k^*$. If k-means clustering algorithm is used by the data miner, the clustering function $f^*(\cdot)$ is

[1] Note that due to efficiency concern, such computation may only take place (i.e., l^* and G^* be updated) once several data tuples are received.

consisted of the center points of the k clusters. Each vector s_i^* ($i \in [1,k]$) is consisted of the largest l^* singular values of T_i^*. Each matrix V_i^* ($i \in [1,k]$) is consisted of the first k right singular vectors of T_i^* (i.e., the first l^* columns of V_i). That is,

$$s_i^* = [\sigma_{i1}^*, \ldots, \sigma_{il^*}^*]. \tag{4}$$

$$V_i^* = [v_{i1}^*, \ldots, v_{il^*}^*]. \tag{5}$$

After computing s_i^* and V_i^*, the data miner transmits $G = \langle f^*, s_1^*, \ldots, s_k^*, V_1^*, \ldots, V_k^* \rangle$ to the data provider.

4.3 Perturbation

After accepting l^* and receiving G^*, a data provider needs to 1) check the validity of G^*, and 2) compute the perturbed data tuple $R(t)$. The validation process is simple: P_i only needs to check if every received V_i^* in G^* is a $n_0 \times l^*$ matrix that satisfies $V_i^{*\prime} V_i^* = I$, where I is the $l^* \times l^*$ identity matrix. If so, then the received G^* is valid.

After the validation process, the data provider computes $R(t)$ as follows. First, the data provider determines the cluster label of its data by computing $f^*(t)$. After that, it computes k ($l^* \times l^*$) diagonal matrices Λ_i for $i \in [1,k]$, such that $\forall j \in [1,l^*]$, $\langle \Lambda_i \rangle_{jj} = \langle s_i^* \rangle_j$. Based on $\Lambda_1, \ldots, \Lambda_k$ and $V_1^*, \ldots, V_k^*$, the data provider computes an intermediate (perturbed) vector $\tilde{t}$ as follows.

$$\tilde{t} = \begin{cases} tV_{f^*(t)}^* \Lambda_{f^*(t)}^{-1} \Lambda_1 V_1^{*\prime}, \text{ with probability of } 1/k, \\ \ldots, \qquad\qquad \ldots, \\ tV_{f^*(t)}^* \Lambda_{f^*(t)}^{-1} \Lambda_k V_k^{*\prime}, \text{ with probability of } 1/k, \end{cases} \tag{6}$$

where $\Lambda_{f^*(t)}^{-1}$ is the inverse matrix of $\Lambda_{f^*(t)}$. Note that since for all $i \in [1,k]$, Λ_i is always a diagonal matrix, the inverse of Λ_i always exists. Also note that the inverse of V_i^* does not exist because all V_i^* ($i \in [1,k]$) are rank-l^* matrices with $l^* < n_0$. Thus, it is impossible to recover t from $\tilde{t}$.

As we can see, the elements in $\tilde{t}$ are real values. Thus, we need an additional step to transform $\tilde{t}$ to a binary vector. In particular, for every $j \in [1, n_0]$, we have

$$\langle R(t) \rangle_j = \begin{cases} 1, \text{ if } r \leq \langle \tilde{t} \rangle_j^2, \\ 0, \text{ otherwise.} \end{cases} \tag{7}$$

where $\langle \tilde{t} \rangle_j$ is the j-th element of $\tilde{t}$, and r is generated uniformly at random from $[0,1]$.

5 Theoretical Analysis

We now analyze the performance of our scheme. We define performance measures on 1) the amount of disclosure on value privacy (i.e., the disclosure of original data tuples), 2) the amount of disclosure on label privacy (i.e., the disclosure of individual cluster labels), and 3) the error of clustering rules built on the perturbed data. We derive bounds on these measures, in order to provide guidelines for system administrators to set parameters in practical systems.

5.1 Value Privacy

Recall that our scheme allows each data provider P_i to choose its individual maximum acceptable disclosure level l_i. Thus, we define the level of value-privacy disclosure based on l_i of individual data providers. Note that we need to consider all possible G^* that can pass the validity test. In particular, we use a information-theoretic measure. Let $H(t_i)$ be the information entropy of t_i (i.e., the amount of information in t_i) and $I(t_i; R(t_i))$ be the mutual information between t_i and $R(t_i)$ (i.e., the amount of information about t_i that can be disclosed by $R(t_i)$). Please refer to [1] for the details of information theory.

Definition 1. *The degree of value-privacy disclosure $\mathcal{P}_{\mathrm{v}}(l_i)$ is the maximum expected percentage of information disclosed by $R(t_i)$ when P_i computes $R(t_i)$ based on a system disclosure level $l^* \leq l_i$ and an arbitrary perturbation guidance G^* that passes the validity test. That is,*

$$\mathcal{P}_{\mathrm{v}}(l_i) = \max_{G^*, l^* \leq l_i} \frac{I(t_i; R(t_i))}{H(t_i)}. \tag{8}$$

As we can see from the definition, the larger $\mathcal{P}_{\mathrm{v}}(l_i)$ is, the more value-privacy-related information about P_i is disclosed to the data miner. We derive an upper bound on $\mathcal{P}_{\mathrm{v}}(l_i)$ as follows.

Theorem 1. *When m is sufficiently large, we have*

$$\mathcal{P}_{\mathrm{v}}(l_i) \leq \frac{\rho_1^2 + \cdots + \rho_{l_i}^2}{kmn}, \tag{9}$$

where ρ_j is the j-th singular value of T.

Due to space limit, please refer to [6] for the proof of this theorem.

5.2 Label Privacy

Definition 2. *The degree of label-privacy disclosure $\mathcal{P}_{\mathrm{c}}$ is the probability that $R(t_i)$ and $R(t_j)$ belong to the same cluster given t_i and t_j in the same cluster. That is,*

$$\mathcal{P}_{\mathrm{c}} = \Pr\{f_{\mathrm{R}}(R(t_i)) = f_{\mathrm{R}}(R(t_j)) | f(t_i) = f(t_j)\}. \tag{10}$$

For the sake of simplicity, we consider the problem in a 2-dimensional setting. That is, we consider the cases where the clustering algorithm groups data tuples based on two (optimal) attributes (chosen from the n attributes) that have the most discrepancy between different clusters. We use $\mathcal{P}_{\mathrm{c}}(2)$ to denote the value of $\mathcal{P}_{\mathrm{c}}$ in this 2-dimensional setting.

Theorem 2. *When m is sufficiently large, we have $\mathcal{P}_{\mathrm{c}}(2) = 1/k$.*

Please refer to [6] for the proof of this theorem.

5.3 Accuracy

Recall that $f_{\mathrm{R}}(\cdot)$ is the clustering function generated from the perturbed data. For every possible value t_{S}, we measure the error on the mined clustering rules about t_{S} as

$$\mathcal{E}(t_{\mathrm{S}}) = P(t_{\mathrm{S}}) \cdot \Pr(f_{\mathrm{R}}(t_{\mathrm{S}}) \neq f(t_{\mathrm{S}})). \tag{11}$$

Definition 3. *The degree of error $\mathcal{E}$ is defined as the maximum of $\mathcal{E}(t_{\mathrm{S}})$ on all possible values of t_{S}.*

Theorem 3. *When m is sufficiently large, we have*

$$\mathcal{E} \leq \max_{j \in [1,k]} \frac{2\mu\sigma_{j1}}{m}, \tag{12}$$

where σ_{j1} is the largest singular value of T_j.

Please refer to [6] for the proof of this theorem.

6 Experimental Evaluation

We conduct our experiments on the congressional voting records database from the UCI machine learning repository [4]. The data set includes 16 key votes (0-no or 1-yes) for each of the 435 United Stated House Representatives in 1984. The task is to cluster the records into two clusters: republicans and democrats. There are 61.38% democrats and 38.62% republications in the data set. The data set contains 392 missing votes, which we fill with values generated uniformly at random from $\{0, 1\}$. We use k-means clustering algorithm in our evaluation.

Since each data tuple contains 16 binary elements, we represent each data tuple by a 32-dimensional binary vector. We apply our scheme on the data set when μ varies from 0.1 to 0.9. The initial value of T^* is consisted of 32 uniformly generated tuples and 32 tuples randomly chosen from T. The data miner updates the perturbation guidance once every 40 data tuples are received.

Figure 2 shows the performance of our scheme on value privacy, label privacy, and accuracy. We evaluate the level of value privacy and label privacy by the disclosure measures defined in Section 5. In order to demonstrate the accuracy of our scheme intuitively, we measure accuracy by the percentage of original data tuples that can be correctly clustered by the clustering rules generated from the perturbed data.

As we can see from the results, our scheme can achieve fairly high level of accuracy (over 90% when $\mu \leq 0.6$) while maintaining a low level of disclosure on both value privacy (less than 0.45 when $\mu \geq 0.3$) and label privacy (less than 0.65 when $\mu \geq 0.3$). Note that since there are 61.38% democrats in the data set, the lowest possible level of label disclosure is 0.6138. As we can see, the disclosure level of our scheme is very close to this lower bound when $\mu \geq 0.3$.

In order to demonstrate the change of l^* when the data miner receives more (perturbed) data tuples, we show the change of l^* with $|T^*|$ in Figure 3 when $\mu = 0.15$. As we can see, the value of l^* decreases fairly quickly (e.g., reduced to 7 when 50 data tuples are received) when the data miner receives more data tuples.

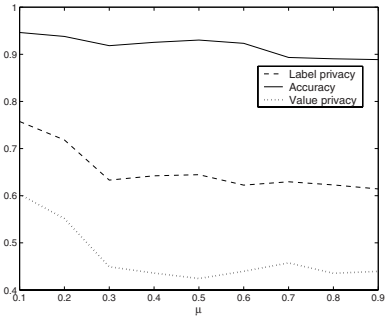


Fig. 2. Performance on Value Privacy, Label Privacy, and Accuracy

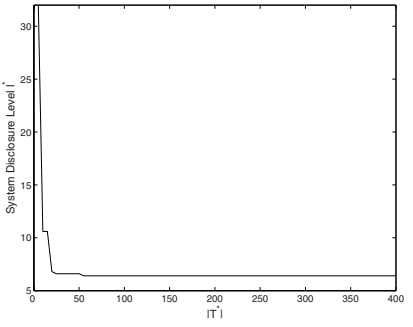


Fig. 3. Change of System Disclosure Level with Number of Received Data Tuples

7 Final Remarks

In this paper, we address the comprehensive protection of privacy in data clustering. Compared with previous work, we identify a new privacy concern in data clustering which is the privacy of individual cluster labels. In order to achieve comprehensive privacy protection, we propose a privacy-preserving clustering scheme that can simultaneously protect the privacy of original data and individual cluster labels. In particular, our scheme allows each data provider to choose a different level of maximum acceptable privacy disclosure level that reflects its individual privacy concern. Our scheme also allows the data miner to distribute perturbation guidance to the data providers. Using this intelligence, the data providers perturb their data tuples and transmit the perturbed tuples to the data miner. As a result, our scheme can achieve both value and label privacy while maintaining the accuracy of clustering rules. We demonstrate the performance of our scheme by theoretical bounds and experimental evaluation.

Our work is preliminary and many extensions can be made. We are currently investigating privacy concerns beyond original data being mined in other data mining problems. We would also like to investigate the integration of our scheme with cryptographic approach in Server-to-Server (S2S) systems.

References

1. T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.
2. J. Han and M. Kamber. *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2001.
3. IBM. IBM-Harris multinational customer privacy survey. Technical report, 1999.
4. D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI repository of machine learning databases, 1998.
5. S. R. M. Oliveira and O. R. Zaïane. Privacy preserving clustering by data transformation. In *Proc. 18th Brazilian Symposium on Databases (SBBD)*, pages 304–318, 2003.
6. N. Zhang. Towards the comprehensive protection of private information in data clustering. Technical Report CSE-2007-2, University of Texas at Arlington, 2007.
7. N. Zhang, S. Wang, and W. Zhao. A new scheme on privacy-preserving data classification. In *KDD*, pages 374–383, 2005.

A Novel Spatial Clustering with Obstacles Constraints Based on Particle Swarm Optimization and K-Medoids

Xueping Zhang [1, 2, 3], Jiayao Wang [2], Mingguang Wu [2], and Yi Cheng[2]

[1] School of Information Science and Engineering, Henan University of Technology, Zhengzhou 450052, China
[2] School of Surveying and Mapping, PLA Information Engineering University, Zhengzhou 450052, China
[3] Geomatics and Applications Laboratory, Liaoning Technical University, Fuxin 123000, China
zhang_xpcn@yahoo.com.cn

Abstract. In this paper, we discuss the problem of spatial clustering with obstacles constraints and propose a novel spatial clustering method based on PSO and K-Medoids, called PKSCOC, which aims to cluster spatial data with obstacles constraints. The PKSCOC algorithm can not only give attention to higher local constringency speed and stronger global optimum search, but also get down to the obstacles constraints and practicalities of spatial clustering. The results on real datasets show that the PKSCOC algorithm performs better than the IKSCOC algorithm in terms of quantization error.

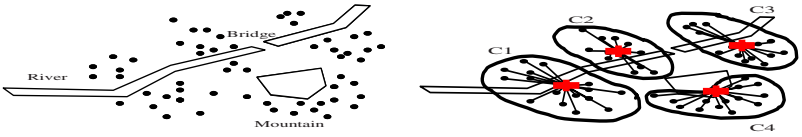
Keywords: Spatial Clustering, Particle Swarm Optimization, K-Medoids Algorithm, Obstacles Constraints.

1 Introduction

Spatial clustering with constraints is an important topic in Spatial Data Mining (SDM). Spatial clustering with constraints has two kinds of forms [1]. One kind is Spatial Clustering with Obstacles Constraints (SCOC).An obstacle is a physical object that obstructs the reach ability among the data objects, such as bridge, river, and highway etc. whose impact on the result should be considered in the clustering process of large spatial data. As an example, Fig.1 shows clustering data objects in relation to their neighbors as well as the physical obstacle constraints. Ignoring the constraints leads to incorrect interpretation of the correlation among data points. The other kind is spatial clustering with handling operational constraints [2], it consider some operation limiting conditions in the clustering process. In this paper, we mainly discuss SCOC. Handling these obstacles constraints can lead to effective and fruitful data mining by capturing application semantics [3-8].

Since K.H.Tung put forward a clustering question COE (Clustering with Obstacles Entities) [3] in 2000, a new studying direction in the field of clustering research have been opened up. To the best of our knowledge, only three clustering algorithms for clustering spatial data with obstacles constraints have been proposed very recently:

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1105–1113, 2007.



(a) Data objects and constraints (b) Clusters ignoring constraints

Fig. 1. Clustering data objects with obstacles constraints

COD-CLARANS [3] based on the Partitioning approach of CLARANS, AUTOCLUST+ [4] based on the Graph partitioning method of AUTOCLUST, and DBCluC [5]-[8] based on the Density-based algorithm. Although these algorithms can deal with some obstacles in the clustering process, many questions exist in them. COD-CLARANS algorithm inherits the shortcoming of CLARANS algorithm, which only gives attention to local constringency. AUTOCLUST+ algorithm inherits the limitation of AUTOCLUST algorithm, which builds a Delaunay structure to cluster data points with obstacles costly and is unfit for a large number of data. DBCluC inherits the shortcoming of DBSCAN algorithm, which cannot run in large high dimensional data sets etc. We proposed GKSCOC (Genetic K-Medoids Spatial Clustering with Obstacles Constraints) based on Genetic algorithms (GAs) and IKSCOC (Improved K-Medoids Spatial Clustering with Obstacles Constraints) in the literature [9]. The results of the experiments on real datasets show that it is better than IKSCOC. But the drawback of GKSCOC is a comparatively slower speed in clustering.

Recently, Particle Swarm Optimization (PSO) has been applied to data clustering [10-13]. This paper explores the applicability of PSO to cluster spatial data with obstacles constraints. We develop a novel Spatial Clustering with Obstacles Constraints based on PSO and K-Medoids, called PKSCOC. The PKSCOC can not only give attention to higher local constringency speed and stronger global optimum search, but also get down to the obstacles constraints and practicalities of spatial clustering. The results on real datasets show that the PKSCOC algorithm performs better than the IKSCOC algorithm in terms of quantization error.

The remainder of the paper is organized as follows. SCOC based on K-Medoids, called KSCOC, is discussed in Section 2. Section 3 introduces PSO. Section 4 presents SCOC based on PSO and K-Medoids, called PKSCOC. The performances of PKSCOC on datasets in comparison with the IKSCOC are showed in Section 5, and Section 6 concludes the paper.

2 SCOC Based on K-Medoids

Partitioning-base algorithm divides n objects into $k(k < n)$ parts, and each part represents one cluster. There are three typical types of partitioning-based algorithm: K-Means, K-Medoids and CLARANS. K-Means takes the average value of a cluster as the cluster centre. While adopting this algorithm, a cluster center possibly just falls on the obstacle (Fig.2), and it cannot be implemented in reality. On the other hand, K-Medoids takes the most central object of a cluster as the cluster centre, and the

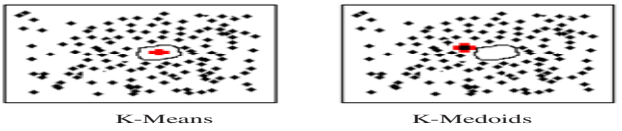


Fig. 2. K-Means vs. K-Medoids

cluster center cannot fall on the obstacle. In view of this, K-Medoids algorithm is adopted in this paper. CLARANS algorithm can be adopted to handle large number of data sets.

2.1 Motivating Concepts

To derive a more efficient algorithm for SCOC, the following definitions are first introduced.

Definition 1 (Visibility graph). Given a set of m obstacle, $O = (o_1, o_2, \ldots, o_m)$,the visibility graph is a graph $VG = (V, E)$ such that each vertex of the obstacles has a corresponding node in V , and two nodes v_1 and v_2 in V are joined by an edge in E if and only if the corresponding vertices they represent are visible to each other.

To generate VG , we use VPIA (VGRAPH Point Incorporation Algorithm) as presented in [14].

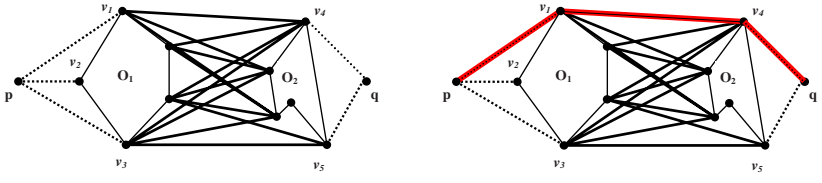


Fig. 3. Visibility graph and Obstructed distance

Definition 2 (Obstructed distance). Given point p and point q , the obstructed distance $d_o(p,q)$ is defined as the length of the shortest Euclidean path between two points p and q without cutting through any obstacles.

We can use Dijkstra Algorithm to compute obstructed distance. The simulation result is in Fig.3 and the red solid line represents the obstructed distance we got.

2.2 Improved SCOC Based on K-Medoids

K-Medoids algorithm selects the most central object of a cluster as the cluster centre. The clustering quality is estimated by an object function. Square-error function is adopted here, and its definition can be defined as:

$$E = \sum_{j=1}^{N_C} \sum_{p \in C_j} (d(p, m_j))^2 \tag{1}$$

where N_c is the number of cluster C_j, m_j is the cluster centre of cluster C_j, $d(p,q)$ is the direct Euclidean distance between the two points p and q.

To handle obstacles constraints, accordingly, criterion function for estimating the quality of spatial clustering with obstacles constraints can be revised as:

$$E_o = \sum_{j=1}^{N_c} \sum_{p \in C_j} (d_o(p, m_j))^2 \tag{2}$$

where $d_o(p,q)$ is the obstructed distance between point p and point q, which is defined by the shortest distance between the two points p and q which cannot be cut off by any obstacle.

As we know, the cost of computing obstructed distance is much more than direct Euclidean distance's, so computing E_o is much more costly than computing E. In order to improve the efficiency of whole algorithm, the method of Improved KSCOC (IKSCOC) is adopted as follows [9].

1. Select N_c objects to be cluster centers at random;
2. Distribute remain objects to the nearest cluster center;
3. Calculate E_o according to equation (2);
4. Do {let current $E = E_o$;
5. Select a not centering point to replace the cluster center m_j randomly;
6. Distribute objects to the nearest center;
7. Calculate E according to equation (1);
8. If $E >$ current E, go to 5;
9. Calculate E_o;
10. If $E_o <$ current E, form new cluster centers;
11. } While (E_o changed).

While IKSCOC still inherits two shortcomings because it is based on standard partitioning algorithm. One shortcoming is that selecting initial value randomly may cause different results of the spatial clustering and even have no solution. The other is that it only gives attention to local constringency and is sensitive to an outlier.

3 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) is a population-based optimization method first proposed by Kennedy and Eberhart [15, 16]. In order to find an optimal or near-optimal solution to the problem, PSO updates the current generation of particles (each particle is a candidate solution to the problem) using the information about the best solution obtained by each particle and the entire population. In the context of PSO, a swarm refers to a number of potential solutions to the optimization problem, where each potential solution is referred to as a particle. The aim of the PSO is to find the

particle position that results in the best evaluation of a given fitness (objective) function. Each particle has a set of attributes: current velocity, current position, the best position discovered by the particle so far and, the best position discovered by the particle and its neighbors so far. The user can define the size of the neighborhood. There is one version of PSO called global PSO in which all the particles are considered to be neighbors of each other. All particles start with randomly initialized velocities and positions. Then the $_n{}^{th}$ component of the new velocity and the new position for the $_i{}^{th}$ particle are updated by using the following equations:

$$V_{i,n}(t+1) = w * V_{i,n}(t) + c_1 * rand() * (G_i(t) - X_{i,n}(t)) + c_2 * rand() * (l_{i,n}(t) - X_{i,n}(t)) \quad (3)$$

$$X_{i,n}(t+1) = X_{i,n}(t) + V_{i,n}(t+1) \quad (4)$$

where w is the inertia weight, c_1 and c_2 are positive constant parameters, and $Rand()$ is a random function with the range [0, 1], G_i is the best particle found so far within the neighbors and $l_{i,n}$ is the best position discovered so far by the corresponding particle. Velocity magnitudes are often clipped to a predetermined maximum value, $V_{\max}$ .The PSO is usually executed with repeated application of equations (3) and (4) until the specified number of iterations haves been exceeded. Alternatively, the algorithm can be terminated when the velocity updates are close to zero over a number of iterations.

PSO is effective in nonlinear optimization problems and it is easy to implement. In addition, only few input parameters need to be adjusted in PSO. Because the update process in PSO is based on simple equations, PSO can be efficiently used on large data sets. A disadvantage of the global PSO is that it tends to be trapped in a local optimum under some initialization conditions [17].

4 Spatial Clustering with Obstacles Constraints Based on PSO and K-Medoids

In order to overcome the disadvantage of partitioning approach which only gives attention to local constringency, and keep the advantage of PSO which has stronger global optimum search at the same time [10], we propose a novel Spatial Clustering with Obstacles Constraints based on PSO and K-Medoids (PKSCOC).

This section first introduces the PSO Clustering in section 4.1, and then presents the PKSCOC algorithm in section 4.2.

4.1 PSO Clustering

In the context of clustering, single particle represents the N_C cluster centroid. That is, each particle x_i is constructed as follows:

$$x_i = (m_{i1}, ..., m_{ij}, ..., m_{iN_d}) \quad (5)$$

where N_d refers to the input dimension, m_{ij} refers to the j^{th} cluster centroid of the i^{th} particle in cluster C_{ij} . Therefore, a swarm represents a number of candidate c1usterings for the current spatial data.

The objective function is used to assign a fitness value to each individual in the population. Therefore, it needs to be designed so that an individual with a high fitness represents a better solution to the problem than an individual with a lower fitness. Here, objective function is defined as follows:

$$f(x_i) = \frac{1}{J_i} \tag{6}$$

$$J_j = \sum_{j=1}^{N_c} \sum_{p \in C_{ij}} d_o(p, m_j) \tag{7}$$

4.2 Spatial Clustering with Obstacles Constraints Based on PSO and K-Medoids

Using the standard gbest PSO, Spatial Clustering with Obstacles Constraints based on PSO and K-Medoids (PKSCOC), which is similar to the K-means PSO hybrid as presented in [11],is adopted as follows.

1. Execute the IKSCOC algorithm to initialize one particle to contain N_C selected cluster centroids;
2. Initialize the other particles of the swarm to contain N_c selected cluster centroids at random;
3. For $t = 1$ to $t_{\max}$ do {
4. For each particle i do {
5. For each object p do {
6. Calculate $d_o(p, m_{ij})$;
7. Assign object p to cluster C_{ij} such that $d_o(p, m_{ij}) = \min \forall c = 1, \ldots, N_d \{d_o(p, m_{ic})\}$;
8. Calculate the fitness according to equation (6) ;}}
9. Update $gBest$ and $pBest_i$;
10. Update the cluster centroids according to equation (3) and equation (4);
11 If $\|v\| \leq \varepsilon$, terminate;
12. Optimize new individuals using the IKSCOC algorithm ;}

where $t_{\max}$ is the maximum number of iteration, ε is the minimum velocity. STEP 1 is to overcome the disadvantage of the global PSO which tends to be trapped in a local optimum under some initialization conditions. STEP 12 is to improve the local constringency speed of the global PSO.

The population-based search of the PKSCOC algorithm reduces the effect that initial conditions has, as opposed to the IKSCOC algorithm; the search starts from multiple positions in parallel. Section 5 shows that the PKSCOC algorithm performs better than the IKSCOC algorithm in terms of quantization error.

5 Results and Discussion

This section presents experimental results on synthetic and real datasets. All experiments were run on a 2.4GHz PC with 512M memory. We have made experiments separately by K-Medoids, IKSCOC and PKSCOC. The number of particles $n = 50, w = 0.72, c_1 = c_2 = 2, V_{\max} = 0.4, t_{\max} = 100, \omega = 0.001$.

Fig.4 shows the results on synthetic Dataset1. Fig.4 (a) shows the original data with simple obstacles. Fig.4 (b) shows the results of 4 clusters found by K-Medoids without considering obstacles constraints. Fig.4(c) shows 4 clusters found by IKSCOC. Fig.4 (d) shows 4 clusters found by PKSCOC. Obviously, the results of the clustering illustrated in Fig.4(c) and Fig.4 (d) both have better practicalities than that in Fig.4 (b), and the one in Fig.4 (d) is superior to the one in Fig.4 (c).

Fig.5 shows the results on synthetic Dataset2. Fig.5 (a) shows the original data with various obstacles. Fig.5 (b) shows 4 clusters found by K-Medoids. Fig.5 (c) shows 4 clusters found by PKSCOC. Obviously, the result of the clustering illustrated in Fig.5(c) has better practicalities than the one in Fig.5 (b).

Fig.6 shows the results on real Dataset3. Fig.6 (a) shows the original real data with obstacles. Fig.6 (b) shows 4 clusters found by K-Medoids. Fig.6 (c) shows 4 clusters found by PKSCOC. The one in Fig.6 (c) is superior to the one in Fig.6 (b), obviously.

Fig.7 is the value of J showed in every experiment on Dataset1. It is showed that IKSCOC is sensitive to initial value and it constringes in different extremely local optimum points by starting at different initial value while PKSCOC constringes nearly in the same optimum points each time. Therefore, we can draw the conclusion that PKSCOC has stronger global constringent ability comparing with IKSCOC; and PKSCOC has not only considered high local constringent speed but also kept good global constringent characteristic.

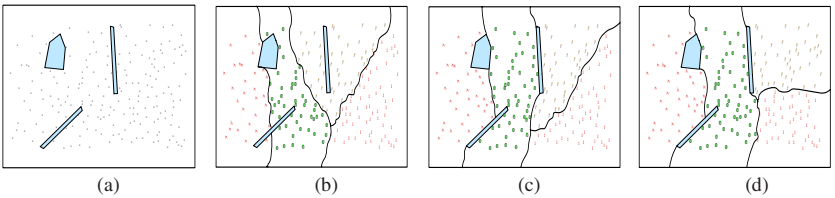


Fig. 4. Clustering dataset Dataset1

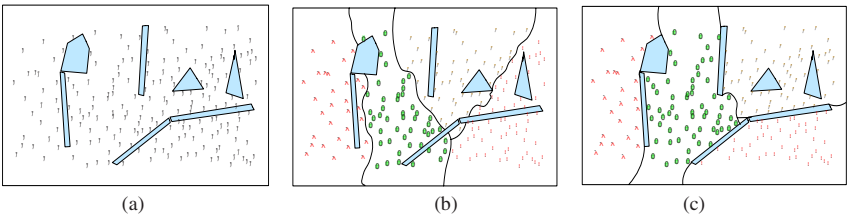


Fig. 5. Clustering dataset Dataset2

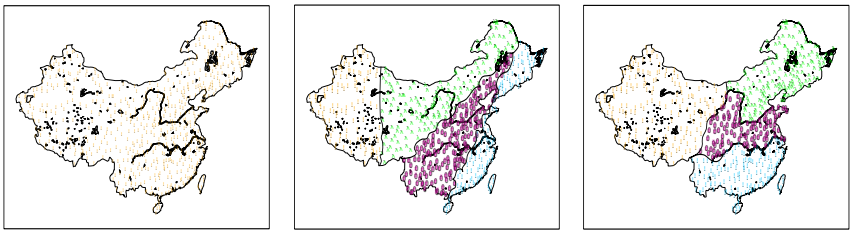


Fig. 6. Clustering dataset Dataset3

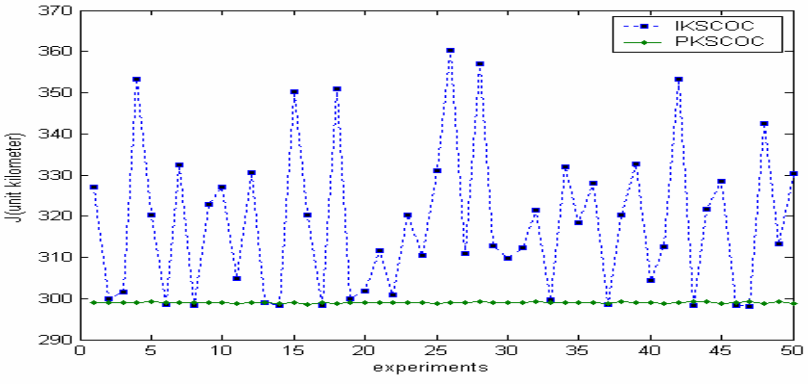


Fig. 7. PKSCOC vs. IKSCOC

6 Conclusions

Spatial clustering has been an active research area in the data mining community. Classic clustering algorithms have ignored the fact that many constraints exit in the real world and could affect the effectiveness of clustering result. In this paper, we discussed the problem of spatial clustering with obstacles constraints and propose a novel PKSCOC based on PSO and K-Medoids. The comparison proves that our method can not only give attention to higher local constringency speed and stronger global optimum search, but also get down to the obstacles constraints and practicalities of spatial clustering. The results of the experiments on real datasets show that it is better than IKSCOC. And its achievements will have more practical value and extensive application prospect.

Acknowledgments. This work is partially supported by the Natural Sciences Fund Council of China (Number: 40471115), the Natural Sciences Fund of Henan (Number:0511011000, 0624220081) and the Open Research Fund Program of the Geomatics and Applications Laboratory, Liaoning Technical University (Number: 2004010).

References

1. A.K.H.Tung, J.Han, L.V.S.Lakshmanan, and R.T.Ng.: Constraint-Based Clustering in Large Databases. In Proceedings of the International Conference on Database Theory (ICDT'01) [C], London, U.K. (2001) 405-419
2. A.K.H.Tung, R.T.Ng, L.V.S.Lakshmanan, and J.Han.: Geo-spatial Clustering with User-Specified Constraints. In Proceedings of the International Workshop on Multimedia Data Mining (MDM/KDD 2000) [C], Boston USA (2000) 1-7
3. A.K.H.Tung, J.Hou, and J.Han.: Spatial Clustering in the Presence of Obstacles. In Proceedings of International Conference on Data Engineering (ICDE'01) [C], Heidelberg Germany (2001) 359-367
4. V.Estivill-Castro and I.J.Lee.: AUTOCLUST+: Automatic Clustering of Point-Data Sets in the Presence of Obstacles. In Proceedings of the International Workshop on Temporal, Spatial and Spatial-Temporal Data Mining [C], Lyon France (2000) 133-146
5. O.R.Zaïane and C. H.Lee.: Clustering Spatial Data When Facing Physical Constraints. In Proceedings of the IEEE International Conference on Data Mining (ICDM'02) [C], Maebashi City Japan (2002) 737-740
6. X.Wang and H.J.Hamilton: DBRS: A Density-Based Spatial Clustering Method with Random Sampling. In Proceedings of the 7th PAKDD [C], Seoul Korea (2003) 563- 575
7. X.Wang, C.Rostoker and H.J.Hamilton: DBRS+: Density-Based Spatial Clustering in the Presence of Obstacles and Facilitators. Ftp.cs.uregina.ca/Research/Techreports/2004-09.pdf, (2004)
8. X.Wang and H.J.Hamilton: Gen and SynGeoDataGen Data Generators for Obstacle Facilitator Constrained Clustering. Ftp.cs.uregina.ca/Research/Techreports/2004-08.pdf, 2004.
9. Xueping Zhang, Jiayao Wang, Fang Wu, Zhongshan Fan and Xiaoqing Li: A Novel Spatial Clustering with Obstacles Constraints Based on Genetic Algorithms and K-Medoids. In Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA 2006) [C], Jinan Shandong China (2006) 605-610
10. Xiang Xiao Dow, E.R. Eberhart, R. Miled, Z.B. Oppelt, R.J. : Gene Clustering Using Self-Organizing Maps and Particle Swarm Optimization. In Proceedings of the International Conference on Parallel and Distributed Processing Symposium (IPDPS) [C], (2003)
11. Van der Merwe DW,Engelbrecht A P.: Data Clustering Using Particle Swarm Optimization. In Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003) [C], (2003) 215-220
12. MGH Omran: Particle Swarm Optimization Methods for Pattern Recognition and Image Processing. Ph.D. thesis, University of Pretoria, (2005)
13. Xiaohui Cui,Potok T.E.,Palathingal P.: Document clustering using particle swarm optimization. In Proceedings of IEEE on Swarm Intelligence Symposium (SIS 2005) [C], (2005)185-191
14. Alade Tokuta: Extending the VGRAPH Algorithm for Robot Path Planning. http://wscg.zcu.cz/wscg98/papers98/Tokuta_98.pdf
15. Russ C. Eberhart and J. Kennedy.: A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya Japan (1995) 39-43
16. J. Kennedy and R. C. Eberhart: Particle Swarm Optimization. In Proceedings of IEEE International Conference on Neural Networks, volume IV, Perth Australia (1995)1942-1948
17. Frans van den Bergh: An Analysis of Particle Swarm Optimizers. Ph.D. thesis, University of Pretoria, (2001)

Online Rare Events Detection

Jun Hua Zhao, Xue Li, and Zhao Yang Dong

School of ITEE, The Univ of Queensland, St. Lucia, QLD 4072, Australia
{zhao,xueli,zdong}@itee.uq.edu.au

Abstract. Rare events detection is regarded as an imbalanced classification problem, which attempts to detect the events with high impact but low probability. Rare events detection has many applications such as network intrusion detection and credit fraud detection. In this paper we propose a novel online algorithm for rare events detection. Different from traditional accuracy-oriented approaches, our approach employs a number of hypothesis tests to perform the cost/benefit analysis. Our approach can handle online data with unbounded data volume by setting up a proper moving-window size and a forgetting factor. A comprehensive theoretical proof of our algorithm is given. We also conduct the experiments that achieve significant improvements compared with the most relevant algorithms based on publicly available real-world datasets.

1 Introduction

Rare events detection is a challenging problem frequently encountered in data mining research. It can be characterized as an imbalanced classification problem. Many important real-world problems fall into this category, such as network intrusion detection [1], and price spike forecasting in electricity markets [2]. These problems are usually of severe class imbalance, i.e. the occurrence frequency or probability of one class label is significantly higher than the others. Classification algorithms usually show a strong bias against rare events. This makes rare events very difficult to predict, although they are highly important.

Detecting rare events on online data has two major requirements: (i) the algorithm should not have any bias against the rare event. Accurate classification of the rare event is the major objective. (ii) The algorithm should be an online algorithm which requires only a single pass over data. This is critical for the algorithm to handle potentially unlimited data volume. Unfortunately, there is no method satisfying both of these requirements according to our knowledge.

In this paper, we propose a novel online algorithm, which is able to classify rare events on online data. Our approach is a cost/benefit sensitive algorithm. Cost sensitive analysis is widely accepted as a reasonable approach to evaluate imbalanced classifiers in many real-world applications [3]. The proposed approach can accurately estimate statistical distributions and calculate the expected benefit of each class label. Our approach chooses the class label with the highest expected benefit, therefore guarantees to obtain the maximum classification benefit with the highest probability.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1114–1121, 2007.

A size-varying moving window and a forgetting factor to smoothly remove old training data are introduced. The window size is dynamically adjusted and we have proven that it theoretically guarantees the accurate density estimation. Our approach is an online algorithm. The experiments on real-world datasets have shown that our approach has consistently outperformed other well-known cost-insensitive online algorithms such as CVFDT [4], online Naïve Bayes [5], Ensemble classifier [6], and Winnow [7].

The rest of this paper is organized as follows: In Section 2, the work related to our research is summarized. In Section 3, the problem of rare event detection is formulated. Evaluation criteria are introduced as well. Section 4 presents theoretical foundation of our approach. In Section 5, extensive tests are conducted on real-world datasets. Section 6 concludes the paper.

2 Related Work

The class imbalance problem [8] has been extensively studied in the context of data mining. The most common evaluation criterion, accuracy/error, is not suitable for a class imbalance situation [8]. Several alternative evaluation criteria and frameworks are therefore introduced, including precision/recall [8] and ROC analysis [9]. These methods do not place more emphasis on common classes, therefore show no bias against rare classes. Different from above methods, cost-sensitive analysis [3] relaxes the assumption that all classification errors have the same cost, and gives more weight to rare classes. Therefore, it is able to evaluate the classifier performance effectively and set a proper classification objective. The absolute/relative lack of data is another primary difficulty of the class imbalance problem. Some sampling methods [10] are proposed to rebalance the class distribution to solve this problem.

There are a few well-known online algorithms currently available: CVFDT [4], Online Naïve Bayes [5], ensemble classifier [6] and Winnow [7]. They update the model by incorporating new data continuously from the data source, and revise the classifier without referring to old data.

3 Problem Formulation

3.1 Basic Problem Formulation

Given a dataset $S = (X_1, y_1)...(X_t, y_t),...(X_n, y_n)$, $1 \le t \le n$, where t represents a time point. $X_t = (x_{t1}, x_{t2},...x_{tm})$ is a m-dimensional vector and $y_t \in \{c_1, c_2...c_w\}$ is the class label assigned to X_t. We assume that an unknown function dependency $f : X \to y$ exists. The objective of classification is to build a model of function dependency f' between observation vector X_t and class label y_t to approximate the underlying relationship f, and to classify the unseen data with the model. In this paper, we focus on binary classification. Given that the occurring probability of one class label is p times higher than the other, a dataset is usually considered having class imbalance when $p \ge 10$. The class label with smaller probability is called the *target event* (or

positive class), while its counterpart is called the *common event* (*negative class*). The positive class and negative class are denoted as c_1 and c_0 respectively. In rare events detection, the data volume can be potentially infinite.

3.2 Evaluation Criteria

A proper evaluation criterion needs to be defined to guide the training process. Here we start with four preliminary definitions:

True positive: *TP = observations_correctly_classified_as_positive*,
False positive: *FP=observations_incorrectly_classified_as_positive,*
True negative: *TN=observations_correctly_classified_as_negative,*
False negative: *FN=observations_incorrectly_classified_as_negative,*

Assuming the cost of misclassifying positive observation is $C(P,N)$ and the cost of misclassifying negative observation is $C(N,P)$ respectively, the total cost of classification is defined as [3],

$$C_{total} = C(P,N) \times FN + C(N,P) \times FP \tag{1}$$

In the context of cost sensitive analysis, the classifier should be designed to minimize C_{total}. The cost metric (1) is a widely used criterion in traditional cost-sensitive learning [3]. However, a slight modification is made to metric (1) in our approach. Let the benefit of correctly classifying positive observation and negative observation be $B(P,P)$ and $B(N,N)$ respectively. The total benefit of classification is defined as,

$$\begin{aligned} B_{total} = {} & B(P,P) \times TP + B(N,N) \times TN \\ & - C(P,N) \times FN - C(N,P) \times FP \end{aligned} \tag{2}$$

The total benefit defined above is actually identical to the cost metric discussed in [11]. This metric is introduced since the commonly used cost-sensitive analysis only considers the cost of the classification error, while $B(P,P)$ and $B(N,N)$ are assumed to be zero. Our approach is designed to maximize the total benefit B_{total} rather than minimizing the cost C_{total}. Note that total benefit (2) is a generalization of the cost metric (1). By setting $B(P,P)=0$ and $B(N,N)=0$, the total benefit (2) will be the same as the cost metric (1). Therefore, the proposed method can also work well with the traditional cost metric (1), which only considers classification costs. This will be justified in the experiments.

4 Statistical Online Cost Sensitive Classification

The proposed algorithm is namely *STOCS* (*STatistical Online Cost Sensitive classification*). It consists of a few hypothesis tests over the Bernoulli distribution. Cost/benefit analysis is incorporated into the decision procedure to determine a proper decision rule. We conduct a theoretical analysis to show that the point estimate of the

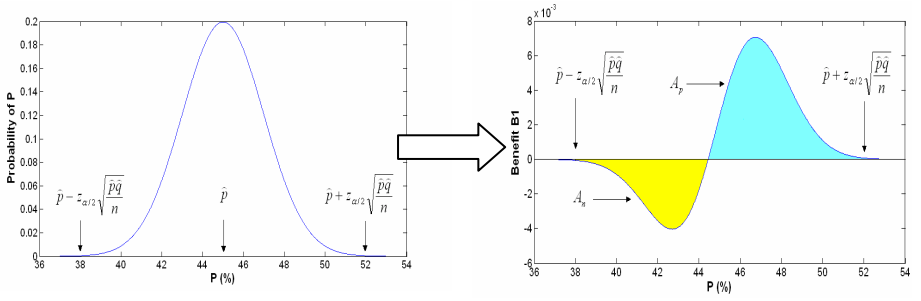


Fig. 1. Calculating the expected benefit

occurring probability of rare events is insufficient for accurate classification. Instead, STOCS employs a confidence interval for density estimation. A size-varying moving window and a forgetting factor are used to make STOCS an online algorithm which only requires a single pass over data.

4.1 Fundamentals of STOCS Approach

Given an observation $X_t = (x_{t1}, x_{t2}, ...x_{tm})$ and a class label $y \in \{c_0, c_1\}$, the posterior probability of the target event c_1 can be calculated using Equation (3) [5].

$$p(c_1 \mid x_{t1}, ..., x_{tm}) = \frac{p(x_{t1} \mid c_1).....p(x_{tm} \mid c_1)p(c_1)}{p(x_{t1})p(x_{t2}).....p(x_{tm})} \tag{3}$$

Assuming that each attribute domain is divided into several intervals and x_{tj} falls into interval I_j, then $p(x_{t1}).....p(x_{tm})$ and $p(x_{t1} \mid c_1).....p(x_{tm} \mid c_1)$ can be replaced by $p(I_1).....p(I_m)$ and $p(I_1 \mid c_1).....p(I_m \mid c_1)$, which are unknown parameters of the underlying distributions $F(X_t)$ and $F(y \mid X_t)$. From Equation (3), obviously $p(c_1 \mid x_{t1}, ..., x_{tm})$ can be replaced by $p(c_1 \mid I_1, ..., I_m)$, which is still an unknown parameter and defined as *target probability*.

As discussed in [3], the cost/benefit analysis should be incorporated into the classifier to handle rare classes. We assume that the benefits of true positive and true negative are $B(P,P)$ and $B(N,N)$, while the costs of false positive and false negative are denoted as $C(N,P)$ and $C(P,N)$. Given an observation X_t and its Euclidean region $(I_1, I_2, ...I_m)$, we assume corresponding target probability $p = p(c_1 \mid I_1, ..., I_m)$ and sample proportion $\hat{p} = \hat{p}(c_1 \mid I_1, I_2, ..., I_m)$ calculated from training data. Now the question is that if we know the exact value of p, how should we determine the class label for X_t? If we classify X_t as c_1, the expected benefit of classification can be calculated as,

$$B_1 = p \times B(P,P) - (1-p) \times C(N,P) \tag{4}$$

Similarly, the expected benefit of classifying X_t as c_0 is:

$$B_0 = (1-p)\times B(N,N) - p\times C(P,N) \tag{5}$$

X_t should be classified as c_1 when $B_1 > B_0$, since $B_1 > B_0$ means that classifying as c_1 has a greater probability to achieve a larger benefit, and vice versa.

Table 1. STOCS algorithm

Inputs: S is the a sequence of observations and their class labels
d is the number of intervals the continuous attributes should be divided
$B(P,P), B(N,N), C(P,N), C(N,P)$ are the classification benefits and costs
α is the confidence level
n_w is the number of Euclidean regions should be checked
Outputs: Class labels of the unlabeled data
Function STOCS
Initialization;
Discretization;
For each observation X_t in S
If (sample size $w > 0$) then
Calculate $\hat{p}$ according to Equation (3);
y_t = Classify ( );
End if
Remove the observation X_{t-w} from the training data, by reducing 1 from the corresponding frequency arrays;
Update the corresponding frequency arrays of X_t;
Set $w = t$ if the window size is large enough;
Record the Euclidean region $(I_1,...I_m)$ of X_t, and update corresponding frequency arrays;
End for
	Inputs: $\hat{p}$ is the sample proportion
	α is the confidence level
	$B(P,P), B(N,N), C(P,N), C(N,P)$ are the classification benefits and costs
	$n(I_j)$ is the number of observations whose jth attribute falling into interval I_j
	w is the size of training data
	X_t is the observation to be classified
	Output: Class label y
	Function Classify
	Calculate B_1 and B_0 according to Equation (6) and (7);
	If ($B_1 \geq B_0$) then
	Return c_1;
	Else
	Return c_0;
	End if

In real-world applications it is usually impossible to know the true value of p, however, we know that p distributes within the z-interval with a probability of $1-\alpha$. Therefore, we will have a very accurate estimation of p, by choosing a very small α, e.g. 0.01. The expected benefit B_1 and B_0 over z-interval of p can be expressed as:

$$B_1 = \int_{lp}^{up} [p\times B(P,P) - (1-p)\times C(N,P)]\times \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(p-\mu)^2}{2\sigma^2}} dp \tag{6}$$

and

$$B_0 = \int_{lp}^{up} [(1-p)\times B(N,N) - p\times C(P,N)]\times \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(p-\mu)^2}{2\sigma^2}} dp \tag{7}$$

where $up = \hat{p} + z_{\alpha/2}\sqrt{\frac{\hat{p}\hat{q}}{n}}$, $lp = \hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}\hat{q}}{n}}$, $u = \hat{p}$ and $\sigma^2 = \frac{\hat{p}\hat{q}}{n}$.

Calculation of expected benefit is demonstrated in Fig.1. The expected benefit can be calculated by integration over the z-interval $[\hat{p} - z_{\alpha/2}\sqrt{\frac{\hat{p}\hat{q}}{n}}, \hat{p} + z_{\alpha/2}\sqrt{\frac{\hat{p}\hat{q}}{n}}]$. It actually equals the positive area A_p minus the negative area A_n in Fig.1.

5 Experiments and Analysis

5.1 Algorithms Selected for Comparison

The performance of STOCS is compared with several well-known online algorithms, including CVFDT [4], online Naïve Bayesian classifier [5], Ensemble classifier [6] and Winnow [7]. Two real-world datasets, Australia electricity price dataset [12] and KDD Cup 99 dataset [13] are used as the benchmarking datasets.

5.2 Experiments on Real-World Datasets

STOCS is tested on two real-world datasets. The first one is the price data of the Australian National Electricity Market (NEM). In the electricity market, the abnormal high electricity price is called *price spike*. The price spikes have very high influence (they can be hundreds of times greater than normal prices), hence are of high interest to the market participants. In our experiment, we consider the prices greater than 75$/MWh as price spikes, which is proven to be appropriate by a statistical method [2]. The price spike forecasting problem is a typical rare events detection problem on online data. Firstly, the new market price data is continuously generated every 5 minutes. Hence a fast algorithm is required to do training and classification. Secondly, price spikes have a small occurrence probability (<2%). Therefore, the electricity price data is suitable to be a benchmark dataset for our algorithm. The price data used in our experiment are downloaded from the website of NEM [12].

In the experiments, parameters of STOCS are set as: $d = 20, \alpha = 0.01, n_w = 5000$. In price spike forecasting, true negative is usually considered having no benefit while false negative is considered having no cost. Therefore, in this experiment we only consider the benefit of true positive and cost of false positive. The total benefits of STOCS and other online algorithms are plotted in Fig. 2. As observed, the total benefits of STOCS and other online-algorithms increase as $B(P,P)/C(N,P)$ increases. The proposed STOCS consistently outperforms all its rivals. Moreover, the performance of STOCS is significantly better than its alternatives when $B(P,P)/C(N,P)$ is large. Because a larger $B(P,P)/C(N,P)$ implies that correctly classifying rare events will lead to larger benefit, i.e. the rare events are more important to the users. Therefore, the results of Fig. 2 demonstrate that STOCS is superior to other methods in dealing with rare events detection.

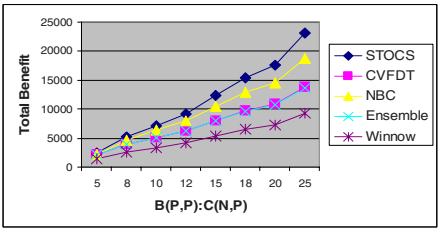


Fig. 2. Performance of STOCS and other online algorithms on the electricity price dataset

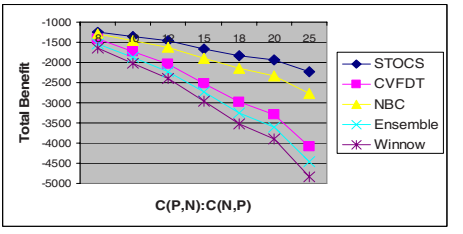


Fig. 3. Performance of STOCS and other online algorithms on KDD Cup 99 dataset

The second real-world dataset used in our experiment is from the well-known data mining competition, KDD Cup 99 [13]. This dataset contains data of network intrusion detection, which shows a typical case of high-speed online data. In our experiment, the class label "snmpgetattack" is regarded as the rare event c_1, while all other class labels are considered as common event c_0. The experiment results are shown in Fig. 3.

As mentioned in section 3, STOCS is also applicable when traditional cost metric (1) is selected as the objective function. We set $B(P,P)=0$ and $B(N,N)=0$ in the experiment on KDD Cup 99 data. Then the evaluation criterion of our algorithm is actually the same as the cost metric (1) discussed in [3]. The total benefits in Fig. 3 are negative because only the costs are calculated in the algorithm.

According to Fig. 3, STOCS again achieves consistently better performance than other algorithms on KDD Cup 99 dataset. The results clearly indicate that STOCS is highly effective in predicting rare events. Moreover, it is proven in the experiment that STOCS is also applicable if applying a traditional cost metric and ignoring classification benefits.

To demonstrate the time effectiveness of STOCS, the cumulative processing time of STOCS on two benchmark datasets are shown in Figs 4-5. As discussed in Section 4.4, STOCS processes data in two stages. In the first stage, STOCS stores incoming data and waits for the window size condition to be satisfied. In the second stage, the window size has been determined and STOCS starts to detect rare events. As seen in Figs 4-5, the cumulative processing time is approximately a linear function of the number of observations in both two stages. This is a clear proof of our claim that,

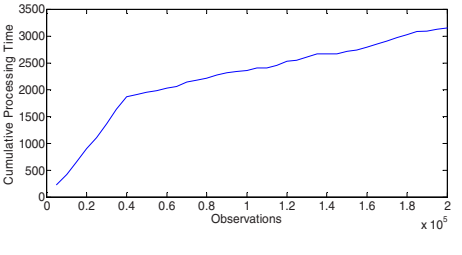


Fig. 4. Cumulative processing time of STOCS on the electricity price dataset

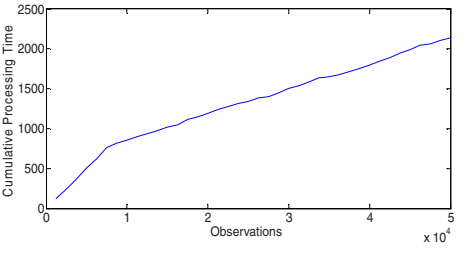


Fig. 5. Cumulative processing time of STOCS on the KDD Cup dataset

STOCS has a linear time complexity and therefore is a highly efficient algorithm for handling high-speed online data.

6 Conclusions

In this paper, a novel approach, namely *STOCS (STatistical Online Cost Sensitive classifier)* is proposed to predict rare events over online data. STOCS consists of several hypothesis tests over Bernoulli distributions. Hence, the probabilities of positive and negative classes in a given Euclidean region can be estimated with normal distribution, according to the Central Limit Theory (CLT). With these probabilities, the observations can be classified by evaluating the expecting benefits.

Two important contributions of this paper are that: (i) By considering the class posterior probability as a density function rather than a point estimate, the proposed algorithm can obtain a better estimate of classification benefit/cost. Our approach therefore demonstrates a significant improvement on predicting rare events over online data. (ii) A size-adjustable moving window and forgetting-factor method are introduced to incrementally revise the classifier with new data.

References

1. R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, S. Zhou: Specification-based anomaly detection: a new approach for detecting network intrusions. In Proc. of the 9th ACM conference on Computer and communications security, (2002).
2. J.H. Zhao, Z.Y. Dong, X. Li and K.P. Wong: A general method for electricity market price spike analysis. IEEE PES General Meeting, (2005).
3. P. Domingos: Metacost: A General Method for Making Classifiers Cost-sensitive, In Proc of SIGKDD (1999).
4. G. Hulten, L. Spencer, and P. Domingos: Mining time-changing data streams. In Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD'01).
5. R.O. Duda and P.E. Hart: Pattern classification and scene analysis, Wiley, New York, 1973.
6. H. Wang, W. Fan, P. S. Yu, J. Han: Mining concept-drifting data streams using ensemble classifiers. Proceedings of SIGKDD (2003).
7. N. Littlestone: Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm, Machine Learning, 2(4):285-318, 1988.
8. G.M. Weiss: Mining with rarity: a unifying framework. SIGKDD Explorations (2004).
9. Fawcett, T: ROC graphs: Notes and practical considerations for data mining representation. Technical Report HPL-2003-4, Hewlett-Packard Labs, Palo Alto, CA (2003).
10. N. V. Chawla, A. Lazarevic, L. O. Hall, and K. Bowyer: SMOTEBoost: Improving prediction of the minority class in boosting. In Proceedings of SIGKDD (2003).
11. C. Elkan: The foundations of cost-sensitive learning. In Proc of IJCAI, (2001).
12. Webpage for downloading Australia electricity market price dataset: http://www.nemmco.com.au/data/csv.htm.
13. Webpage for downloading the dataset of KDD Cup 99: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Structural Learning About Independence Graphs from Multiple Databases

Qiang Zhao, Hua Chen*, and Zhi Geng

Peking University, Beijing 100871, China
chenhua@math.pku.edu.cn

Abstract. In this paper, we propose an approach for structural learning of independence graphs from multiple databases or prior knowledge of conditional independencies. In our approach, we first learn a local graph from each database separately, and then we combine these local graphs together to construct a global graph over all variables. This approach can also be used in structural learning to utilize the prior knowledge of conditional independencies.

Keywords: Graphical models, Structural learning, Independence graphs, Multiple databases.

1 Introduction

Graphical models including independence graphs, directed acyclic graphs (DAG) and Bayesian networks have been applied widely to many fields, such as data mining, pattern recognition, artificial intelligence and causal discovery [4, 8, 9, 10]. Graphical models can be used to cope with uncertainty for a large system with a great number of variables. Structural learning of graphical models from data is an important and difficult problem, and has been discussed by many authors [4, 5, 8, 9, 10, 11]. There are two main kinds of structural learning methods. One is constraint-based learning and the other is score-based learning. Most of structural learning approaches deal with only one database with completely observed data. With the development and popularity of computers, various databases have been built, which may contain different sets of variables and overlap each other. For example, in medical research, a researcher collects data of these variables, another researcher may collect data of other variables, and they have some common variables.

In this paper, we discuss how to learn the structures of independence graphs from multiple databases with different and overlapped variables. In our approach, we first learn a local subgraph from each database separately, and then we combine these subgraphs together to construct a global graph over all variables. Several theoretical results are shown for the validity of our algorithm. Our approach can validly discover independence graphs from multiple databases. The advantage of our local discovery approach is that each independence test is performed conditionally on a small set of variables rather than on all other variables

* Corresponding author.

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1122–1130, 2007.

so that the tests become more powerful and less computational complex. This approach can also utilize the prior knowledge of conditional independencies to reduce the number of variables in each conditional set.

Section 2 gives notation and definitions. In Section 3, we show how to construct the independence graph with multiple databases. We give an example in Section 4 to illustrate our approach for recovering an independence graph. Finally Section 5 discusses the advantages and complexity of the proposed algorithm.

2 Notation and Definitions

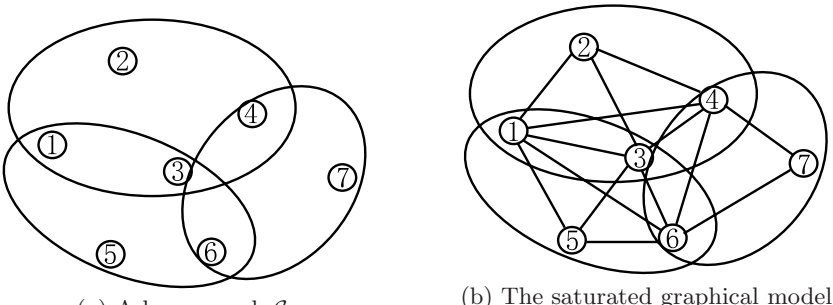
A graph is a pair $G = (V, E)$ where $V = \{x_1, x_2, \ldots, x_n\}$ is a finite set of vertices and E is a subset $V \times V$ of distinct vertices, called the set of edges. An edge is directed pointing from x to y if $< x, y > \in E$. If $< x, y > \in E$ and $< y, x > \in E$, an edge between vertices x and y is undirected, denoted by (x, y) and depicted by a line in the graph. A graph is undirected if it contains only undirected edges. In this paper, we concentrate only on undirected graphs. For an undirected graph G, vertices x and y are adjacent if there is an undirected edges between x and y. Let $ne(x)$ denotes all the vertices that are adjacent with x, called the neighbor set of x. The neighbor set of a vertex set A is defined as $ne(A) = [\cup_{x \in A} ne(x)] \setminus A$. An undirected graph is called complete if each pair of vertices are connected by an edge in the graph.

The vertex set V in a graph G is used to denote an n-dimensional vector of random variables. An independence graph, or more precisely a conditional independence graph, for the variable set V is an undirected graph $G = (V, E)$ in which $(x, y) \notin E$ if and only if x and y are independent conditionally on all other variables, denoted by $x \perp\!\!\!\perp y | V \setminus \{x, y\}$ [12].

A hypergraph is a collection of vertex sets [2, 3]. Multiple databases $\mathcal{C} = \{C_1, \ldots, C_H\}$ are depicted as a hypergraph where a hyperedge C_h is an observed variable set in a database, and $\cup_{h=1}^{H} C_h = V$ [6, 11]. A database with an observed variable set C_h is treated as a sample from a marginal distribution of the variable set C_h. Let $D_h = C_h \cap (\cup_{k \neq h} C_k)$, which is the intersection of C_h and the other sets. Given a collection of databases $\mathcal{C}$, the graphical model with the edge set $E = \cup_h E_h$ is the saturated graphical model where $E_h = C_h \times C_h$ is the edge set of the complete graph over the vertex set C_h since there is no information on higher interactions over different databases. For the saturated graphical model, we have the conditional independencies $(C_h \setminus D_h) \perp\!\!\!\perp (V \setminus C_h) | D_h$. The hypergraph can also be used to depict the prior knowledge of conditional independencies [11].

Example 1. Let $\mathcal{C} = \{C_1 = \{1, 2, 3, 4\}, C_2 = \{1, 3, 5, 6\}, C_3 = \{4, 6, 7\}\}$ be a hypergraph, as shown in Fig. 1 (a). We can get that $D_1 = \{1, 3, 4\}$, $D_2 = \{1, 3, 6\}$ and $D_3 = \{4, 6\}$. The saturated graphical model corresponding to $\mathcal{C}$ is shown in Fig. 1 (b). From the saturated graphical model, we can see that $(C_1 \setminus D_1) \perp\!\!\!\perp (V \setminus C_1) | D_1$, i.e. $\{2\} \perp\!\!\!\perp \{5, 6, 7\} | \{1, 3, 4\}$.

For multiple databases $\mathcal{C} = \{C_1, \ldots, C_H\}$, it should be noted that there is no information on the association among variables that are never observed together, and thus parameters that relate to the association are inestimable without other assumptions. The condition to make our algorithm correct for structural learning from multiple databases $\mathcal{C}$ is that $\mathcal{C}$ must contain sufficient data such that parameters of the underlying independence graph are estimable. For an independence graph, its parameters are estimable if, for each clique g_i, there is a database C_h in $\mathcal{C}$ which contains g_i. Thus multiple databases $\mathcal{C}$ have sufficient data for correct structural learning if there is a database C_h in $\mathcal{C}$ for each clique g_i such that C_h contains g_i in the underlying independence graph. Every database C_h can be seen as a maximum complete undirected graph to depict possible association among variables in C_h. We assume that all independencies inferred from multiple databases are true for the underlying independence graph.



(a) A hypergraph $\mathcal{C}$.

(b) The saturated graphical model corresponding to $\mathcal{C}$.

Fig. 1. A hypergraph

3 Structural Learning of Independence Graphs

In this section, we propose an approach for structural learning of independence graphs. In our approach, we first learn a subgraph from each database, and then we combine these subgraphs together to construct a global graph over all variables. Below we give the theoretical results which ensures the correctness of this approach. By definition of an independence graph, the existence of an edge between x and y can be determined by testing conditional independence $x \perp\!\!\!\perp y | V \setminus \{x, y\}$. Usually all databases are needed to calculate the statistics for testing this independence. In the following theorems, we show that this may not be needed for some edges. First we give a lemma to be used in proofs of theorems.

Lemma 1. *Properties of conditional independence:*

1. $(X \perp\!\!\!\perp Y | Z) \Rightarrow (Y \perp\!\!\!\perp X | Z)$;
2. $(X \perp\!\!\!\perp YW | Z) \Rightarrow (X \perp\!\!\!\perp Y | Z)$;
3. $(X \perp\!\!\!\perp YW | Z) \Rightarrow (X \perp\!\!\!\perp Y | ZW)$;
4. $(X \perp\!\!\!\perp Y | Z) \& (X \perp\!\!\!\perp W | ZY) \Rightarrow (X \perp\!\!\!\perp YW | Z)$;
5. $(X \perp\!\!\!\perp W | ZY) \& (X \perp\!\!\!\perp Y | ZW) \Rightarrow (X \perp\!\!\!\perp YW | Z)$.

Proof. See page 11 of [9] for the proof.

Theorem 1. *Let A, B and C be a partition of all variables in V, and variables x and y be contained in A. Suppose that $A \perp\!\!\!\perp B|C$. Then $x \perp\!\!\!\perp y|V \setminus \{x, y\}$ if and only if $x \perp\!\!\!\perp y|(A \cup C) \setminus \{x, y\}$.*

Proof. We first prove the sufficiency. Because $A \perp\!\!\!\perp B|C$, we can get from the third property of Lemma 1

$$P(x|A \setminus \{x\}, B, C) = P(x|A \setminus \{x\}, C).$$

From the sufficient condition, the right hand side of the above formula can be rewritten as

$$\begin{aligned} P(x|A \setminus \{x, y\}, C) &= \frac{P(x, A \setminus \{x, y\}|C)}{P(A \setminus \{x, y\}|C)} \\ &= \frac{P(x, A \setminus \{x, y\}|C)P(B|C)}{P(A \setminus \{x, y\}|C)P(B|C)} \\ &= \frac{P(x, A \setminus \{x, y\}, B|C)}{P(A \setminus \{x, y\}, B|C)} \\ &= P(x|A \setminus \{x, y\}, B, C). \end{aligned}$$

Thus we proved the sufficiency.

Next we prove the necessity. Because $A \perp\!\!\!\perp B|C$, we can get from the third property of Lemma 1

$$P(x|A \setminus \{x\}, C) = P(x|A \setminus \{x\}, B, C).$$

Because $x \perp\!\!\!\perp y|(A \setminus \{x, y\}, B, C)$, the right of the above formula is equal to

$$\begin{aligned} P(x|A \setminus \{x, y\}, B, C) &= \frac{P(x, A \setminus \{x, y\}, B|C)}{P(A \setminus \{x, y\}, B|C)} \\ &= \frac{P(x, A \setminus \{x, y\}|C)P(B|C)}{P(A \setminus \{x, y\}|C)P(B|C)} \\ &= \frac{P(x, A \setminus \{x, y\}|C)}{P(A \setminus \{x, y\}|C)} \\ &= P(x|A \setminus \{x, y\}, C). \end{aligned}$$

Thus we proved Theorem 1.

Let $A = C_h \setminus D_h$, which is a vertex set that only appears in the database C_h. According to Theorem 1, we can see that the existence of an edge whose two vertices fall into only one database can be determined validly by using the database C_h only.

Theorem 2. *Let A, B and C be a partition of all variables in V, and variables x and y be contained in A and C respectively. Suppose $A \perp\!\!\!\perp B|C$. Then $x \perp\!\!\!\perp y|V \setminus \{x, y\}$ if and only if $x \perp\!\!\!\perp y|(A \cup C) \setminus \{x, y\}$.*

Proof. Because $A \perp\!\!\!\perp B|C$, we can get from the third property of Lemma 1 that $x \perp\!\!\!\perp B|(A \setminus \{x\}, C)$. We first prove the sufficiency. Because $x \perp\!\!\!\perp y|(A \setminus \{x\}, C \setminus \{y\})$ and $x \perp\!\!\!\perp B|(y, A \setminus \{x\}, C \setminus \{y\})$, we can get from the fourth property of Lemma 1 that $x \perp\!\!\!\perp (y, B)|(A \setminus \{x\}, C \setminus \{y\})$. From the third property of Lemma 1, we can get $x \perp\!\!\!\perp y|(A \setminus \{x\}, B, C \setminus \{y\})$. Thus we proved the sufficiency.

Next we prove the necessity. Because $x \perp\!\!\!\perp y|(A \setminus \{x\}, B, C \setminus \{y\})$ and $x \perp\!\!\!\perp B|(y, A \setminus \{x\}, C \setminus \{y\})$, we can get from the fifth property of Lemma 1 that $x \perp\!\!\!\perp (y, B)|(A \setminus \{x\}, C \setminus \{y\})$. From the second property of Lemma 1, we can get $x \perp\!\!\!\perp y|(A \setminus \{x\}, C \setminus \{y\})$. Thus we proved Theorem 2.

According to Theorem 2, the existence of an edge whose one of two vertices is contained only by one database can also be determined validly by using the database only.

From the above two Theorems, we know that an edge whose at least one of two vertices is contained only by one database C_h can be determined by using the marginal distribution of C_h without requirement of the other databases.

Now we consider how to determine an edge (x, y) both of whose vertices are contained by at least two databases. Let $C_{xy} = \cup_{\{h: x \in C_h or\ y \in C_h\}} C_h$. From Theorems 1 and 2, it can be shown that the existence of edge (x, y) can be determined by testing whether x and y are independent conditionally on $C_{xy} \setminus \{x, y\}$. But the union set C_{xy} may contain a large number of variables. Below we discuss how to reduce variables from C_{xy}.

For a database C_h, suppose that both x and y are contained in D_h, where $D_h = C_h \cap (\cup_{k \neq h} C_k)$. Let NE_h denote the neighbor set of D_h in G_h which is the independence graph obtained by using database C_h. From Theorems 1 and 2 we can get that NE_h is contained in $ne(D_h)$ of G which is the independence graph for the whole variable set V. Let $C'_{xy} = \cup_{\{h: x \in C_h or\ y \in C_h\}} (NE_h \cup D_h)$. From the property of multiple databases, we have that $\cup_{\{h: x \in C_h or\ y \in C_h\}} D_h$ is independent of $V \setminus C'_{xy}$ conditionally on $\cup_{\{h: x \in C_h or\ y \in C_h\}} NE_h$ and that neither x nor y is contained in $\cup_{\{h: x \in C_h or\ y \in C_h\}} NE_h$. Thus from Theorem 1, we can determine the existence of edge (x, y) by the marginal distribution of C'_{xy}, which is a subset of C_{xy}.

Now we give the algorithm for structural learning of independence graphical models.

Algorithm: Construct an independence graph from multiple databases

1. Input: Multiple databases $\mathcal{C} = \{C_1, \ldots, C_H\}$.
2. Construct a local independence graph G_h from database C_h separately for each h:
   - Initialize G_h as a complete undirected graph;
   - For x and y are not both contained in D_h, then delete edge (x, y) from G_h if $x \perp\!\!\!\perp y|C_h \setminus \{x, y\}$.
3. Construct the global independence graph G_V:
   - Initialize the edge set E of G_V as the union of all edge sets of G_h for $h = 1, \ldots, H$;

– For a pair of variables x and y contained in some D_h, determine the existence of edge (x, y) by testing $x \perp\!\!\!\perp y | C'_{xy} \setminus \{x, y\}$.

4. Output: the independence graph G_V.

According to Theorems 1 and 2, the independence graph constructed by the above algorithm is validly, and the statistical inference is more efficient than the traditional approach in which each edge (x, y) is determined by testing $x \perp\!\!\!\perp y | V \setminus \{x, y\}$ since the conditional set $V \setminus \{x, y\}$ for independence tests is much larger than $C_h \setminus \{x, y\}$ and $C'_{xy} \setminus \{x, y\}$.

Step 3 in the algorithm becomes much simpler if we assume that the conditional independence $x \perp\!\!\!\perp y | A$ implies $x \perp\!\!\!\perp y | B$ for all $A \subseteq B$. The assumption is similar to the faithfulness assumption for DAGs [10]. Under this assumption, we ensure that edges are deleted validly at Step 2, and thus an edge between x and y should be absent in G_V if it is absent in any subgraph G_h.

4 Illustration of Structural Learning

In this section, we illustrate our algorithm using the ALARM network in Fig. 2 that is often used to evaluate structural learning algorithms [1, 7, 10]. The ALARM network in Fig. 2 describes associations among 37 variables in a medical diagnostic system for patient monitoring. Using the network, some researchers generate continuous data from normal distributions and others generate discrete data from multinomial distributions [7, 10]. Our approach is applicable for both continuous and discrete data. Since the validity of our algorithm can be ensured by Theorems 1 and 2, the algorithm is illustrated by using conditional independencies from the underlying independence graph in Fig. 2 rather than conditional independence tests from simulated data.

Suppose that we have three databases as depicted by the hypergraph in Fig. 2. Database C_1 contains variables $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 27, 28,$

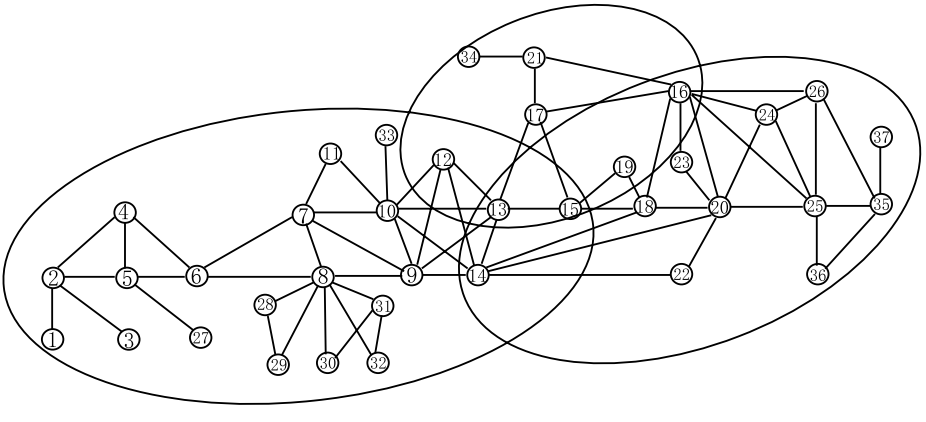
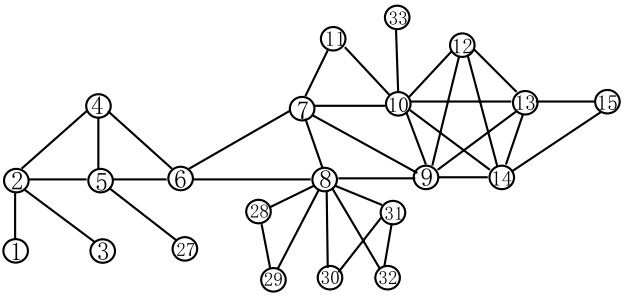
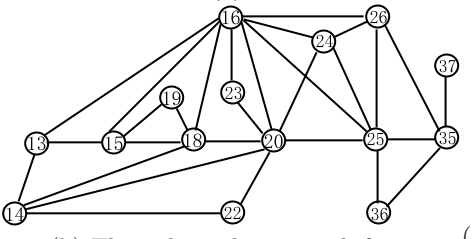


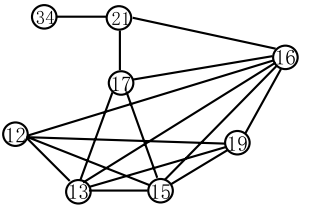
Fig. 2. The ALARM network and multiple databases



(a) The independence graph from database C_1



(b) The independence graph from database C_2



(c) The independence graph from database C_3

Fig. 3. Local independence graphs for all databases

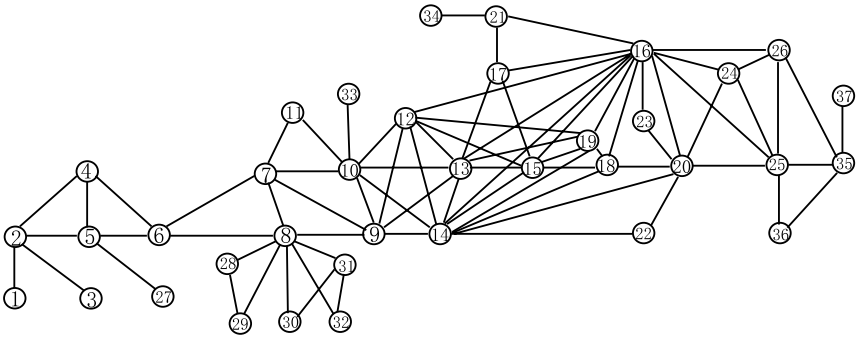


Fig. 4. The initial graph obtained by combining all local graphs

$29, 30, 31, 32, 33\}$, database C_2 contains variables $\{12, 13, 15, 16, 17, 19, 21, 34\}$, and database C_3 contains variables $\{13, 14, 15, 16, 18, 19, 20, 22, 23, 24, 25, 26, 35, 36, 37\}$. Thus $D_1 = \{12, 13, 14, 15\}$, $D_2 = \{12, 13, 15, 16, 19\}$ and $D_3 = \{13, 14, 15, 16, 19\}$. At Step 2, the local independence graphs are obtained separately from the three databases, as shown in Fig. 3 (a), (b) and (c) respectively. From Fig. 3, we can get that $NE_1 = \{9, 10\}$, $NE_2 = \{17, 21\}$ and $NE_3 = \{18, 20, 22, 23, 24, 25, 26\}$.

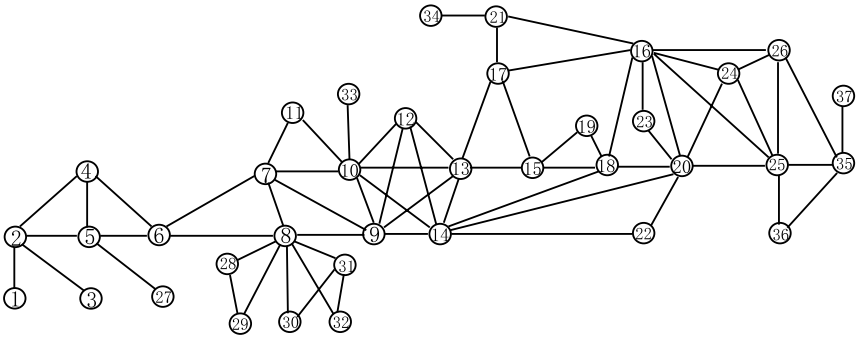


Fig. 5. The independence graph constructed from multiple databases

At Step 3, we first initialize the edge set E of the global graph G_V as the union of all edge sets of G_h for $h = 1, 2, 3$, as shown in Fig. 4. For any pairs of variables contained in every D_h, we must redetermine the existence of corresponding edges. For example, for variables 13 and 19 contained in $D_2 = \{12, 13, 15, 16, 19\}$, $C'_{13,19} = \cup_{\{h:13 \in C_h or 19 \in C_h\}}(NE_h \cup D_h) = \{9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26\}$. we should delete edge $(13, 19)$ because $13 \perp\!\!\!\perp 19 | C'_{13,19} \setminus \{13, 19\}$. Finally we get the global independence graph G_V showed in Fig. 5, which is the same as the underlying graph in Fig. 2.

5 Advantages and Complexity

There are several obvious advantages of our approach for structural learning. Firstly, independence tests are performed only conditionally on smaller sets contained in a database C_h or C'_{xy} rather than on the full set of all other variables. Thus our algorithm has higher power for statistical tests.

Secondly, the theoretical results proposed in this paper can be applied to scheme design of multiple databases. Without loss of information on structural learning of independence graphs, a joint data set can be replaced by a group of incomplete data set based on the prior knowledge of conditional independencies among variables.

Thirdly, for complexity, our approach tests as many times of conditional independence as the ordinary approaches. However, for the ordinary approaches, each test is performed conditionally on $n-2$ variables. For a large n, an independence test conditionally on such a large set of discrete variables is impractical. In our approach, these tests are taken conditionally only on smaller sets of variables such that they become more practical. On the other hand, the EM algorithm over all n variables is required for each test in the ordinary approaches, which makes computation more complex. In our approach, only those conditional independence tests for edges falling in separators requires the EM algorithm over a smaller variable set. Thus our approach is less computational complex than the ordinary approaches.

Finally, we discuss the validity of the decomposition approach proposed in this paper. For a given collection of databases $\mathcal{C}$, the decomposition approach can obtain the same independence graph as that obtained by using the ordinary approaches if there were no errors of independence tests. In the decomposition approach, observed data are collapsed into marginal data, and thus independence tests are more efficient. If each clique g_i of the underlying independence graph is contained by some database C_h in $\mathcal{C}$, then the joint distribution can be identified from these marginal distributions of observed variables, and thus the decomposition approach is valid for recovering the correct structure of the underlying independence graph.

Acknowledgements

This research was supported by NSFC, MSRA, NBRP 2003CB715900 and PSFC 20060400365.

References

[1] Beinlich, I., Suermondt, H., Chavez, R., Cooper, G.: The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks, in: Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine, Springer-Verlag, Berlin, 1989, pp. 247-256.
[2] Beeri, C., Fagin, R., Maier, D., Yannakakis, M.: On the desirability of acyclic database schemes, J. Association for Computing Machinery 30 (1983) 479-513.
[3] Berge, C.: Graphs and Hypergraphs, 2nd ed., North-Holland, Amsterdam, 1976.
[4] Cowell, R. G., David, A. P., Lauritzen, S. L., Spiegelhalter, D. J.: Probabilistic Networks and Expert Systems, Springer Publications, New York, 1999.
[5] Deng, K., Liu, D., Gao, S., Geng, Z.: Structural learning of graphical models and its applications to traditional Chinese medicine, Fuzzy Systems and Knowledge Discovery, Lipo Wang and Yaochu Jin Ed. Lecture Notes in Computer Science Vol. 3614, 362-367, Springer-Verlag, 2005.
[6] Geng, Z., Wan, K., Tao, F.: Mixed graphical models with missing data and the partial imputation EM algorithm, Scan. J. of Stat. 27 (2000) 433-444.
[7] Heckerman, D.: A tutorial on learning with Bayesian networks, Learning in Graphical Models, M. I. Jordan,(Ed.), 301-354, Kluwer Academic Pub., Netherlands, 1998.
[8] Lauritzen, S. L.: Graphical Models, Clarendon Press, Oxford, 1996.
[9] Pearl, J.: Causality, Cambridge University Press, Cambridge, 2000.
[10] Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction and Search, 2nd ed. MIT Press, Cambridge, 2000.
[11] Xie, X., Geng, Z., Zhao, Q.: Decomposition of structural learning about directed acyclic graphs. Artificial Intelligence 170 (2006) 422-439.
[12] Whittaker, J.: Graphical models in applied multivariate statistics. Chichester, U.K.:Wiley, 1990.

An Effective Method For Calculating Natural Adjacency Relation in Spatial Database

Renliang Zhao[1] and Jiatian Li[1, 2]

[1] National Geometrics Center of China, Beijing, China, 100044
[2] School of Resources and Safety Engineering, CMUT, Beijing, China, 100083

Abstract. This paper explores the way in which natural adjacency relation, spatial database are closely integrated through the spatial index for spatial data querying and mining. A Delaunay triangulation approach for constructing spatial index is proposed, which overcomes the conflict between line-intersection computation and natural adjacency isn't satisfying constrained condition of Euclidean distance. Based on this approach, a spatial index prototype for discrete areal objects-Quad GridFile is designed and implemented by using Java extending Oracle Spatial. It demonstrates foundational information extraction ability for geospatial database.

1 Introduction

Sibson (1980) firstly introduced the concept of natural adjacency in the interpolation method. He defined two spatial objects, which had shared Voronoi boundary as the natural neighborhoods, that is to say, there is a natural adjacency relationship between the two spatial objects. The ability for efficiently computing natural adjacency relationship has become more and more important in an increasing number of applications including spatial data mining [6], spatial query language [1], and mapping generalization [7]. Over the past decades, many scholars mainly have been study for adjacency computation with Voronoi diagram method. Gold (1992) represented the natural adjacency relationship extracted from the Voronoi diagram as a VAG (Voronoi adjacency graph) structure. In this graph, the spatial object is expressed as a node. If there is the natural adjacency relationship between the two spatial objects, the relationship is expressed as an edge. In this way, the extraction of the natural adjacency relationship is translated into the graph structure. Okabe *et al.* (1992) adopted the Winged-Edge structure which can definitely represented the relationships among Voronoi nodes, Voronoi edges and Voronoi areas. Extracting the natural adjacency relationship based on Winged-Edge has simple querying process and doesn't need extra computation. However, the Winged-Edge structure is a topology structure, which need maintain the node table, edge table and area table. So, due to the bad ability to locally updating of Voronoi diagram, the whole Voronoi diagram of all spatial objects and the Winged-Edge structure has to be reconstructed even if there is a small change of spatial object [3].

The main property of natural adjacency relation is that it isn't satisfying Euclidean distance restriction. The conventional line intersection method isn't competent for

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1131–1139, 2007.

computing the natural adjacency relationship among discrete objects in spatial database [1]. Therefore, the property results in candidate set aren't complete with current spatial index methods. This paper is organized as follows. In Section 2, we put forward a new structure-UnitsDelaunay and its constructing algorithm for describing the natural adjacency relationship based on Delaunay triangulation. Section 3 presents a Quad GridFile spatial index integrated the UnitsDelaunay for the filter process in spatial database. The experimental design and analysis are presented in Section 4. Finally, Section 5 concludes the paper and points out future work.

2 Natural Adjacency Describing with Delaunay Triangulation

In Euclidean space $\mathcal{R}^2$, firstly seek the complement ($\mathcal{R}^2\backslash\mathbf{Q}$) of the areal objects set **Q** based on the Delaunay triangulation, then classify, merge and represent these complement over again by conditions of adjacent to the triangle, consequently form the UnitsDelauany which is the combo of spatial objects and the natural adjacency relationship regions.

2.1 Classification of Delaunay Triangles and UnitsDelaunay Structure

For **Q**, the Delaunay structure is a constraint structure, which is generated based on boundary nodes set **P**. A triangle is made up of three edges and it is a simplex. If the edge is queried by points, the function $\boldsymbol{f}$ in definition 1 can be established in order to classify the edges.

Definition 1. $\mathbf{p}_m$, $\mathbf{p}_n$ are individually the boundaries of Q_m, Q_n(Q_m, $Q_n \in \mathbf{Q}$), $\forall t \in \mathcal{D}(\mathbf{P})$, p_1, p_2 and p_3 are the three vertices of t, if the condition of p_i, p_j ($i \neq j$, i, j=1, 2, 3)$\wedge$ $p_i \in \mathbf{p}_m \wedge p_j \in \mathbf{p}_n$ is true, then $\boldsymbol{f}(p_i, p_j)$=0; if the condition of p_i, p_j($i \neq j$, i, j=1, 2, 3)$\wedge$(p_i, $p_j \in \mathbf{p}_m \vee p_i$, $p_j \in \mathbf{p}_n$) is tenable, then $\boldsymbol{f}(p_i, p_j)$=1; if the condition of p_i, p_j(i=j, i, j=1, 2, 3) is tenable, then $\boldsymbol{f}(p_i, p_j)$=2.

For $\forall t \in \mathcal{D}(\mathbf{P})$, we can classify the triangle as three kinds if we reference the types of the three edges of the triangle. By absolute value of the formula (1)

$$\boldsymbol{C} = \begin{vmatrix} f(p_1,p_1) & f(p_1,p_2) & f(p_1,p_3) \\ f(p_2,p_1) & f(p_2,p_2) & f(p_2,p_3) \\ f(p_3,p_1) & f(p_3,p_2) & f(p_3,p_3) \end{vmatrix}, \tag{1}$$

we can get 5 kinds of expression for t,

$$C_1 = \begin{vmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{vmatrix} \quad C_2 = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{vmatrix} \quad C_3 = \begin{vmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 2 \end{vmatrix} \quad C_4 = \begin{vmatrix} 2 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 2 \end{vmatrix} \quad C_5 = \begin{vmatrix} 2 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{vmatrix}$$

Definition 2. For $\forall t \in \mathcal{D}(\mathbf{P})$, if $|\boldsymbol{C}|$=4, then the three edges come from the same spatial object (type α); if $|\boldsymbol{C}|$=6, the two of the three edges come from two different spatial objects, the other one of the three edges comes from a spatial object (type β); if $|\boldsymbol{C}|$=8, the three edges come from three different spatial objects (type γ).

Theorem 1. Triangle of type α doesn't completely exist in areal object interior.

Proof. According to the natural **D2** of Delaunay introduced in [10], the triangulation composed by the boundary points **p** of the convex polygon Q is $\mathcal{A}(\mathbf{p})=\{t_1, \ldots, t_n, n\in\mathcal{N}\}$, and the external Delaunay boundary of $\mathcal{A}(\mathbf{p})$ is the convex hull of $\mathcal{A}(\mathbf{p})$, i.e. $\bigcup_{i=1}^{n} t_i = \mathrm{CH}(\mathbf{p})$, $\because Q$ is convex, $\therefore \mathrm{CH}(\mathbf{p})=\partial Q$. If Q is concave polygon, then CH(**p**) must contain the ∂Q, $\because \forall t_i\in\mathcal{A}(\mathbf{p})$, $i\in\mathcal{N}$, the vertices of t_i all come from Q, $\therefore$ type of t_i is α, $\therefore$ theorem 1 is tenable. □

According to the theorem 1, the type α of triangles should be progressively classified. For every triangle t of type α in the Delaunay triangulation, point b is its barycenter, the three vertices of the triangle come from polygon Q. The *Contain*(Q, b) is boolean operator, which is used to judge whether Q contains b, then we can define:

Definition 3. For $\forall t\in\mathcal{D}(\mathbf{P})$, if *contain*($Q$, b) is true, then t's type is δ; if *contain*(Q, b) is false, then t's type is α.

Theorem 2. The space scope of **Q** is equal to the set $\mathbf{T}_\alpha$ of all triangles of type α.

Proof. If Q is convex polygon, then **p** has the unique partial result, i.e. $\mathcal{A}(\mathbf{p})=\mathbf{T}_\alpha$, if Q is concave polygon, then it can generate limited convex partition and every convex partition is a convex polygon. So, theorem 2 is tenable. □

Corollary 1. In the $\mathcal{R}^2$, the complement of **Q** can be represented by the set of triangles of type β, γ and δ.

Proof. According to theorem 2, $\mathbf{T}_\alpha=\mathbf{Q}$, $\because \mathcal{R}^2=\mathbf{T}_\alpha\cup\mathbf{T}_\beta\cup\mathbf{T}_\gamma\cup\mathbf{T}_\delta$, the complement of **Q** can be expressed as $\mathcal{R}^2\backslash\mathbf{Q}=\mathcal{D}(\mathbf{P})\backslash\mathbf{T}_\alpha=\{\mathbf{T}_\beta\cup\mathbf{T}_\gamma\cup\mathbf{T}_\delta\}$, so, the corollary 1 is tenable. □

Corollary 2. The natural adjacency relationship can only exist in $\{\mathbf{T}_\beta\cup\mathbf{T}_\gamma\cup\mathbf{T}_\delta\}$.

Proof. Assuming q is a querying point in $\mathcal{R}^2$, if $q\cap\mathbf{T}_\alpha=\neg\varnothing$, according to theorem 2, there is $q\cap\mathbf{Q}=\neg\varnothing$ i.e. q intersects with some spatial objects. The natural adjacency relationship only exists among the discrete spatial objects. Corollary 2 is tenable. □

After a new point was inserted, the Delaunay triangulation as the Voronoi dual also had corresponding changes [2], [9]. The three vertices of each triangle are the natural adjacency of the inserted point, i.e., every vertex all has natural adjacent spatial relations with the inserted point.

Definition 4. For $\forall t\in\mathcal{D}(\mathbf{P})$, if $t\in\mathbf{T}_\beta$, assuming the vertices of t come from M, N(M, $N\in\mathbf{Q}$), then M and N have adjacency relationship, denoted as *adj*(M, N). For $\forall t\in\mathcal{D}(\mathbf{P})$, if $t\in\mathbf{T}_\gamma$, assuming the vertices of t come from M, N and V(M, N, $V\in\mathbf{Q}$), then there are *adj*(M, N), *adj*(M, V), *adj*(N, V).

Theorem 4. If M and N have adjacency relationship, then the querying point q which drops into the region $\{\bigcup_{k=1}^{n}(t_k \setminus p_{ij}),\ t_k\in\mathbf{T}_{MN}\wedge f(p_i, p_j)=1\}$ only has adjacency relationships with the spatial object M and N.

Proof. The querying point drops into $\mathcal{D}(\mathbf{P})\backslash\{\bigcup_{k=1}^{n}(t_k \setminus p_{ij}),\ t_k \in \mathbf{T}_{MN} \wedge f(p_i, p_j)=1\}$, if $q \cap \mathbf{T}_\alpha = \neg\varnothing$, according to the theorem 2, then there isn't spatial objects adjacent to the querying point. If $q \cap \mathbf{T}_\alpha = \varnothing$, the spatial objects adjacent to the point isn't only the M and N.

The querying point drops into $\{\bigcup_{k=1}^{n} t_k,\ t_k \in \mathbf{T}_{MN}\}$, if $q \cap p_i p_j = \neg\varnothing$ $(p_i p_j \in t_k)$, i.e. $q \cap \mathbf{T}_\alpha = \neg\varnothing$, there still isn't spatial objects adjacent to the querying point. So the theorem 4 is tenable. □

The region $\{\bigcup_{k=1}^{n}(t_k \setminus p_{ij}),\ t_k \in \mathbf{T}_{MN} \wedge f(p_i, p_j)=1\}$ generated by M and N (M, $N \in \mathbf{Q}$) is called unit. Theorem 4 is the same as the units composed by the triangles of type $\mathbf{T}_\gamma$.

Definition 5. In $\mathcal{D}(\mathbf{P})$, the set of all units is called as the UnitsDelaunay structure, i.e. $\bigcup_{i=1}^{n} \text{unit}_i$, denoted as $\mathcal{U}(\mathbf{P})$.

In UnitsDelaunay, every unit is made up of the set of $\mathbf{T}_{MN}$ (M, $N \in \mathbf{Q}$) or $\mathbf{T}_{KMN}$ (K, M, $N \in \mathbf{Q}$), which is denoted as unit^2 or unit^3. The unit represents the unique of the adjacent spatial object. For the type $\mathbf{T}_{MN}$, we can confirm:

$$\text{unit}^2 \Rightarrow \{adj(M, N)\}, \tag{2}$$

$$\boldsymbol{PQ}(q, \text{unit}^2) \wedge q \cap \text{unit}^2 = \neg\varnothing \Rightarrow \{adj(q, M), adj(q, N)\}. \tag{3}$$

For the type $\mathbf{T}_{KMN}$, we can confirm:

$$\text{unit}^3 \Rightarrow \{adj(K, M), adj(K, N), adj(M, N)\}, \tag{4}$$

$$\boldsymbol{PQ}(q, \text{unit}^3) \wedge q \cap \text{unit}^3 = \neg\varnothing \Rightarrow \{adj(q, K), adj(q, M), adj(q, N)\}. \tag{5}$$

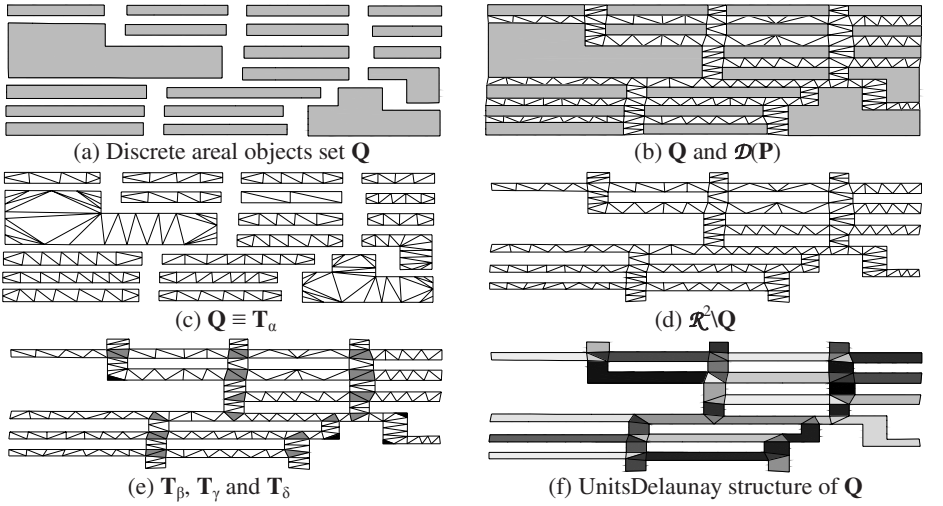


Fig. 1. Computing based on classifying triangles

2.2 Construction Algorithm for UnitsDelaunay Structure

When the units of the triangles of type β, γ were generated, the triangles of type δ which have shared boundaries must be considered in order to ensure completeness of the result. The construction algorithm of UnitsDelaunay structure: input the arrays $\mathbf{T}_\beta$, $\mathbf{T}_\gamma$ and $\mathbf{T}_\delta$, output UnitsDelaunay =$\{\text{unit}_1, \ldots, \text{unit}_n\}$.

Step 1: Seek the boundary nodes of arbitrary unit.
(1) for $t \in \mathbf{T}_\beta$, respectively add the vertices of t into $\boldsymbol{arr}_1$ and $\boldsymbol{arr}_2$ according to the different *fid*. For $t \in \mathbf{T}_\gamma$, deal in step 3; (2) in $\mathbf{T}_\delta$, if $t'(t' \in \mathbf{T}_\delta)$ and t have the common boundary relationship, add the vertices of t' into $\boldsymbol{arr}_1$ or $\boldsymbol{arr}_2$, $t \leftarrow t'$, repeat (2); (3) in $\mathbf{T}_\beta$, if t'' $(t'' \in \mathbf{T}_\beta)$ and t have the same source, respectively add the vertices of t'' into $\boldsymbol{arr}_1$ and $\boldsymbol{arr}_2$ according to the different *fid*, and then perform step 2 with parameter t''.

Step 2: Generate the units based on boundary nodes.
(1) sort the $\boldsymbol{arr}_1$ with the descend *pid*, if the *pid* every node in the $\boldsymbol{arr}_1$ is all not equal to 0, then the process is over and goes to (4); (2) if there is a node in the $\boldsymbol{arr}_1$ which it's *pid* is equal to 0, then judge whether the $\boldsymbol{arr}_1$ is continuous in $\mathbb{Z}$ or not, if the $\boldsymbol{arr}_1$ is continuous, then process is over; (3) if $\boldsymbol{arr}_1$ isn't continuous, seek the broken point and evaluate the *pid* of the point with *pid*'. Find the points which *pid* is less than *pid*' in $\boldsymbol{arr}_1$, sort by descendible *pid* and append the sorted result into $\boldsymbol{arr}_1$; (4) deal with $\boldsymbol{arr}_2$ according to (1)~(3); (5) append $\boldsymbol{arr}_2$ into $\boldsymbol{arr}_1$; (6) construct the unit with $\boldsymbol{arr}_1$, ended.

Step 3: Build the unit of triangle t of type γ.
(1) store the 3 vertices of t into array $\boldsymbol{arr}$; (2) in $\mathbf{T}_\delta$, if $p_ip_j(t' \in \mathbf{T}_\delta, p_ip_j \in t')$ is equal to edge h_mh_n of t, insert $p_k(k \neq i, j)$ into $\boldsymbol{arr}$ at the position between h_m and h_n, repeat (2); (3) construct $\boldsymbol{arr}$, ended.

Assuming that the point set $\mathbf{P}$ is composed by v spatial objects, array $\mathbf{T}_\beta$, $\mathbf{T}_\gamma$, $\mathbf{T}_\delta$ respectively contain n, m, k elements. The time complexity of step 3 is $\boldsymbol{O}(mk)$, step 2 is $\boldsymbol{O}((n+m+k)/v*\log((n+m+k)/v))$, and step 1 is $\boldsymbol{O}(n^2+mn)$. Generally, $m \ll n$ and $k \ll n$, the time complexity of the algorithm is $\boldsymbol{O}(n^3)$.

3 Quad GridFile Spatial Index

In spatial database, the spatial selection method is of performance for two computing levels: filter and refinement. The filter level, which is commonly supported by spatial index, is the precondition and base of quick computation. By combining the UnitsDelaunay structure with the spatial index, the relations between spatial selection and spatial object can be built. Thus, the line intersection computation can be applied to compute the natural adjacency relationship.

3.1 Spatial Approximation for Unit^2 and Unit^3

The key idea of the approximation of spatial objects is to regard the bounding geometry object of a spatial object as an operated object. Approximation policy is the approximation method in spatial index for unit^2 and unit^3.

Theorem 5. q is a querying point, if $q \cap C(t_\beta) = \neg\varnothing \wedge q \cap C(t_\gamma) = \neg\varnothing \wedge (t_\beta$ and t_γ have shared boundary), then $\boldsymbol{PQ}(q, t_\beta) \cup \boldsymbol{PQ}(q, t_\gamma) \equiv \boldsymbol{PQ}(q, t_\gamma)$.

Proof. $\exists t_\gamma \in \text{unit}^3$, $\exists t_\beta \in \text{unit}^2$, by formula (4) and (6), they satisfy $\boldsymbol{PQ}(q, t_\beta) \subset \boldsymbol{PQ}(q, t_\gamma)$. So, the theorem 5 is tenable. □

Definition 6. The repository of all binary relationships for querying point q is called as the closure of q, recorded as q^+.

Theorem 6. q^+ is complete.

Proof. For the querying point q, the completeness of the q^+ means that q^+ includes all the binary relationships with q in $\mathcal{U}(\mathbf{P})$.Assuming there is a binary relationship $R<M, N>$, which has natural adjacency relationship with q, but $R \notin q^+$. According to the definition 4 and theorem 4, the R does not exist. So, the theorem 6 is tenable. □

The unit^3 is composed by a triangle of type γ. According to theorem 5 and theorem 6, in order to confirm the complete candidate set of the natural adjacency, the MBR of minimum boundary circle (MBC) is adopted as the approximate geometry object. For the unit^2, the MBR is adopted.

3.2 Basic Structure of Quad GridFile

Quad partition is a regular level partition structure. It is a divide-conquer method based on two aspects: Firstly, the Quad GridFile can be built based on the Quad GridFile of the cell. This can effectively use the memory and balance the CPU task; secondly, updating index structure can be completed by reconstructing GridFile structure of correlative cell region.

The GridFile spatial index structure divides the embedded 2-Dimension space by multi-attribute index. Use its objective is to realize the principles of secondary diskettes visit: first visit gets the directory items; another visit actually buckets to obtain the actual records. Each grid directory points a catalogue data bucket, which stored in the actual data tuple structure.

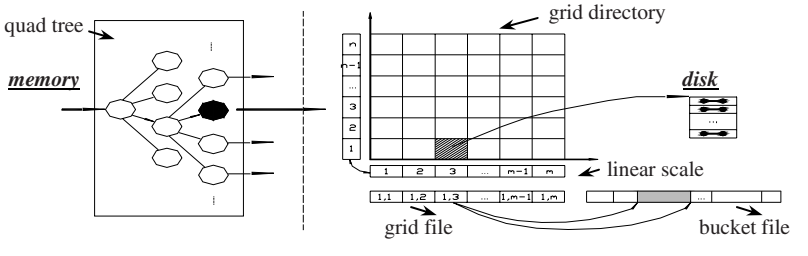


Fig. 2. Basic composition of Quad GridFile

4 Experiment

Experiment of proposed structure and method was carried out under Oracle 9.2.1 and Eclipse 3.1 with JDK 1.5. The progress of creating GridFile includes three parts:

(1) generating Delaunay triangular irregular network; (2) generating unitsDelaunay structure; (3) creating GridFile spatial files. The relation between number of areal object boundary nodes and time for creating GridFile as shown in Fig.3.

There is the linear increasing relation between time for creating GridFile (t) and number of areal object boundary nodes (n), and it is represented as

$$t = k \cdot n + b, \tag{6}$$

while the inserting, deleting and modifying operators are happen, updating operation for Quad GridFile is required. We replace whole Quad GridFile index updating with one reconstructing progress of GridFile, which is pointed by Quad one leaf node. Assuming the maximum required time is $t_{\max}$ (ms) for reconstructing progress, the number of boundary nodes is N, the number of Quad level is l, there is

$$t_{\max} \geq k \cdot n + b, \tag{7}$$

where k, b are constant and n is

$$n = \mathrm{N} / 4^{l}, \tag{8}$$

the relation of l and N is

$$l \geq \left\lceil \log_4 \frac{k \cdot \mathrm{N}}{t_{\max} - b} \right\rceil. \tag{9}$$

The fitted equation is $t = 0.3773n + 1576.8$, which is got by Fig.3. If the $t_{\max}$ is 4000(ms) and N is 50000, upon fitted equation substitution to (9), we can get the l = 2, i.e. every GridFile includes 3125 boundary nodes, thus, the reconstructing about time of GridFile is 2756(ms). In order to test capability of Quad GridFile spatial index, we analyzed test data, which got by moved point query in fixed grid form. For seven random query points, test data are shown in Table 1.

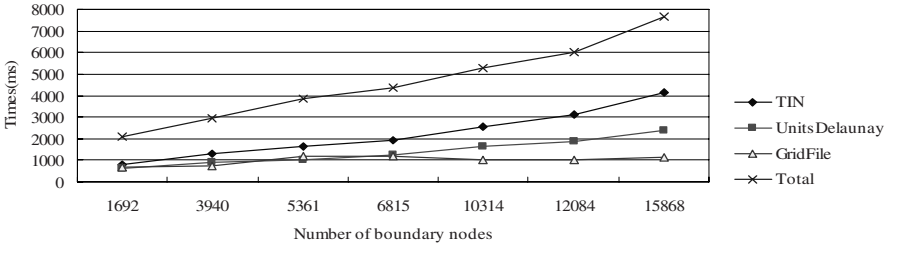


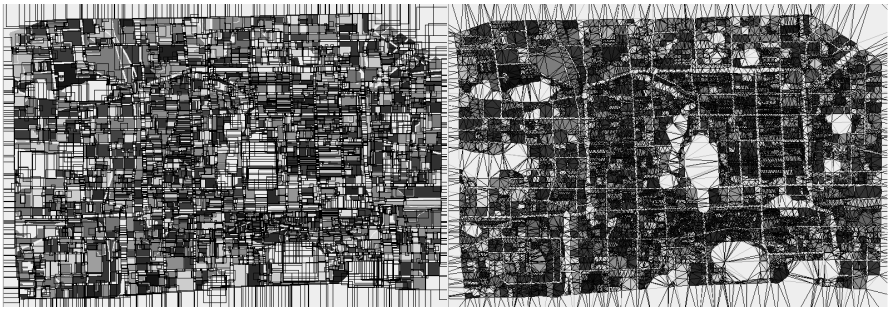
Fig. 3. Relation curve between number of boundary nodes and time of generated GridFile

From Table 1 we can see that, 2.42 binary relationships are stored in a bucket and 4.428 candidate geometries are searched in every time. The average time of reading geometry from Oracle is 11 (ms), and final shooting rate of GridFile is 74.18%.

Table 1. Capability of fixed grid 50×50

ID	Row	Column	RBS	NC	Time	NR
1	15	29	32	4	44	3
2	19	21	32	4	44	3
3	22	28	32	3	33	3
4	36	15	32	4	44	4
5	22	27	64	7	77	2
6	25	21	48	5	55	4
7	37	22	32	4	44	4
∑/*n*				4.428		3.285

BRS-reading bytes from disk; NC-number of candidate set; Time1-time of getting candidate set from Oracle; NR-number of refinement set.



(a)Delaunay triangulation and Classification (c) UnitsDelaunay spatial approximate

Fig. 4. Partial experimental Results

5 Conclusions

In this paper, we proposed a new data structure UnitsDelaunay and integrated it into Quad Gridfile spatial index to calculate the natural adjacency relations in spatial database. Reviewing this paper, we think that the following two areas are to be strengthened in future: (1) The spatial approximation of unit^2. The MBR of the spatial object is regarded as approximate geometry object. However, in practice we found its scope too large to cause large candidate set and increase the size of index file. The performance of spatial query processing can be improved by decomposing a complex object into a small number of simple components. We will consider decomposing complex unit structure with Delaunay triangulation method. (2) We also note that, the natural adjacency relationship can be got from Delaunay structure. Can the others relationship and the new model of topological relation calculation based on Delaunay be got? If these can, we will calculate spatial relationship on the TIN, which will be an interesting and meaningful work.

Acknowledgments. The authors would like thank the National Science Foundation of China for their financial support of this work under research grant No. 40301042.

References

1. Jun, C., Voronoi-based Dynamic Spatial Data Model. Beijing: Publishing house of Surveying and Mapping, (2002) (in Chinese)
2. Devillers, O., On deletion in Delaunay triangulations. 15th Annual ACM Symposium on Computational Geometry, 181-188
3. Gahegan, M., Lee, I., Data structures and algorithms to support interactive spatial analysis using dynamic Voronoi diagrams. Computers, Environment and Urbar Systems, (2000)24: 509-537
4. Gold, C.M., Edwards, G., The Voronoi spatial data model: 2d and 3d applications in image analysis. ITC Journal, (1992)1: 11-19
5. Gold, C.M., Problems with handling spatial data-The Voronoi approach. CISM Journal, (1992)45(1): 65-80
6. Jiawei, Han. Micheline, Kamber., Data Mining Concepts and Techniques. Beijing: China Machine Press (2001) (in Chinese)
7. Jones, C.B., Bundy, G.L., Ware, J.M., Map Generalization with a Triangulated Data Structure. Cartography and GIS, (1995)22(4): 317-331
8. Jun, C., Renliang, Z., Zhilin, L., Voronoi-based K-order adjacencys relations for spatial analysis. ISPRS Journal of Photogrammetry and Remote Sensing, (2004)59(1-2): 60-72
9. Mir, Abolfazl, Mostafavi., Christopher, Gold., Maciej, Dakowicz., Delete and insert operations in Voronoi/Delaunay methods. Computers & Geosciences, (2003)29(4): 523-530
10. Okabe, A., Boots, B.N., Sugihara, K., Spatial tessellations: concepts and applications of Voronoi diagrams (2nd Edition). John Wiley and Sons(1992)

K-Centers Algorithm for Clustering Mixed Type Data

Wei-Dong Zhao[1], Wei-Hui Dai[2], and Chun-Bin Tang[2]

[1] Software School, Fudan University, Shanghai, 200433 China
[2] School of Management, Fudan University, Shanghai, 200433 China
{wdzhao, whdai, 011025523}@fudan.edu.cn

Abstract. The K-modes and K-prototypes algorithms both apply the frequency-based update method for centroids, regarding attribute values with the highest frequency but neglecting other attribute values, which affects the accuracy of clustering results. To solve this problem, the K-centers clustering algorithm is proposed to handle mixed type data. As the extension to the K-prototypes algorithms, hard and fuzzy K-centers algorithm, focusing on effects of attribute values with different frequencies on clustering accuracy, a new update method for centroids is proposed in this paper. Experiments on many UCI machine-learning databases show that the K-centers algorithm can cluster categorical and mixed-type data more efficiently and effectively than the K-modes and K-prototypes algorithms.

Keywords: Cluster analysis, K-centers algorithm, centroid, mixed type data.

1 Introduction

As a well-known data mining method, cluster analysis plays an important role in many applications. Clustering can be used to discover clusters of data sets and has been applied to a variety of fields.

The basic K-means algorithm, using the centroid (mean) of objects with numeric values in the same group to represent a cluster, is proposed by Macqueen [1]. Since then, many clustering algorithms have been proposed to improve cluster analysis from different perspectives. Bezdek et al developed a fuzzy version of the K-means algorithm[2]. The K-means algorithm is widely used for clustering not only because of its simplicity and efficiency but also its ability to clustering large data sets. However, working on only numeric data limits the use of the K-means algorithm when much categorical data is frequently dealt with. The K-modes algorithm, using a new dissimilarity measure and frequency-based method to update modes, can then cluster categorical data[3]. The K-prototypes algorithm then integrates the K-means and K-modes algorithms to allow for clustering mixed numeric and categorical valued data sets[4,5]. Moreover, the fuzzy K-modes and fuzzy K-prototypes algorithms are developed based on fuzzy set theory[6,7], and they have built up clustering for mixed numeric and categorical valued data. However, both of the K-modes and K-prototypes algorithms use the frequency-based update method, that is, they both select attribute values which appear most frequently as centroids (cluster centers), but they fail to take into consideration the effect of other attribute values with low frequency on

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1140–1147, 2007.

centroids, which leads to the instability of these algorithms and decreases clustering accuracy.

A new similarity measure is put forward in this paper, which considers the influence of attribute values with different frequencies on centroids. The new algorithm based on the K-prototypes algorithm can also cluster data of mixed nature (numeric and categorical).

2 Basic Concepts

Definition 1. Let $A_1, A_2, \ldots, A_m$ be a set of attributes in a mixed numeric and categorical valued data set D in which $A_1, A_2, \ldots, A_p$ are numeric attributes and $A_{p+1}, A_{p+2}, \ldots, A_m$ are categorical attributes. The domain of A_j is denoted by $Dom(A_j)$. The domain of A_j for categorical attributes is denoted by $\{a_j^{(1)}, a_j^{(2)}, \ldots, a_j^{(n_j)}\}$, where n_j is the number of categorical attribute A_j. An object $X_i \in D$ can be represented as an m-dimension vector $(x_{i1}, x_{i2}, \ldots, x_{im})$, where $x_{ij} \in Dom(A_j)$. It is obvious that p satisfies $0 \le p \le m$. Specifically, when $p = 0$ all the attributes are categorical and when $p = m$ all attributes are numeric, while attributes are of mixed type when $0 < p < m$.

Definition 2. Let Z_l be a centroid (clustering centers), denoted by $Z_l = (z_{l1}, z_{l2}, \ldots, z_{lm})$. z_{lj} is used to represent the mean of attribute A_j for $0 \le j \le p$. For $p+1 \le j \le m$, z_{lj} is a n_j dimensional vector represented as $(z_{lj}^{(1)}, z_{lj}^{(2)}, \ldots, z_{lj}^{(n_j)})$, where $z_{lj}^{(r)}$ denotes the similarity between the centroid Z_{lj}^* and attribute value $a_j^{(r)}$ which satisfies $0 \le z_{lj}^{(r)} \le 1$. $n_{lj}^{(r)}$ is used to represent the number of attribute values $a_j^{(r)}$. If attribute value $a_j^{(r)}$ doesn't exist in a cluster, that is $n_{lj}^{(r)} = 0$, then $z_{lj}^{(r)} = 0$. The sum of all these similarities $z_{lj}^{(r)}$ equals 1 if all attribute values appear in the cluster.

For example, if categorical attribute A_j has 3 possible values and the centroid Z_{lj}^* is $(0.5, 0.3, 0.2)$, then the similarity between $a_{lj}^{(1)}$ and the centroid is 0.5.

Definition 3. The dissimilarity (distance) $d(X_i, Z_l)$ between object X_i and centroid Z_l is composed of two parts: numeric dissimilarity and categorical dissimilarity.

$$d(X_i, Z_l) = \beta d_r(X_i, Z_l) + \gamma d_c(X_i, Z_l) = \beta \sum_{j=1}^{p} (x_{ij} - z_{lj})^2 + \gamma \sum_{j=p+1}^{m} [1 - f(x_{ij}, z_{lj})]^2$$

where $f(x_{ij}, z_{lj}) = \{ z_{lj}^{(r)} \,|\, x_{ij} = a_{lj}^{(r)} \}$.

In definition 3, Euclidian distance is used for numeric attributes, while the categorical dissimilarity is derived from the similarity between corresponding categorical attributes. The determination of weight parameters β and γ are relatively complicated.

If $p = 0$, which means data is categorical type, then $\beta = 0, \gamma = 1$; if $p = m$, which means data is numeric type,then $\beta = 1, \gamma = 0$. However, if $0 < p < m$, which means data is mixed type, it is difficult to choose the right weight parameters. Generally, we set $\beta = 1$ and choose a greater weight parameter for γ if we give emphasis to categorical valued attributes or a smaller value for γ otherwise.

3 K-Centers Algorithm

The K-centers clustering technique, using the objective function similar to that of the basic K-means algorithm, redefines centroids and the dissimilarity of categorical objects and produces several clusters after a number of iterations. The aim is to minimize the objective function $F(W, Z) = \sum_{l=1}^{k} \sum_{i=1}^{n} w_{li}^{\alpha} d(X_i, Z_l)$, Subject to

$$0 \le w_{li} \le 1$$

$$\sum_{l=1}^{k} w_{li} = 1$$

$$0 < \sum_{i=1}^{n} w_{li} < n, \quad \sum_{r=1}^{n_j} z_{lj}^{(r)} = 1$$

where $\alpha \ge 1$ is the fuzzy parameter. When $\alpha = 1$, it is hard clustering, and when $\alpha \ge 1$, it is fuzzy clustering. *W* is a *k*-by-*n* membership matrix, implying belonging of each object to a cluster to some degree. So an object belongs to only one cluster when $\alpha = 1$, but may belong to several clusters indefinitely when $\alpha > 1$.

Similar to the K-means algorithm, minimizing the objective function *F* is a nonlinear programming problem. The K-centers algorithm focuses on a incremental optimization: first initialize centroids Z, find the proper membership matrix *W* to minimize the objective function F, and then regulate *W* and minimize objective function *F* to get new centroids *Z*. The steps will be iterated until *F* cannot be optimized further.

Theorem 1. Let centroids Z^* be fixed, then the objective function *F* is minimized if and only if *W* satisfies:

$$when\ \alpha = 1,\ w_{li}^* = \begin{cases} 1, & d(X_i, Z_l^*) \le d(X_i, Z_h^*), 1 \le h \le k, \\ 0, & otherwise\ . \end{cases}$$

$$when\ \ \alpha > 1,\ \ w_{li}^* = \begin{cases} 1, & if\ X_i = Z_l^*, \\ 0, & if\ Xi = Z_h^*, h \ne l, \\ 1 \Big/ \sum_{h=1}^{k} [\frac{d(X_i, Z_l^*)}{d(X_i, Z_h^*)}]^{1/(\alpha-1)}, & otherwise. \end{cases}$$

When $\alpha = 1$, the objective function *F* is linear programming problem and when $\alpha > 1$, *F* is a convex function. The minimum value of this nonlinear function can be computed using the Lagrange Multiplier.

Theorem 2. Let membership matrix W^* be fixed, then the objective function F reaches the minimum value if and only if centroids Z is assigned as follows: for numeric attribute A_j ($1 \le j \le p$) ,

$$z_{lj} = (\sum_{i=1}^{n} w_{li}^{\alpha} x_{ij}) / \sum_{i=1}^{n} w_{li}^{\alpha} \tag{1}$$

For categorical attribute A_j, $p+1 \le j \le m$,

$$z_{lj}^{(r)} = \begin{cases} 1 - \dfrac{(|N_{lj}|-1)\dfrac{1}{n_{lj}^{(r)}}}{\sum\limits_{t \in N_{lj}} \dfrac{1}{n_{lj}^{(t)}}}, & r \in N_{lj}, \\ 0, \quad otherwise. \end{cases} \tag{2}$$

Where $n_{lj}^{(r)}$ represents the number of attribute value $a_j^{(r)}$, that is $n_{lj}^{(r)} = \sum_{i=1}^{n} \{ w_{li}^{\alpha} \mid x_{ij} = a_{lj}^{(r)} \}$, N_{lj} is a value set of $a_j^{(r)}$ which have ever appeared in data sets. $N_{lj} = \{ r \mid n_{lj}^{(r)} > 0 \}$ and $|N_{ij}|$ is the number of elements in N_{lj}.

The equation (2) can be transformed as followings:

$$z_{lj}^{(r)} = 1 - \frac{(|N_{lj}|-1)\dfrac{1}{n_{lj}^{(r)}}}{\sum\limits_{t \in N_{lj}} \dfrac{1}{n_{lj}^{(t)}}} = \frac{\dfrac{1}{n_{lj}^{(r)}} + \sum\limits_{t \in N_{lj}} (\dfrac{1}{n_{lj}^{(t)}} - \dfrac{1}{n_{lj}^{(r)}})}{\sum\limits_{t \in N_{lj}} \dfrac{1}{n_{lj}^{(t)}}}$$

It is more reasonable that for categorical data sets, the K-centers algorithm can derive centroids by computing the percentage of the reciprocal of attribute values, considering the effect of attribute values with different frequencies.

The following properties of the K-centers algorithm are intuitive:

Property 1. If $n_{lj}^{(r)} \sum_{t \in N_{lj}} \dfrac{1}{n_{lj}^{(t)}} < |N_{lj}| - 1$, then $z_{lj}^{(r)} < 0$

This is inconsistent with the definition $z_{lj}^{(r)} \ge 0$. Meanwhile, it indicates that $n_{lj}^{(r)} > 0$. However, it can be neglected because of low frequency. Thus the centroid for A_j should be computed again.

Property 2. For an attribute A_j, it is impossible to neglect all attribute values.

Property 3. if $n_{lj}^{(r)} > n_{lj}^{(t)}$, then $z_{lj}^{(r)} > z_{lj}^{(t)}$.

Property 4. Let $n_{lj}^{(r)} = \max\{ n_{lj}^{(p)}, p \in N_{lj} \}$, $n_{lj}^{(s)} = \min\{ n_{lj}^{(p)}, p \in N_{lj} \}$,then $z_{lj}^{(r)} \ge [z_{lj}^{(r)}]'$, $z_{lj}^{(s)} \le [z_{lj}^{(s)}]'$,where $[z_{lj}^{(r)}]' = n_{lj}^{(r)} / \sum_{t=1}^{n_j} n_{lj}^{(t)}$ which emphasizes the percentage of different categorical attribute values.

Property 3 and 4 indicate that the K-centers algorithm assigns higher similarity to attribute values that occupies higher percentage and lower similarity to those, which occupy lower percentage.

The K-centers algorithm, both hard and fuzzy clustering, can be described as follows:

(1) Choose initial centroids $Z^{(1)}$ and a parameter ξ to decide whether to terminate the iteration or not.

(2) Determine $W^{(1)}$ that minimizes $F(W, Z^{(1)})$, Set $t = 1$.

(3) Determine $Z^{(t+1)}$ that minimizes $F(W^{(t)}, Z)$. If

$| F(W^{(t)}, Z^{(t+1)}) - F(W^{(t)}, Z^{(t)}) | < \xi$, then stop;

(4) Determine $W^{(t+1)}$ that minimizes $F(W, Z^{(t+1)})$. If

$| F(W^{(t+1)}, Z^{(t+1)}) - F(W^{(t)}, Z^{(t+1)}) | < \xi$, then stop;otherwise,set $t = t + 1$ and go to step (3).

It can be proved that the iteration of the K-centers algorithm, the fuzzy parameter α and the convergence of the objective function satisfy theorem 3.

Theorem 3. If the fuzzy parameter α is large enough, the membership matrix cannot determine definitely which cluster the data belong to. So the fuzzy parameter α cannot be assigned to too large a value in the K-centers algorithm, otherwise clustering results may be unsatisfactory.

Generally speaking, the K-centers algorithm is relatively simple. Its time complexity is $O(tkmn)$, where t is the number the algorithm iterates. In most cases, $t, k, m << n$ and thus the time complexity is approximately $O(n)$, which indicates that the relationship between the size of data sets and computation complexity is linear. So the K-centers algorithm is an efficient clustering algorithm, and can process large data sets effectively. However, the K-centers algorithm also needs to be improved. For example, the algorithm is more efficient for convex or spherical shape data sets; it also predefines a user-specified number k and cannot deal with outliers effectively.

4 Experimental Results

To evaluate the performance of the K-centers algorithm, experiments have been performed on several data sets and then the impact of different parameters on the algorithm is discussed. The data for the experiments are from UCI machine learning repository, including categorical databases soybean and voting, mixed type databases credit and cleve. Usually, numeric data are measured in different units. To deal with this problem, the transformation is necessary before clustering in order to map all the numeric data to the range [0, 1]. Then, according to experiment results, we make comparisons between the K-centers, K-modes and K-prototypes algorithms.

When data are clustered by hard K-centers clustering algorithm, an object is only partitioned to one cluster; when data are clustered by fuzzy K-centers clustering algorithm, an object may be assigned to multiple clusters indefinitely. Although the membership matrix describes belongings of each object to different clusters, it fails to

clearly indicate which cluster one object belongs to. In practice, one object is assigned to the cluster l to which it has the maximum membership degree, that is $w_{lj} = \{w_{lj} \geq w_{tj} \mid t = 1,2,...,k\}$.

Experiment results can be analyzed using the accuracy $r = (\sum_{l=1}^{k} a_l)/n$, where a_l is the number of objects shared by the cluster l and its original data set, n is the size of data sets.

As Fig. 1 shows, for categorical data, the accuracy of hard K-centers algorithm is close to that of hard K-modes algorithm, but the accuracy of fuzzy K-centers algorithm is higher than that of fuzzy K-modes algorithm. On the other hand, as the fuzzy parameter α increases, the accuracy of the K-centers algorithm improves rapidly, but that of the K-modes algorithm decreases slightly. It is obvious that the K-centers algorithm is more accurate than the K-modes algorithm when applied to soybean database. Similarly, the accuracy of the K-centers and K-modes algorithms on voting database can also be compared in Fig. 2. In general, when the fuzzy parameter increases, the clustering accuracy is relatively stable no matter it is the K-centers or K-modes algorithm, that is, there is not notable improvement, which means the K-centers and K-modes algorithms can result in rather accurate results. But the K-centers algorithm remains superior to the K-modes algorithm. Furthermore, we also find that the results are stable using the fuzzy K-centers algorithm. The algorithm will produce the same result on voting database no matter how initial centroids are chosen. But for the fuzzy K-modes algorithm, results vary with different initial centroids.

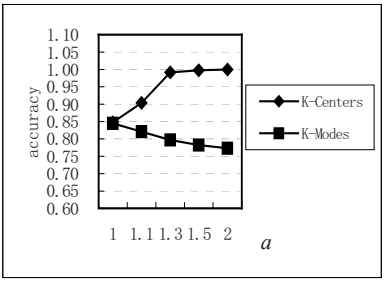


Fig. 1. K-centers and K-modes on soybean

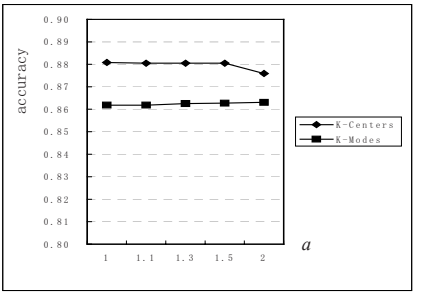


Fig. 2. K-centers and K-modes on voting

The impact of weight parameter γ used in clustering data with mixed numeric and categorical values cannot be neglected, because improper γ will have great influence on the effectiveness of clustering. Usually, the selection of γ requires professional knowledge. In most cases, if categorical data is dealt with, we may choose a larger value for γ; otherwise we choose a smaller value. Similarly, the K-prototypes algorithm has difficulty with how to identify the value of γ. Therefore, different values of γ are tried in the experiments. Since initial centroids may have different effect on clustering results, each of the two algorithms is run 100 times. Fig. 3 shows the average accuracy of clustering results with respect to variant fuzzy parameters. The average accuracy of clustering results first reaches a peak point as the fuzzy parameter α increases, and then begins to decrease. Through further analysis, we find that when the fuzzy parameter α is

larger than 1.1, the accuracy remains unchanged no matter whatever the initial centroids are, but when the fuzzy parameter α is 1.2, clustering accuracy reaches the peak. Given the same fuzzy parameter α, the best γ is shown to be 0.1 when the accuracy is the highest. Fig. 3 also shows the clustering accuracy of the K-prototypes algorithm with varying fuzzy parameters. In general, with different fuzzy parameters, the difference in accuracy seems to be slight, and the best accuracy is 0.76. The results also show that the selection of initial centroids for K-prototype has obvious impact on the clustering accuracy. As shown in Fig. 3, the accuracy of K-centers can exceed 0.82, which indicates that the K-centers algorithm excel the K-prototypes algorithm in clustering credit database.

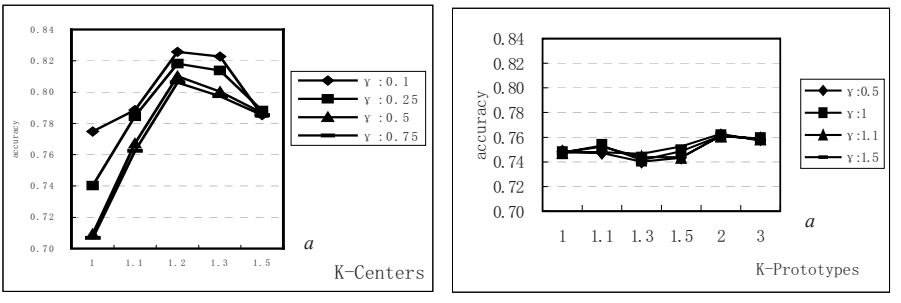


Fig. 3. Clustering accuracy for K-centers and K-prototypes applied to credit

Fig. 4 shows how K-centers and K-prototypes behave when applied to cleve database with different weight γ. The accuracy of the K-centers algorithm improves as the fuzzy parameter increases. It reaches the peak when the fuzzy parameter is 2, but the accuracy of the K-prototypes algorithm decreases slightly. After careful analysis of clustering results, we also find that the results of fuzzy K-prototypes algorithm are not stable when initial centroids differ. As for fuzzy K-centers algorithm, the results are stable, which means that the algorithm is not sensitive to the change of initial centroids. The better weight value for the K-centers algorithm is 0.3. The clustering accuracy reaches 0.84 when α is 2. But the accuracy of K-prototypes algorithm is less than 0.78, which seems that the K-centers algorithm is more effective than K-prototypes applied to cleve database.

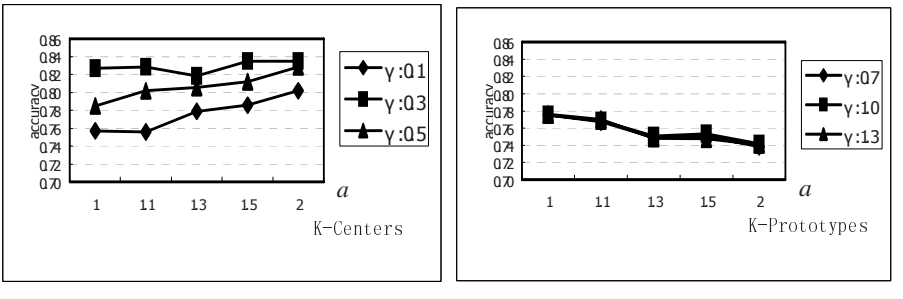


Fig. 4. Clustering accuracy of K-centers and K-prototypes algorithm on cleve

For space limit, not all experimental curves or graphs are given above and the results are also observed as for many other data sets. Domains that regularly produce data of mixed nature will be benefited although it is needed to provide further justification for these results.

In conclusion, the fuzzy K-centers algorithm gives more accurate clustering results than the corresponding hard one, and can produce more stable results with different initial centroids. Compared with the K-prototypes algorithm, the K-centers algorithm produces more effective results.

5 Conclusion

As the extension to the K-prototypes algorithm, the K-centers algorithm can cluster mixed type data more effectively. In hard and fuzzy K-centers algorithm, we propose a new update method for centroids. Unlike the K-modes and K-prototypes algorithms, which only focus on attribute values with the highest frequency, the proposed algorithm considers attribute values with various frequencies. The K-centers algorithm is illustrated to be able to perform clustering better in most cases. The experiments on UCI machine learning database also show that the K-centers algorithm produce more accurate results than the K-modes and K-prototypes algorithms. Nevertheless, further improvements still can be made on the K-centers algorithm. For example, we can combine the K-centers algorithm with genetic algorithm so as to overcome local optimum. How to provide general guideline for users to choose appropriate parameter α, γ is also worth further research.

Acknowledgments. This research was supported by the Natural Science Foundation of China (No. 70301004).

References

1. MacQueen J. B.: Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, Berkeley (1967)281-297.
2. Bezdek J.C.: A convergence theorem for the fuzzy ISODATA clustering algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence.1 (1980) 1-8.
3. Zhe X. Huang: A fast clustering Algorithm to Cluster very Large Classifiable Data Sets in Data Mining. In: Proceedings of the SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery. Canada: The University of British Columbia (1997) 1–8.
4. Zhe X. Huang: Clustering Large Data Sets with Mixed Numeric and Categorical Values. In: Proceedings of the First Pacific Asia Knowledge Discovery and Data Mining Conference. World Scientific, Singapore (1997) 21–34.
5. Zhe X. Huang: Extensions to the K-means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining Knowledge Discovery. 3(1998) 283-304.
6. Zhe X. Huang, M. K. Ng: A Fuzzy K-modes Algorithm for Clustering Classifiable (categorical) data. IEEE Transactions on fuzzy systems.4(1999) 446-452.
7. Chen Ning, Chen An, Zhou Longxiang: Fuzzy K-prototypes Algorithm for Clustering Mixed Numeric and Categorical Valued Data. Chinese Journal of Software.8 (2001) 1107-1119

Proposion and Analysis of a TCP Feature of P2P Traffic

Li-Juan Zhou, Zhi-Tang Li, and Hao Tu

Network and Computing Center,Huazhong University of Science&Technology
430074, P.R. China
{ljzhou, leeying ,tuhao}@hust.edu.cn

Abstract. Over the last years, the wide use of the P2P application has lead to the rapid growth of network traffic. Thus, the accurate classification of P2P traffic becomes a challenging problem. This paper proposes some new fundamental characteristics of TCP traffic to achieve its discrimination. To prove the universality of the values of our proposed features, we also present some analytic estimates of them using Pareto as the distribution of user lifetime in real P2P systems. Finally, we apply SVM to the discrimination of these features in order to automate the processing of the discrimination, and its accuracy is demonstrated as a result of some experiments.

1 Introduction

A significant trend has emerged in recent years whereby use of the Internet for Peer-to-Peer file sharing. The leading content shared in the P2P systems, such as audio and video files, tend to be large in size[1]. So, the wide use of the P2P application has lead to the rapid growth of network traffic in addition to the illegal file sharing. Thus, it has become an urgent requirement to discriminate the users of P2P file sharing application from other users.

In this paper, we first propose some univeral features of file-sharing P2P traffic derived. Then, we present some analytic estimates of them to prove the value gained from our experiment of the proposed feature. Our methodology can discriminates the P2P users based on flow connection patterns of P2P traffic, and without relying on packet payload. Finally, we will demonstrate the effectiveness of the proposed discrimination.

2 Previous Work for Discriminating P2P Traffic and Problems

The most P2P traffic researches have used the data collected from routers across some large ISP's backbone and performs systematic characterization of P2P traffic, such as distribution and workload, often motivated by the dominance of that protocol in a particular provider's infrastructure or during a specific time period [2] [3] [4] [5] [6]. And more importantly, the analysis using aggregated data could cover some real specialties of P2P traffic. For example,in [2] the authors find "newly initiating connection rate " is totally different between P2P and traditional traffic .After analyzing

Z.-H. Zhou, H. Li, and Q. Yang (Eds.): PAKDD 2007, LNAI 4426, pp. 1148–1155, 2007.

many traffic of single host in CERNET including both P2P and tradition, we find this conclusion isn't suitable for all kinds of P2P traffic because a P2P host can make several TCP connections with an other P2P peer and with the same destination port and different source port within a download process. From the discussions above, there is a strong requirement for a new discrimination technique based on some universal features of the P2P traffic.

3 Proposal of Features for Discriminating P2P Traffic

We focus on the P2P traffic observed at a single host, and use tens of seconds as the time granularity of analysis, not an hour like in the relative work before. The nature of the P2P traffic lies in the fact every P2P host services as both server and client, conducing to the highly decentralized, self-organizing systems, and the large number of hosts involved and the transient peer membership. These hosts may differ in many aspects, especially the rate of random departure decisions of end-users. Any changes of these aspects can make the connectivity of P2P networks very different at any moment [7]. To keep it's download speed not to decrease, a P2P host can continually initiate TCP connections with others which could be online in a very low probability due to the dynamics of P2P systems.P2P hosts should have a small success rate of initiating connections, and a common (not P2P) host would connect with all kinds of servers in the Internet at a much higher success rate. It is because all the servers must keep their service always acquirable in any client/server systems while peers in P2P system are personal computers: and not always operational and stable. We think the low success rate of TCP connections of peers is the common character of all P2P systems. Based on it, we propose some traffic features for P2P discrimination technique. The main focuses of the proposed features are SYN and SYN/ACK packets, since these packets are used for the establishment of TCP connections of P2P application without exception.

a = number of different destination IPs of transmitted SYN packets
b = number of different soucre IPs of recieved SYN/ACK packets
From the above attributes we derived the following features:
(1) connection responsed success rate

$$r_c = b/a \tag{1}$$

(2) percentage of the instantaneous responsed rate whose value is equal to 1 and calculated every 30 seconds

We proposes r_c as the universal feature for discriminating P2P traffic from other traffic. A key reason for choosing this feature is the large difference in its values of P2P traffic between other traffic, which will be demonstrated in detail in the following sections.

4 Methodology

We focuse on two P2P protocols:BitTorrent and eMule which are used most popularly in CERNET(China Education and Research Netwrok). From the analysis of the

protocols[10], we can get the conclusion the most used transfer layer protocol of connections established between P2P peers is TCP[11].We prepared two kinds of traffic data set:one is P2P traffic generated by eMule or BitTorrent;the other is the traffic without P2P communication, such as traffic of email/DNS server and common clients, even a port scanning application *superscan*. Using the prepared traffic data sets, we firstly calculated the connection responsed success rate rc of one hour traffic. Figure 1 shows the distribution of rc of two kind of traffic.

From Fig. 1,we can see that the tradition traffic has very high value in r_c as our expectation , and the P2P traffic has the value in the range [0.2,0.65] . There is a part of P2P traffic generated by eMule also has high value of r_c, so we can see a little overlap. After analyzing the communicating process of eMule [9],we find that eMule peer would periodically send UDP messages which are used to find out whether it can start download the file. This mechanism leads to the high responsed rate of TCP connecting request at the cost of very low success rate of UDP response. At the bottom of Fig. 1,there are a few points which generated by *Superscan* .This kind traffic has a very low responsed rate which is even much smaller than P2P traffic because of its totally randomly behavior. We think the responsed connection rate of P2P traffic in a short time piece between two peers can't be always high(=1), while it can be easy for normal tradition traffic. So, we next calculated the percentage of rc=1 every 30 seconds for all data sets. We think 30 seconds is felicitous to be chosen due to the data retransfer mechanism of many applications.Fig.2 shows the distinct difference between the proportion of rc=1 of P2P and traditional traffic. The P2P plots are in area between 0 and 0.2, while the tradition traffic has the value mostly in [0.5,0.9]. The superscan still has very low value in this feature as we expected.

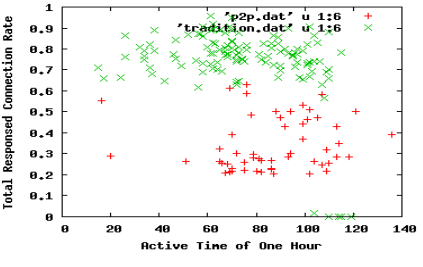


Fig. 1. Connection Responsed Success Rate

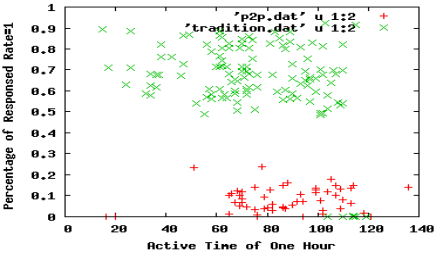


Fig. 2. Percentage of Responsed Rate=1

From Fig.2,we can see although some P2P traffic which have a high value in total responsed rate of an hour get a very low percentage of r_c=1. The reason is the P2P applications try to connect with other peers as many as they can even within a very short time to keep a certain download speed. In tradition network behavior model, connecting with different hosts in high frequency is unnormal. Form Figs. 1 and 2, neither connection responsed rate of an hour nor the distribution of rc=1 can be used to discriminate P2P traffic as a single traffic feature. But they can be combined together to get an accurate identification of P2P traffic.

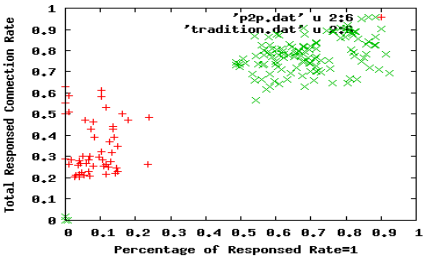


Fig. 3. Identification of P2P traffic Through Two Features Combined

From Fig. 3 ,we can find all the traffic are plot out three parts clearly: normal tradition,P2P and unnormal tradition traffic from top to bottom. As we described above, our proposed traffic features can be useful for accurate traffic discrimination.

In the two following sections, We will prove the usability of our proposed features both from theoretical and experimental ways.

5 Theoretical Evaluation

In this section we present some analytic estimates of the universality of the values of our proposed features based on the distribution of user lifetime in real P2P systems.

5.1 Lifetime Model of Real P2P Systems

In our model, each arriving user is assigned a random lifetime *Li* drawn from some distribution $F(x)$, which reflects the behavior of the user and represents the duration of his/her services to the P2P community. It has been observed that the distribution of user lifetimes in real P2P systems is often heavy-tailed (i.e., Pareto)[8],[9], where most users spend minutes per day browsing the network while a handful of other peers exhibit server-like behavior and keep their computers logged in for weeks at a time. To allow arbitrarily small lifetimes, we use a shifted Pareto distribution (3) to represent heavy-tailed user lifetimes, where scale parameter $\beta > 0$ .Note that the mean of this distribution(4) is finite only if $\alpha > 1$, which we assume holds in the rest of the paper.

$$F(Li \le x) = 1 - (1 + x/\beta)^{-\alpha}, x > 0, \alpha > 1 \tag{2}$$

$$E[Li] = \beta/(\alpha - 1) \tag{3}$$

There have been a number of studies reporting on experimental data collected from currently deployed P2P systems. From them we find that it may appear that $E[Li] = 1$ hour(the mean online stay is 1 hour) is rather large for current P2P systems. If we set Pareto lifetimes with $\alpha = 3$ and $E[Li] = 1$ hour, we can get $\beta = 2$ by (3)and then by (2) can calculate that 51% of the users depart within 30 minutes of their arrival which accords with the fact of BitTorrent-like system, but in eMule-like system users will stay more longer.

5.2 Reason for Considering Surviving Rate of Peers

As mentioned above, the P2P traffic features we proposed are different from the traditional traffic because the dynamics of P2P system: users voluntarily decide to leave the system based on their attention span and/or browsing habits, so that even a user announced its existence to you at some time before, you could still fail when you touch with it just after a short time. In the following, we will answer the question :"if we know some peers are online at time T1 in a real P2P system, how many peers are still online when passing a time t ",in other words, what is the surviving rate of these peers after certain time t?

In BitTorrent system, during the download process the tracker periodically sends updated information about new download locations which are collected through the announcement of all clients. In the eMule network, a client connects to an eMule server for getting information about desired files and available clients [11].According to the statement above about the communication mechanism of centralized file-sharing P2P systems, the feature "connection responsed success rate " of one peer must be determined by the surviving rate of all those clients contacted ,and we even think they should be in some kind directly related if the systems are large enough and steady. Next, we will infer the expression of the surviving rate of P2P peers.

5.3 Derivation of Surviving Rate

To keep the derivations tractable, we impose serveral restrictions on the system we study. We first assume that users join a network that has evolved sufficiently long so as to overcome any transient effects. This assumption is usually satisfied in practice since P2P systems continuously evolve for hundreds of days or weeks before being restarted (if ever) and the average lifetime $E[Li]$ is negligible compared to the age of the whole system when any given peer joins it. Our second assumption requires certain stationarity of lifetime Li. This means that users joining the system at different times of the day or month have their lifetimes drawn from the same distribution $F(x)$. Finally, we should note that these stationarity assumptions do not apply to the number of nodes n as long as $n >> 1$ stays sufficiently large. In the following, we give a precise description of the question we need to study. It includes three statements as below:

Statement 1. In P2P systems, if N_{T1} is the number of online peers at moment T1 ,after time t, the surviving rate of these peers (defined as SR) is (4).

Statement 2. When a user contacts with a group of peers whose locations got from trackers or index servers constantly, the intervals of these peers announce themselves to trackers and the time when the user gets the locations from servers are various according to the time successively. We assume the probability distrubiton of the sum of announcing peers is average at any moment and at one moment the probability of peers is also mean for various lifetimes. So, the connection responsed success rate could be simply represented as the mean value of all SR within some time T whose length is decided by the practical mechanism of the system as (5).

$$SR_t = \frac{N_{T1+t}}{N_{T1}} \tag{4}$$

Statement 3. If the system are large and steady enough ,the SR is only correlative with the length of passing time t.

$$Rc \approx \frac{1}{2T}\int_0^T SR_t dt \tag{5}$$

Assuming that N is large and the system has reached stationarity, the PDF of *Li* is given by:

$$f(Li = x) = F(Li \le x)' \tag{6}$$

Applying the formula (2) to (6),we get :

$$f(Li = x) = \frac{\alpha}{\beta}(1 + x/\beta)^{-\alpha-1}, x > 0, \alpha > 1 \tag{7}$$

The probability density function (PDF) *f(x)* serves to represent a probability distribution in terms of user lifetime at arbitrary moment in the system. For example, the percentage of user who have existed in system for 10 minutes at any time could be represented by f(10). We must notice an important understanding that is at one moment the peers alive in the system maybe have already survived for any potential amount of time from $0 \to +\infty$.Then,we have the following result:

Theorem 1. From time T1 to time T1+t, the SR_t of peers for Patero lifetimes with $F(x) = 1 - (1 + x/\beta)^{-\alpha}$, $\alpha > 1$ is given by:

$$SR_t = 1 - \int_0^{+\infty} [F(x)' - F(x+t)']dx \tag{8}$$

The part of integral in (8) represents the probability of those peers departing from the system midway. According (6) and (7),re-write (8):

$$SR_t = 1 - \int_0^{+\infty} \frac{\alpha}{\beta}\left[(1 + x/\beta)^{-\alpha-1} - (1 + (x+t)/\beta)^{-\alpha-1}\right]dx \tag{9}$$

The final expression we get through mathematic operations is as follows:

$$SR_t = (1 + t/\beta)^{-\alpha} \tag{10}$$

This result gives a proof to our statement 3.Substituting (10) into (5),we get:

$$Rc \approx \frac{1}{2T}\int_0^T (1 + t/\beta)^{-\alpha} dt \tag{11}$$

Next setting α=3, β=2 and T=0.5/1/1.2 hour, we compute $Rc_{0.5}$,Rc_1 and $Rc_{1.2}$.We choose these three time pieces as the intervals of time T, because the very busy trackers in BitTorrent system always set the announcment interval of client as 1.2 hour or

more longer while the general trackers choose 0.5 hour ,and 1 hour is the mean lifetime. The results are :$Rc_{0.5} = 0.36$,$Rc_1 = 0.28$,$Rc_{1.2} = 0.25$.The conclusions above are based on the assumption that the trackers or index servers always ensure all offline peers are not in the list given to the client. So, we can see that the theoretical results are consistent with what we obtained through our experimentations before.

6 Experimental Evaluation

In this section we evaluate the accuracy of our methodology by using the SVM method to discriminate P2P traffic with our proposed traffic features. Indeed, to minimize false positives in P2P traffic identification, we can firstly filter P2P traffic by some well-known ports. We took an SVM package LibSVM-2.82 using default parameters except the parameter *g(gamma)* must be set to bigger than 2.We choose 100 examples from the data sets used in the previous section as a training set to determine a separating hyperplane. We gathered 10 traffic data sets of another P2P application named PPLive which is a famous P2P application [12], 5 traffic data sets generated by *nmap* , another popular scanning application , and 10 data sets of several common hosts when no P2P or scanning applications run on them. All these 25 traffic data are put in one file used for testing sets. We built three training file: train1,train2, train3: the first one only using the total responsed rate of one hour as the training attribute, the second one using the percentage of responsed rate which is equal to 1 counted every 30 seconds and the third one using both of them. The accuracy values of the predictions of SVM using three training file respectively and a same testing data sets are compared in Table 1.

Table 1. Comparison of Accuracy of three training file The feature1 and feature2 below implies "reponsed rate" and "percentage of reponsed rate =1" respectively

attri. / file	Feature1	Feature2	Accuracy
Train1	√		48%
Train2		√	80%
Train3	√	√	100%

Only using responsed rate as training attribute like in train1,the 8 sets of 10 traffic of PPLive are classified to normal tradition traffic and all 5 sets of *nmap* traffic are labeled as P2P traffic. Similarly, when percentage of responsed rate =1 used for training attribute, all *nmap* data sets are considered as P2P traffic. From Table 1, the one feature case shows bad results compared to two features case, and the accuracy can get at 100% when two features combined.

7 Conclusion

This paper firstly propose some universal features of file-sharing P2P traffic, and present some analytic and experimental estimates of our proposed feature. The characteristics of P2P traffic are distinct from other traffic which is confirmed both by the

result of our theoretical analysis and the effective data mining of SVM. Future study will also be needed to adapt our algorithm for active real-time monitoring of P2P traffic and to cover P2P applications other than eMule and BitTorrent.

References

1. Sen.S ,Jia Wang : Analyzing peer-to-peer traffic across large networks. Networking, IEEE/ACM Transactions on, Volume 12, Issue 2, April 2004 Page(s):219 – 232
2. Matsuda, T. Nakamura, F. Wakahara, and Y. Tanaka : Traffic Features Fit for P2P Discrimination. Information and Telecommunication Technologies, 2005. APSITT 2005 Proceedings. 6th Asia-Pacific Symposium on
3. Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc claffy: Transport Layer Identification of P2P Traffic. Proceedings of *the 4th ACM SIGCOMM conference on Internet measurement*, Oct. 2004
4. Spognardi. A, Lucarelli,.A., and Di Pietro.R. : A methodology for P2P file-sharing traffic detection. Hot Topics in Peer-to-Peer Systems, 2005. HOT-P2P 2005. Second International Workshop on,21 July 2005 Page(s):52 – 61
5. Hamada. T, Chujo. K., Chujo. T., and Yang. X.: Peer-to-peer traffic in metro networks: analysis, modeling, and policies. Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP, Volume 1,,19-23 April 2004 Page(s):425 - 438 Vol.1
6. Kattirtzis. C. , Varvarigos. E. , Vlachos. K. , Stathakopoulos. G. , and Paraskevas, M. :Analyzing Traffic across the Greek School Network. Local and Metropolitan Area Networks, 2005. LANMAN 2005. The 14th IEEE Workshop on, 18-21 Sept. 2005 Page(s):1 – 6
7. Kin Wah Kwong, Tsang, and D.H.K.: On the relationship of node capacity distribution and P2P topology formation. High Performance Switching and Routing, 2005. HPSR. 2005 Workshop on, 12-14 May 2005 Page(s):123 - 127
8. F. E. Bustemante and Y. Qiao: Friendships that Last: Peer Lifespan and its Role in P2P Protocols. Intl. Workshop on Web Caching and Distribution, September 2003.
9. S. Saroiu, P.K. Gummadi, and S.D. Gribble: A Measurement Study of Peer-to-Peer File Sharing Systems. MMCN, 2002.
10. http://www.bittorrent.org
11. http://www.cs.huji.ac.il/labs/danss/p2p/eMule/
12. http://www.pplive.com

Author Index